Enabling Agents to Communicate Entirely in Latent Space

Zhuoyun Du[1,2,3†*] **Runze Wang**[2†] **Huiyu Bai**[2] **Zouying Cao**[4]
Xiaoyong Zhu[2] **Yu Cheng**[2] **Bo Zheng**[2] **Wei Chen**[1] **Haochao Ying**[5✉]
[1]State Key Lab of CAD&CG, Zhejiang University
[2]Future Living Lab of Alibaba
[3]Zhejiang Key Laboratory of Medical Imaging Artificial Intelligence
[4]Shanghai Jiao Tong University [5]Zhejiang University
{duzy, haochaoying}@zju.edu.cn, yunze.wrz@alibaba-inc.com

Abstract

While natural language is the de facto communication medium for LLM-based agents, it presents a fundamental constraint. The process of downsampling rich, internal latent states into discrete tokens inherently limits the depth and nuance of information that can be transmitted, thereby hindering collaborative problem-solving. Inspired by telepathy, which bypasses symbolic language in communication, we propose **Interlat** (***Inter****-agent* ***Lat****ent Space Communication*), a paradigm that leverages the continuous last hidden states of an LLM as a representation of its thought for direct communication (termed **"latent communication"**). An additional learned compression process further compresses latent communication via latent space reasoning. Experiments demonstrate that Interlat outperforms both fine-tuned chain-of-thought (CoT) prompting and single-agent baselines, even across heterogeneous models, promoting more exploratory behavior and enabling genuine utilization of latent information. Further compression not only substantially accelerates inference by **up to 24**× but also maintains competitive performance through an efficient information-preserving mechanism. We position this work as a feasibility study of entirely latent space inter-agent communication, and our results highlight its potential, offering valuable insights for future research. Our code is available at https://github.com/XiaoDu-flying/Interlat.

"We do not have organs of communication. Our brains can display our thoughts to the outside world, thereby achieving communication."
— Cixin Liu, *The Dark Forest*.

1 Introduction

Large language model (LLM)-based agentic systems have emerged as a promising paradigm for solving complex tasks by orchestrating multiple agents through natural language communication (Wang et al., 2025, 2024; Zhang et al., 2024b; Tran et al., 2025). Despite its human readability, natural language imposes fundamental constraints on inter-agent communication. To communicate, an agent must compress its rich, high-dimensional internal states into a sequence of discrete tokens, typically exposing only a single linear message (*i.e.,* a chain of thought (CoT) (Wei et al., 2022) plan). This downsampling not only discards alternative reasoning paths, but also incurs substantial redundancy, as much of the generated text serves linguistic coherence rather than task-relevant information (Zhang et al., 2024a). As a result, language-based communication can be ambiguous and lossy, which has been identified as a major source of coordination failures in multi-agent systems (Chen et al., 2025; Cemri et al., 2025).

To move beyond language space, we explore the direct transmission of internal representations for more precise and information-preserving communication. In multi-agent settings, we refer to this as ***latent communication***. While direct sharing is challenging for humans, which is often depicted in fictions (Liu, 2008), *i.e., telepathy*, LLM-based agents naturally perform most of their computation in latent space and produce rich hidden states throughout generation, which can be extracted to support direct, expressive communication. Previous hidden-state-based communication methods either rely on one-shot activation grafting (Ramesh and Li, 2025) or remain coupled to language trajectories (Tang et al., 2025), and typically require ad-hoc layer selection, adding tuning overhead.

In this work, we propose Interlat, a novel framework that realizes this vision by enabling

[†]Equal Contribution.
[*]Work done during an internship at Alibaba Group.
[✉]Corresponding Authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27106–27129
July 2-7, 2026

inter-agent communication entirely in latent space. Rather than transmitting discrete tokens decoded by the language head, Interlat directly transmits the temporally aligned last-layer hidden states corresponding to an agent's generated message, treating them as a representation of its thoughts. Under this formulation, we design a supervised objective that explicitly encourages the interpretation and utilization of task-relevant latent information, with a simple but effective stochastic token–latent mixing curriculum to stabilize training. To overcome the rigidity of full-trajectory message communication while preserving information integrity, we further train a separate reasoning process that autoregressively generates compact latent messages with a controllable number of generation steps directly in latent space, without decoding to language space tokens. This allows Interlat to compress long reasoning trajectories into concise latent prefixes, substantially improving efficiency while retaining task-critical information for downstream execution.

Experimentally, we focus on a two-agent sender-receiver scenario, which is the building block of various multi-agent systems. To reduce confounding factors, we intentionally exclude components such as tool use, retrieval, or multi-round debate. Analysis reveals that agents utilizing latent communication exhibit more exploratory behavior patterns that lead to higher success rates by leveraging task-relevant latent information rather than superficial pattern matching. Moreover, we demonstrate that latent messages can be compressed to as few as **8 tokens** while maintaining competitive performance, achieving up to a $24\times$ reduction in communication latency. Further analysis of the output probability distribution after compression reveals how task-critical information is effectively preserved.

2 Related Work

Latent Reasoning in LLMs. Recent research has begun shifting reasoning processes from the discrete language space to continuous latent representations, bypassing the bandwidth and efficiency limits of text (≈ 15 bits/token vs. ≈ 40k bits/hidden-state) (Zhu et al., 2025b). To expand computation during inference, (Goyal et al., 2023) introduces pause tokens, while (Pfau et al., 2024) employs filler tokens to scaffold intermediate reasoning. Beyond token scheduling, (Liu et al., 2024) proposes a latent coprocessor that modifies the transformer KV cache. Other work (Hao et al., 2024; Shen et al., 2025; Cheng and Van Durme, 2024) enables multi-path parallel reasoning by feeding the last hidden state back as the next input embedding. Modular frameworks (Bae et al., 2024; Gao et al., 2024; Geiping et al., 2025) decouple encoding, latent reasoning, and decoding. Building upon these insights, we shift focus from *single-model* latent reasoning to *inter-agent* communication entirely in latent space.

Multi-agent Communication. LLM-based agent systems typically orchestrate in natural language (Zhu et al., 2025a; Wang et al., 2024), which can introduce ambiguity and computational overhead (Zhang et al., 2024a; Yu et al., 2024; Cemri et al., 2025; Chen et al., 2025). Emergent communication studies (Lazaridou et al., 2016, 2018; Tucker et al., 2021, 2022) show that non-linguistic protocols can emerge, yet depend on channels learned from scratch and disconnected from internal reasoning. Recent work explores richer forms: (Pham et al., 2023) transmits probability-weighted tokenizer embeddings; (Ramesh and Li, 2025) blends hidden activations between agents; and (Tang et al., 2025) communicates per-token latent deltas tied to language trajectories.(Zheng et al., 2025) infers latent "thoughts" from hidden states via an autoencoder and injects them as prefixes using recovered dependencies. Unlike these works, our method directly transmits temporally aligned sequences of last hidden states and further compresses them to enable efficient, language-free communication. Interlat preserves agent autonomy by requiring no parameter sharing (Christianos et al., 2021), memory coupling (Salemi et al., 2025), or cache synchronization (Fu et al., 2025; Zou et al., 2025), while expanding the effective communication bandwidth.

3 Interlat

In this section, we formalize how to extract an agent's states as the representation of its "thought" for inter-agent latent space communication. Let $x = (x_1, \ldots, x_T)$ denote a sequence consisting of a prompt $x_{1:m}$ and a completion sequence $y = (y_1, \ldots, y_L)$ such that $y_\ell = x_{m+\ell}$ and $L = T - m$. For each decoding step $\ell = 1, \ldots, L$, define:

$$h_\ell = \text{Transformer}(x_{\le m+\ell-1})_{m+\ell-1},$$
$$H = [\, h_1,\, h_2,\, \ldots,\, h_L \,],$$

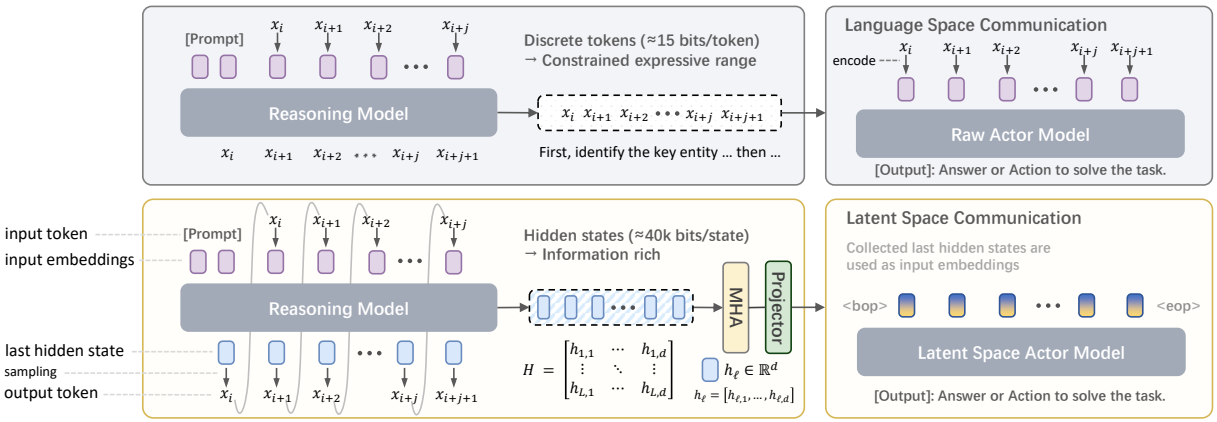


Figure 1: A comparison of Interlat with language-space communication. In language space, an agent transmits a discrete token sequence $[x_i, x_{i+1}, \ldots, x_{i+j+1}]$ (*e.g.,* a CoT plan) to another. In Interlat, the model leverages its last hidden states as a representation of its internal "thought", processed by a communication adapter, and then transmits them directly to the other agent, enabling communication entirely in latent space with higher expressive capacity.

where $h_\ell \in \mathbb{R}^d$ is the last-layer hidden state immediately before predicting y_ℓ (*i.e.,* at position $m+\ell-1$ in the full sequence). $H \in \mathbb{R}^{L\times d}$ collects these last hidden states for the completion region.

3.1 Latent Communication

Interlat removes natural language constraints by letting agents transmit their thoughts by directly passing the collected last hidden states, which we termed latent communication. As shown in Figure 1, this transmission occurs at the end of an agent's message generation process. Special tokens, x_i = <bop> and x_j = <eop>, are added to mark the beginning and the end of the latent communications. Consider an agent $\mathcal{M}_i$ solving a task $\mathcal{T} = \{x_1, \ldots, x_m\}$. Upon receiving a latent communication $H = \{h_1, h_2, \ldots, h_L\}$ from another agent, it forms its input embedding as:

$$E = [\, e(x_1), \ldots, e(x_i), h_1, h_2, \ldots, h_L, e(x_j) \,],$$

where $e(\cdot)$ is the token embedding. This inference process is analogous to language space multi-agent systems, except that it directly feeds hidden states between agents. The latent communications are processed by a trainable light-weight self-attention and a projection layer as a *communication adapter* for rescaling and interpretation. For brevity, we may refer to *latent communication* as *latents* where unambiguous.

3.2 Training Procedure

In this work, we consider two agents: a ***reasoning*** agent as a sender that produces a task-specific plan together with its last-layer hidden states, and an ***actor*** agent as a receiver that consumes this communication to generate actions to solve tasks. This two-agent setting can serve as a fundamental building block for more complex multi-agent systems.

Let Y_t denote the next token at supervised position $t \in S$, where S indexes the actor's output tokens corresponding to the ground-truth response, and C_t the decoder prefix up to position t. We encourage the actor to utilize H by maximizing a supervised fine-tuning objective regularized by conditional distributional separation:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{S}}\, \mathcal{L}_{\text{sep}} + \lambda_{\text{A}}\, \mathcal{L}_{\text{align}},$$

where $\lambda_{\text{S}}, \lambda_{\text{A}} > 0$, and $\mathcal{L}_{\text{task}}$ is the standard cross-entropy loss that ensures the model produces accurate and coherent responses for the given task.

Conditional thought separation. We compare the conditional output distributions p_θ induced by matched latent communications H and mismatched latents $\tilde{H}$ (*i.e.,* latent communications sampled from a different task). Specifically, we minimize a weighted Jensen–Shannon divergence (Lin, 2002):

$$\mathcal{L}_{\text{sep}} = -\frac{1}{|S|} \sum_{t\in S} \text{JS}\big(p_\theta(\cdot\,|\,C_t, H),\, p_\theta(\cdot\,|\,C_t, \tilde{H})\big).$$

This objective separates matched from mismatched conditional distributions, providing a robust training signal that encourages the actor to attend to and leverage task-relevant latent information.

Plan-aligned regulation. While maximizing separation encourages sensitivity to H, it also introduces a failure mode where the model may exploit the objective by shifting probability mass toward idiosyncratic tokens that increase divergence while harming task utility. To mitigate this, we regularize

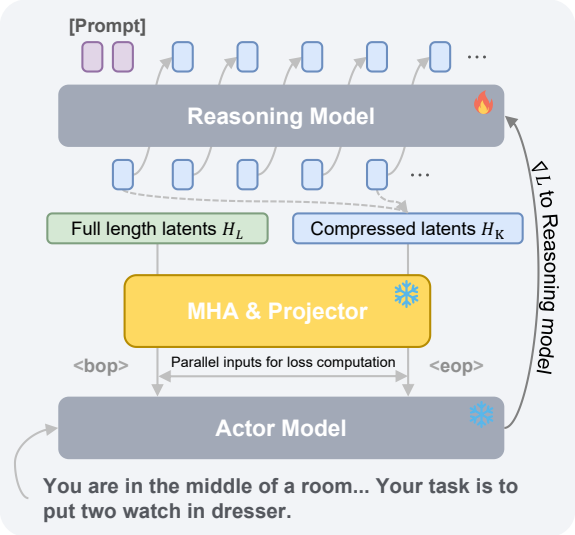


Figure 2: Training the reasoning model with frozen-actor supervision.

predictions conditioned on H using those conditioned on the corresponding language-space plan P, generated by the same instruction-tuned model during the autoregressive generation of H. Let $p_{\text{plan}}(\cdot \mid C_t, P)$ denote the distribution when only the plan is provided. For brevity, we omit explicit conditioning on C_t in the following.

$$\mathcal{L}_{\text{align}} = \frac{\beta}{|S|} \sum_{t \in S} \mathrm{KL}\big(p_\theta(\cdot \mid H) \,\|\, p_{\text{plan}}(\cdot \mid P)\big) + \frac{\alpha}{|S|} \sum_{t \in S} \big(1 - \cos\big(\mathbf{z}_\theta(H), \mathbf{z}_{\text{plan}}(P)\big)\big),$$

where $\mathbf{z}_\theta$ and $\mathbf{z}_{\text{plan}}$ denote the corresponding normalized logit. All divergences and cosine similarities are computed at supervised positions, with probabilities obtained from the softmax of logits.

Curriculum Learning. Learning to interpret latents from scratch is unstable. We thus adopt a token-to-latent curriculum that stochastically replaces early communication positions with their corresponding plan token embeddings. Concretely, we sample a replacement rate $r \in \{0, 0.1, \ldots, 1.0\}$ and form a mixed communication

$$H^{(r)} = \underbrace{e_1, \ldots, e_{\lfloor r \cdot L \rfloor}}_{\text{token embeddings}} \oplus \underbrace{h_{\lfloor r \cdot L \rfloor + 1}, \ldots, h_L}_{\text{latent states}}.$$

This method enhances training efficiency while achieving strong model performance.

4 Information Compression

While full-length latents $H_L \in \mathbb{R}^{L \times d}$ are highly expressive, their temporal length (often dozens to hundreds of steps) introduces substantial communication latency. Unlike natural-language tokens, whose semantics are discrete and inherently sequential, latent states are continuous and over-parameterized relative to task requirements, suggesting that much of the temporal structure is redundant. Our goal is therefore to learn an *information-preserving bottleneck* that compresses latents while retaining their utility for downstream agents.

Compression via Latent-Space Reasoning. To this end, we train a separate reasoning model M_ϕ to generate compact latent messages $H_K \in \mathbb{R}^{K \times d}$ with $K \ll L$, while keeping the actor model and its communication adapter frozen. Rather than truncating or subsampling H_L, M_ϕ performs autoregressive reasoning entirely in latent space by feeding its last hidden state back as the next input embedding through a lightweight projection:

$$\langle M_\phi(E_i) \to h_i,\ E_{i+1} = E_i \oplus \mathrm{Proj}(h_i) \rangle.$$

This design enables an end-to-end differentiable latent generation loop without decoding to tokens, and isolates compression from changes in the actor's behavior. During training, only the parameters of M_ϕ are updated, ensuring that compression is learned purely by adapting the latent message itself.

Training Objective. We train the compression model using a composite objective:

$$\mathcal{L}_{\text{compress}} = \lambda_{\text{task}} \mathcal{L}_{\text{task}} + \lambda_{\text{pref}} \mathcal{L}_{\text{pref}} + \lambda_{\text{geom}} \mathcal{L}_{\text{geom}},$$

which jointly addresses the main failure modes of aggressive compression. The task loss $\mathcal{L}_{\text{task}}$ is a cross-entropy on the frozen actor's predictions conditioned on H_K, ensuring downstream task utility.

Motivated by the non-uniform information density of token-level language communication (Shannon, 1951; Zhang et al., 2024a), we introduce an uncertainty-weighted agreement loss that selectively aligns the actor's behavior under compressed and full-length communications. Let $p_t^{(A)}$, $p_t^{(D)}$, and $p_t^{(B)}$ denote the actor's output distributions at position t when conditioned on H_K, H_L, and no latents, respectively. We define per-token weights $w_t \propto \max\big(\mathcal{H}(p_t^{(B)}) - \mathcal{H}(p_t^{(D)}), 0\big)$. where $\mathcal{H}(\cdot)$ denotes entropy, and compute $\mathcal{L}_{\text{pref}} = \sum_{t \in S} w_t \,\mathrm{KL}\Big(p_t^{(D)} \,\|\, p_t^{(A)}\Big)$. This objective emphasizes positions where latents meaningfully reduces predictive uncertainty, while avoiding over-regularization where latents are uninformative.

Method	Qwen2.5-7B-Base				Qwen2.5-0.5B-Base				LLaMA3.1-8B			
	Seen	Steps	Unseen	Steps	Seen	Steps	Unseen	Steps	Seen	Steps	Unseen	Steps
Interlat												
Ours	**70.48**	9.41/12.54	**65.42**	9.86/13.37	**61.19**	10.55/14.22	**57.46**	9.38/13.90	**70.71**	8.02/12.58	**70.90**	8.21/12.96
Text	64.29	8.76/12.77	62.44	9.79/13.63	54.52	9.50/14.28	47.26	9.70/15.13	62.86	7.91/12.94	60.82	8.14/13.21
No-Comm	62.14	10.19/13.90	62.19	10.23/13.92	50.48	8.23/14.06	44.03	9.10/15.20	63.57	8.35/12.59	58.40	9.47/13.85
Baselines												
CoT (full)	67.14	8.15/12.04	64.93	9.02/12.87	57.86	8.30/13.23	50.75	8.94/14.39	69.35	7.62/12.32	70.82	7.88/12.47
No-CoT	65.71	8.23/12.27	62.69	9.15/13.20	57.14	8.96/13.69	50.25	9.80/14.87	67.18	7.85/12.61	70.34	8.02/12.88
Variants												
CrossTask	61.43	8.42/12.89	61.94	9.51/13.50	53.57	9.40/14.32	47.01	10.06/15.33	65.00	8.05/12.24	63.43	9.86/13.57
Noised												
CovNoise-0.5×	64.29	8.54/12.63	60.95	8.71/13.12	53.33	8.80/14.03	46.77	9.64/15.16	64.29	8.10/12.50	65.68	9.47/12.77
CovNoise-1.0×	63.81	8.66/12.76	63.68	8.72/12.82	53.10	8.96/14.14	44.53	9.68/15.40	58.57	7.80/12.80	64.93	9.66/13.28
WhiteNoise	61.90	8.65/12.97	61.19	9.32/13.46	57.38	8.00/13.11	57.21	9.18/13.81	61.43	8.01/12.64	64.93	9.52/13.19
CovGauss-0μ	60.00	8.79/13.27	61.94	9.59/13.55	13.81	11.25/18.79	13.18	12.93/19.07	57.86	8.04/13.08	66.42	9.51/13.03
CovGauss-μ	65.71	8.58/12.50	64.93	8.63/12.62	44.52	9.21/15.20	34.33	10.19/16.63	60.71	7.69/12.53	64.93	8.85/12.76
RandomRot	57.86	8.43/13.31	63.68	9.37/13.23	59.05	8.24/13.06	51.99	9.12/14.34	57.86	7.67/12.86	63.44	9.04/13.25
Cross family												
Qwen2LLaMA	70.95	8.47/12.01	71.39	9.21/13.05	–	–	–	–	–	–	–	–

Table 1: Performance of different methods and variants on seen and unseen tasks of Alfworld under three model backbones. Higher success rates indicate stronger inter-agent collaboration and task-solving ability. "Steps" reports average steps on successful tasks and average steps over all tasks. Best in **bold**, second-best underlined.

Method	Overall	Level-3	Level-4	Level-5
Ours	36.88	40.08	27.45	**15.80**
Text	34.35	37.60	26.30	14.20
No-Comm	33.27	36.40	26.20	13.10
CoT (full)	**38.35**	**45.65**	**31.19**	15.05
No-CoT	36.25	40.10	26.80	14.80

Table 2: Accuracy (%) on the MATH benchmark. While CoT benefits from linguistic constraints on simpler tasks, Interlat outperforms the strong CoT baseline on the most challenging Level-5 tasks.

Finally, to prevent representational drift under strong compression, we apply a latent geometry alignment loss. Let $Z_k^{(A)}$ and $Z_k^{(D)}$ denote the actor-side latent features induced by H_K and H_L after adapter processing and length alignment, and define their step-averaged directions $\bar{z}^{(A)}$ and $\bar{z}^{(D)}$. We enforce $\mathcal{L}_{\text{geom}} = 1 - \cos\big(\bar{z}^{(A)}, \bar{z}^{(D)}\big)$, which preserves the global semantic orientation of the original communication in the compressed latent space. Together, these objectives encourage M_ϕ to learn an information-preserving bottleneck that discards redundant temporal structure while retaining task-critical functional and geometric properties. Full derivations are provided in Appendix B.

and trajectory data from (Song et al., 2024)

5 Experiments

Implementation Details. We evaluate our approach on Alfworld (Shridhar et al., 2020), and MATH (Hendrycks et al., 2021). For Alfworld, training is conducted using data from (Song et al., 2024). Qwen2.5-7B/0.5B-Base (Yang et al., 2024) and LLaMA3.1-8B-Base (Dubey et al., 2024) are employed as actor agents to isolate benefits from instruction-tuning priors. CoT plans and compression-free latents are generated by their instruction-tuned counterparts. We use base models as reasoning models in the information compression experiments. Alfworld episodes are capped at 20 steps; unfinished episodes are failures. All models are trained using bfloat16, FlashAttention-2 (Dao, 2023), and DeepSpeed (Rajbhandari et al., 2020). For the actor model, we optimize using AdamW (Loshchilov and Hutter, 2017) with a learning rate of 1×10^{-5}, a global batch size of 16, and a 3% linear warmup. We fix $\lambda_{\text{task}} = 1$ and dynamically anneal the regularization coefficients during training ($\lambda_{\text{sep}} \in [0.1, 1.0]$, $\lambda_{\text{align}} \in [0.1, 0.2]$). Negative samples for $\mathcal{L}_{\text{sep}}$ are constructed using latents from different tasks within the same batch. For information compression, we train the reasoning agent with a frozen actor, utilizing a learning rate of 5×10^{-5} and unit weights for all three compression objectives. We select the best models based on a 5% validation split. All reported results are averaged over three independent runs.

Baselines and variants in Interlat. We study the feasibility of Interlat and compare against two baselines: **CoT (full)** uses complete CoT plans from instruction-tuned models for full-parameter supervised fine-tuning; **No-CoT** directly predicts final answers without any plan.

We further evaluate *variants* of our method: **Text** replaces latent messages with the corresponding CoT plan; **No-Comm** removes communication entirely; **CrossTask** replaces the current task's latents with one sampled from a different task. **Noised** adds structured or unstructured perturbations to

Figure 3: Training dynamics of the separation loss: an initial plateau near 0.69 indicates no separation between matched/mismatched latents, followed by a sharp drop after $\sim$ 2.2k steps, marking the model's "aha" moment in exploiting task-relevant latent information.

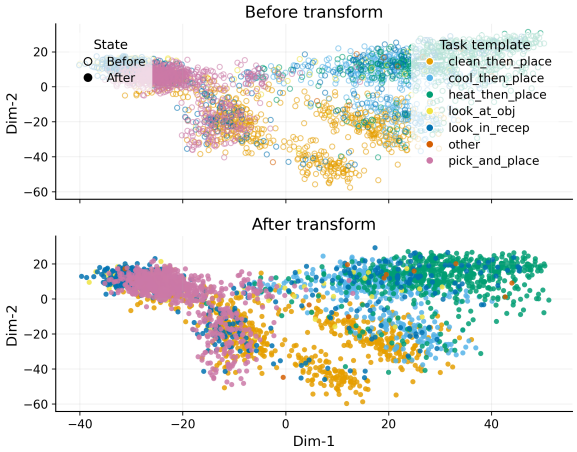


Figure 4: PCA visualization of latent communications, showing distinguishable task-specific structure in latent space both before and after the communication adapter.

H; **CovGauss** and **RandomRot** preserve mean or covariance statistics while destroying higher-order structure. **Qwen2LLaMA** uses latents from Qwen2.5-7B to train LLaMA3.1-8B model. See Appendix B.1 for detailed implementation setups.

5.1 Main Results

Table 1 presents a comprehensive comparison of the Interlat framework against baselines. Latent communication improves agents' task-solving performance, as evidenced by gains over both fine-tuned single-agent baselines and agents trained to communicate in natural language. We highlight several key observations below.

Latent Communication Prompts Exploration. Beyond improvements in success rates, latent communication enables agents to execute longer yet more successful trajectories. By leveraging multiple plausible reasoning paths encoded in latents from other agents, the actor naturally exhibits more thorough exploratory behavior, even without explicit exploration training. Importantly, this increased trajectory length correlates with higher success rates rather than degraded efficiency, indicating informed exploration instead of random wandering. This behavior suggests a stronger environmental understanding enabled by latent communication, where parallel hypotheses are preserved and gradually resolved during action execution. This pattern is analyzed in detail in Appendix D.

Semantics and Learning Dynamics. To assess whether the actor genuinely exploits latent information rather than superficial patterns, we conduct structured perturbation experiments. Replacing task-matched latents with cross-task latents leads to a substantial performance drop, indicating that the actor relies on task-specific reasoning content encoded in the latents. Performance degrades further under covariance-matched Gaussian surrogates or random orthogonal rotations, which preserve first- and second-order statistics while destroying higher-order structure, supporting the interpretation that the actor is sensitive to meaningful latent geometry rather than low-order moments alone. Additive and white noise perturbations similarly impair performance, further indicating reliance on structured internal information instead of noise-robust heuristics. Experiments involving *cross-family* latent inputs: feeding Qwen-derived latents to train an LLaMA actor, yield even stronger performance gains. Since these model families exhibit distinct latent manifolds, the improvement cannot be attributed to superficial architectural compatibility. Instead, it suggests latent-level inter-agent understanding that transfers across heterogeneous representations. This observation aligns with findings in language-space agentic systems, where heterogeneous LLM agents often outperform homogeneous ensembles due to complementary inductive biases and reduced error correlations (Shinn et al., 2023; Wu et al., 2024).To corroborate these findings qualitatively, Figure 4 and Appendix G visualize clear semantic clustering before and after processing by the communication adapter, confirming effective semantic alignment for downstream use.

Training dynamics further reveal how the actor learns to interpret latent communication. As shown in Figure 3, the separation loss remains near $\ln 2$ for approximately the first 2k steps, indicating no effective distinction between matched and mismatched

Ratio	Seen	Unseen	Time
Untrained (An instruction-tuned Model)			
Full	$\underline{70.48}_{\pm 1.01}$	$\underline{65.42}_{\pm 0.87}$	9.19s
90%	$68.57_{\pm 1.63}$	$\mathbf{67.16}_{\pm 1.97}$	-
80%	$68.10_{\pm 1.83}$	$61.69_{\pm 1.43}$	-
70%	$67.14_{\pm 1.82}$	$63.43_{\pm 2.24}$	-
60%	$66.43_{\pm 1.63}$	$59.20_{\pm 3.69}$	-
50%	$\mathbf{72.14}_{\pm 1.48}$	$61.19_{\pm 2.84}$	-
40%	$66.90_{\pm 2.31}$	$59.95_{\pm 2.64}$	-
30%	$65.95_{\pm 2.12}$	$62.19_{\pm 1.58}$	-
20%	$67.86_{\pm 3.23}$	$61.44_{\pm 1.58}$	-
10%	$67.86_{\pm 2.12}$	$62.44_{\pm 2.64}$	-
5%	$64.29_{\pm 1.12}$	$60.95_{\pm 1.35}$	-
0%	$62.14_{\pm 2.01}$	$62.14_{\pm 2.32}$	-

Ratio	Seen	Unseen	Time
Untrained (An instruction-tuned Model)			
128L	$64.55_{\pm 2.26}$	$60.25_{\pm 2.06}$	3.55s
64L	$66.23_{\pm 1.95}$	$61.53_{\pm 4.32}$	1.83s
32L	$63.57_{\pm 2.01}$	$60.18_{\pm 3.58}$	1.03s
16L	$64.29_{\pm 1.34}$	$60.00_{\pm 3.01}$	0.62s
8L	$64.00_{\pm 2.18}$	$57.46_{\pm 2.69}$	0.39s
Trained			
128L	$\mathbf{68.10}_{\pm 1.93}$	$\mathbf{62.94}_{\pm 2.03}$	2.25s
64L	$\underline{67.14}_{\pm 1.56}$	$\underline{61.94}_{\pm 2.13}$	1.16s
32L	$66.90_{\pm 1.46}$	$61.94_{\pm 2.56}$	0.60s
16L	$66.43_{\pm 2.05}$	$61.69_{\pm 2.56}$	0.33s
8L	$66.43_{\pm 1.22}$	$60.45_{\pm 2.23}$	0.20s

Table 3: Compression results on Alfworld. **Left:** training-free sweep over retained ratio R. **Right:** varying latent length with untrained and trained models. Time denotes end-to-end latency (s) of the message generation process. Best results are **bold**, second-best are underlined. **See Appendix E for results on LLaMA model.**

messages. It then drops sharply, marking an "aha" moment in which the actor begins to exploit and leverage task-relevant latents, consistent with the intended effect of the separation objective.

Generalization to Symbolic Reasoning. To assess generalization beyond interactive settings, we evaluate Interlat on the MATH benchmark. While prior work on latent-space reasoning (Hao et al., 2024; Shen et al., 2025; Ramesh and Li, 2025) often reports degraded performance relative to CoT supervision, Table 2 reveals an intriguing inversion: although Interlat slightly underperforms on simpler problems, it surpasses the CoT baseline on the most challenging Level 5 tasks. We attribute this to the duality of linguistic constraints. For lower-complexity problems, the strict linearization of natural language acts as a beneficial regularizer, efficiently pruning the search space. However, for high-complexity tasks, this forced discretization causes a "premature collapse" of the reasoning distribution. In contrast, Interlat maintains a superposition of parallel hypotheses in its continuous representations. This capability allows the model to effectively conduct a broader search in latent space that is inaccessible to linear text decoding.

5.2 Compression Analysis

Compression Performance. Theoretically, due to their substantially higher expressive capacity, latent communications can encode rich information in far fewer positions. To quantify this compression capacity, we consider two settings. **i) Untrained:** We directly use Qwen2.5-7B-Instruct to generate full-length latents for actor training, and then truncate them to shorter lengths. This setting evaluates the empirical compressibility of raw latents. **ii) Trained:** We use a compression-trained Qwen2.5-7B-Base reasoning model. Results on the LLaMA model are provided in Appendix E.

As shown in Table 3, naive truncation performs best at moderate compression (50%) but degrades under more aggressive shortening, revealing the limits of untrained compression. In contrast, compression training enables consistently higher and more stable success rates across compressed latent ranging from 8 to 128 steps (around 1.8% to 28.8% of the full sequence), indicating that the reasoning model learns an information-preserving pattern that discards temporal redundancy while retaining task-relevant semantics. Furthermore, compression substantially improves efficiency, reducing end-to-end latency from 9.19 s to 0.39 s with 8-step latents (nearly 24× speed-up), and further to 0.20 s with a lightweight bridge module by largely eliminating decode–re-encode overhead.

Why compression is effective. To understand why compression preserves performance, we analyze its effect on the actor's predictive uncertainty. We sweep the communication rate $R \in [0, 1]$ and measure the task-averaged relative change in cross-entropy (CE), $\Delta\mathrm{CE}\%(R) = 100 \times \frac{\mathrm{CE}_{\mathrm{comp}}(R) - \mathrm{CE}_{\mathrm{full}}}{\mathrm{CE}_{\mathrm{full}}}$. As shown in Figure 6, $\Delta\mathrm{CE}\%$ decreases monotonically with increasing R and plateaus between roughly 30% and 75%, aligning with the range of strongest empirical performance. Across all rates, learned compression consistently yields lower CE than training-free truncation, with a maximum gap of approximately 11 percentage points, indicating better preservation of predictive

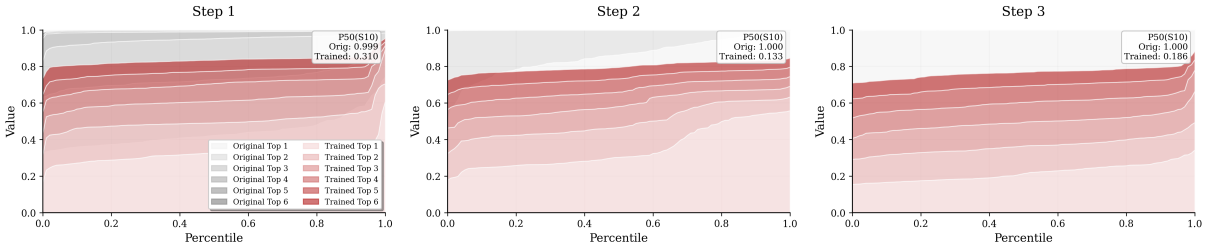


Figure 5: Parallelism analysis of latent communication over the first three steps. Trained latents (red) maintain stable Top-k gaps and lower $P_{50}(S_{10})$, indicating stronger parallelism, while untrained latents (gray) collapse toward Top-1. See Appendix F for extended results.

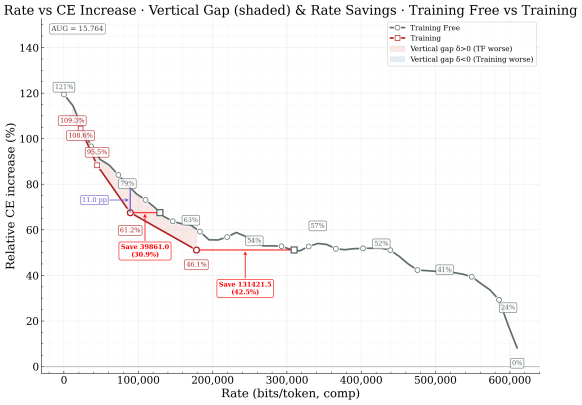


Figure 6: Task-averaged relative change ΔCE and relative savings before and after compression training.

confidence under reduced communication.

We further examine how information is preserved under compression via the actor's output distributions. Following (Hao et al., 2024), broader probability mass is interpreted as indicating that the model maintains more plausible alternatives. Figure 5 analyzes the latent steps of the reasoning agent by plotting the cumulative probability mass of the top-6 tokens across communication percentiles, with probabilities normalized over the top-10 tokens for comparability. Latents produced by the trained reasoning agent exhibit stable gaps between successive top-6 curves across steps, whereas untrained compression shows rapid concentration toward top-ranked tokens. We quantify this behavior using a head-coverage statistic, $\mathrm{P50}(S_{10})$, defined as the median cumulative probability mass of the top-10 tokens, which is consistently lower for the trained model and indicates broader support over plausible alternatives.

Together, these observations suggest that learned compression preserves diverse hypotheses across multiple reasoning steps, avoiding collapse into a single trajectory and thereby retaining richer information for downstream decision making.

5.3 Ablation Studies

Table 4 presents a systematic ablation of the training components. For the actor model, removing curriculum learning forces the model to interpret latents from scratch, leading to extremely unstable training dynamics and severely degraded comprehension (Appendix, Figure 7). Removing the separation loss induces shortcut behavior; the model learns to ignore the latent communication and rely only on the textual task prompt, causing performance to regress toward the single agent baseline. Removing the communication adapter causes the largest drop; despite generating fluent and coherent responses, the model fails to complete tasks, underscoring the adapter's role in bridging the agents' latent spaces and enabling interpretation of latents.

For the reasoning model, which is trained to generate compressed latents, we ablated its three core loss functions with compressed target length $K = 128$. The most critical component is the direction alignment loss ($\mathcal{L}_{\text{geom}}$). This highlights the importance of maintaining geometric consistency between the compressed latents and the uncompressed ones. The agreement loss ($\mathcal{L}_{\text{pref}}$) is also vital, as removing it significantly impairs the model's ability to produce latents that elicit the correct behavior from the actor. Removing the cross-entropy loss ($\mathcal{L}_{\text{task}}$) degrades performance on seen tasks but slightly improves unseen performance, suggesting a minor trade-off between in-distribution optimization and generalization. We leave a deeper investigation into this trade-off to future work.

5.4 Generalization to Complex Topologies.

To directly verify that Interlat scales beyond simple pairwise communication, we evaluate it on 3-agent topologies using the Qwen2.5-0.5B-Base model on the ALFWorld benchmark. We ensure the entire multi-agent system can be jointly trained end-to-end by having all non-output sender agents gen-

Actor	Seen	Unseen
Ours Full	**70.48**$_{\pm 1.01}$	**65.42**$_{\pm 0.87}$
w/o curri	33.10$_{\pm 2.97}$	20.65$_{\pm 2.15}$
w/o $\mathcal{L}_{\text{sep}}$	<u>58.81</u>$_{\pm 1.41}$	<u>60.70</u>$_{\pm 5.50}$
w/o $\mathcal{L}_{\text{align}}$	56.90$_{\pm 1.41}$	53.98$_{\pm 3.35}$
w/o adapter	4.05$_{\pm 1.70}$	4.48$_{\pm 1.31}$

Reasoning	Seen	Unseen
Ours Full	**68.10**$_{\pm 1.93}$	<u>62.94</u>$_{\pm 2.03}$
w/o $\mathcal{L}_{\text{task}}$	65.71$_{\pm 1.43}$	**63.18**$_{\pm 3.47}$
w/o $\mathcal{L}_{\text{pref}}$	64.76$_{\pm 2.97}$	60.20$_{\pm 3.13}$
w/o $\mathcal{L}_{\text{geom}}$	64.05$_{\pm 3.55}$	59.45$_{\pm 3.01}$

Table 4: Ablation of training components. **Left:** For the actor, the communication adapter and curriculum learning are foundational; removing the adapter leads to near-zero success rates. **Right:** For the reasoning model, the geometry alignment loss ($\mathcal{L}_{\text{geom}}$) proves to be the most critical objective for maintaining performance under compression. **See Appendix D for full results and analysis.**

Communication	Topology (N)	Seen (%)	Unseen (%)	Structural Description
Text (CoT)	text-chain-2	57.86	50.75	Sequential: 1 Sender → 1 Actor
	text-chain-3	58.34	51.89	Sequential: Sender → Sender → Actor
	text-tree-3	61.19	56.75	Parallel: 2 Senders → Actor (Merged by agent)
Latent (Ours)	chain-2 (Base)	61.19	57.46	Sequential: 1 Sender → 1 Actor
	chain-3	62.14	59.44	Sequential: Sender → Sender → Actor
	tree-3	**62.85**	**60.77**	Parallel: 2 Senders → Actor (Adapter merged)

Table 5: Performance of complex agent topologies on ALFWorld using Qwen2.5-0.5B-Base. We compare natural language CoT communication ("text-") with Interlat ("chain-" and "tree-") across varying agent counts (N). Latent communication consistently yields higher success rates, with the parallel tree topology performing the best.

erate fixed-length ($N = 32$) messages entirely in latent space. Specifically, we investigate two distinct configurations:

Sequential Chain Topology. In this sequential structure (Agent 1 → Agent 2 → Actor), a *Strategist* agent first generates a high-level latent plan capturing the task goals. A *Compiler* agent then refines this received plan into a more actionable representation before passing it to the final Actor.

Parallel Tree Topology. This parallel structure ([Agent 1 + Agent 2] → Actor) utilizes an *Explorer* agent to generate diverse candidate reasoning paths and a *Critic* agent to produce verification constraints simultaneously. To dynamically fuse these parallel messages, we extend the communication adapter with an additional learned projector that maps the aggregated latents from multiple senders into a unified representation for the Actor.

As shown in Table 5, both 3-agent latent configurations consistently outperform their natural language CoT counterparts with identical agent structures (e.g., 62.85% vs. 61.19% on seen tasks for the parallel tree). Moreover, both configurations yield solid improvements over the original 2-agent Interlat baseline (61.19%). These results demonstrate that the performance gains stem intrinsically from the rich expressive capacity and information preservation of the latent space, rather than a mere increase in the number of agents. Notably, the superior performance of the `tree-3` configuration confirms that Interlat effectively supports parallel multi-agent reasoning, allowing downstream actors to synthesize heterogeneous latent perspectives without requiring pair-specific training.

6 Conclusion

In this work, we introduced Interlat, a paradigm that enables inter-agent communication entirely in latent space. Across experiments, our results show that directly transmitting and reasoning over latent states improves task performance and achieves substantially higher communication efficiency, even demonstrating compatibility across heterogeneous models. Beyond full-length latent exchange, we show that latent communication can be aggressively compressed through latent-space reasoning, forming a compact, task-preserving representation that retains parallel hypotheses while discarding redundant structure. Together, these findings suggest that communication need not be bound to language tokens, highlighting latent states as a viable, efficient, and generalizable medium for next-generation multi-agent systems.

7 Limitations

Our study has explored how inter-agent communication can be realized entirely in latent space and demonstrated its potential benefits in terms of performance and efficiency. However, several important limitations should be considered when interpreting our results and applying this paradigm more broadly.

First, our experiments mainly focus on a controlled two-agent setting, evaluated on one embodied interactive benchmark (ALFWorld) and one non-interactive symbolic reasoning benchmark (MATH). While these benchmarks jointly cover both interactive planning and single-turn reasoning scenarios, they do not yet fully capture the diversity of real-world multi-agent systems, such as settings with larger teams, dynamic role assignments, long-horizon collaboration, or richer environments involving tool use and external memory. Extending latent communication to larger-scale and more heterogeneous agent ecosystems is a natural and important direction for future work.

Second, our approach assumes access to internal model representations, specifically last-layer hidden states, in order to enable latent communication. This assumption may not hold for closed-source or API-only models, where hidden states are inaccessible. As a result, the current formulation of Interlat is primarily applicable to open or inspectable models, and extending latent communication to restricted-access settings is an open challenge.

Finally, latent communication trades human interpretability for efficiency and expressive capacity. Unlike language-based communication, latent messages are not directly human-readable, which complicates debugging, monitoring, and failure analysis in complex systems. While this work primarily focuses on task performance and efficiency and provides preliminary interpretability analysis via PCA (Figure 4), developing principled tools to improve both the interpretability and controllability of latent communication remains an important direction for broader deployment.

Despite these limitations, we view this work as an initial yet concrete step toward understanding and enabling inter-agent communication beyond language. Our results provide empirical evidence that latent space communication can support effective coordination and can be aggressively compressed while preserving task utility, highlighting its potential as a complementary paradigm to language-based interaction in LLM-powered agent systems.

8 Ethical Considerations

No human participants, crowdsourcing, or personally identifiable information (PII) were involved in this research. All experiments were conducted within a simulated environment using standard dataset splits.

Our study focuses on inter-agent communication in latent space, utilizing the last hidden states and their compressed variants. A potential theoretical risk is that such latent communication could be exploited to circumvent language-based safety mechanisms. To mitigate this concern to the greatest extent possible, we neither trained on nor evaluated any harmful instructions, and no harmful actions occurred during our experiments. Furthermore, to promote transparency, we provide PCA-based visualizations of latent communication in Figure 4 the analyze internal probability distribution of compressed latent communications in Figure 5, offering a clearer understanding of the information being transmitted.

9 Acknowledgments

This research was partially supported by National Key R&D Program of China under Grant No. 2024YFF0907802, Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020011, and Alibaba Group through Alibaba Research Intern Program.

References

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. 2024. Relaxed recursive transformers: Effective parameter sharing with layer-wise lora. *arXiv preprint arXiv:2410.20672*.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, and 1 others. 2025. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, pages 1989–1998. PMLR.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Yihang Gao, Chuanyang Zheng, Enze Xie, Han Shi, Tianyang Hu, Yu Li, Michael K Ng, Zhenguo Li, and Zhaoqiang Liu. 2024. Algoformer: An efficient transformer framework with algorithmic structures. *arXiv preprint arXiv:2402.13572*.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2023. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*.

Jianhua Lin. 2002. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Cixin Liu. 2008. *The Dark Forest*. Chongqing Publishing House, Chongqing, China. English translation published by Tor Books, 2015.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Francesco Mezzadri. 2006. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*.

Jacob Pfau, William Merrill, and Samuel R Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

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

Alireza Salemi, Mihir Parmar, Palash Goyal, Yiwen Song, Jinsung Yoon, Hamed Zamani, Hamid Palangi, and Tomas Pfister. 2025. Llm-based multi-agent blackboard system for information discovery in data science. *arXiv preprint arXiv:2510.01285*.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. *arXiv preprint arXiv:2403.02502*.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. 2025. Augmenting multi-agent communication with state delta trajectory. *arXiv preprint arXiv:2506.19209*.

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

Mycal Tucker, Roger Levy, Julie A Shah, and Noga Zaslavsky. 2022. Trading off utility, informativeness, and complexity in emergent communication. *Advances in neural information processing systems*, 35:22214–22228.

Mycal Tucker, Huao Li, Siddharth Agrawal, Dana Hughes, Katia Sycara, Michael Lewis, and Julie A Shah. 2021. Emergent discrete communication in semantic spaces. *Advances in neural information processing systems*, 34:10574–10586.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Yingxu Wang, Siwei Liu, Jinyuan Fang, and Zaiqiao Meng. 2025. Evoagentx: An automated framework for evolving agentic workflows. *arXiv preprint arXiv:2507.03616*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2024. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024a. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2024b. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

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

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and 1 others. 2025a. Multiagentbench: Evaluating the collaboration and competition of llm agents. *arXiv preprint arXiv:2503.01935*.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, and 1 others. 2025b. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*.

Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, and 1 others. 2025. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*.

Appendix

The supplementary information accompanying the main paper provides additional data, explanations, and details.

A LLM usage

ChatGPT[1] was used purely with the language of the paper during the writing process, including spell-checking and paraphrasing the authors' original content, without suggesting new content. Any content generated with the assistant underwent meticulous manual review and subsequently received final approval from the authors.

B Compression Loss

Setup. After training an actor M_θ to consume latent communications, we freeze M_θ and train a reasoning model M_ϕ to *produce* compact, information-dense latent communications of length K that the frozen actor can still exploit. For an input instance with supervised token indices S (the teacher-forced window after the first user turn), let

$$H^{\text{gen}}_{1:K} \;=\; M_\phi(x) \quad \text{and} \quad H^{\text{full}}_{1:L} \;=\; M_{\text{ins}}(x), \tag{1}$$

denote respectively the *generated* latent communication from the trainable reasoning model and the full-length latent communication extracted from a fixed instruction-tuned model M_{ins}. A lightweight communication adapter $g(\cdot)$ (kept frozen) preprocesses the latent communication before concatenation with boundary tokens <bop>/<eop>. For brevity, we use $H_K \equiv H^{\text{gen}}_{1:K}$ and $H_L \equiv H^{\text{full}}_{1:L}$.

We define three actor-scored forward paths through the frozen actor M_θ given a prompt x: (i) **Path A (generated latents)**: $E^{(A)} = [\,e(x),\, e(\texttt{<bop>}),\, g(H_K),\, e(\texttt{<eop>})\,]$; (ii) **Path D (full-length latents)**: $E^{(D)} = [\,e(x),\, e(\texttt{<bop>}),\, g(H_L),\, e(\texttt{<eop>})\,]$; (iii) **Path B (no latents)**: $E^{(B)} = [\,e(x)\,]$. Let $\mathbf{z}_t^{(q)}$ be the frozen-actor logits at position $t \in S$ under path $q \in \{A, D, B\}$, and

$$p_t^{(q)} \;=\; \text{softmax}\big(\mathbf{z}_t^{(q)}/T\big) \tag{2}$$

be the corresponding token distributions with temperature $T \geq 1$ used for distillation. Unless stated otherwise, gradients do not flow into M_θ or $g(\cdot)$.

[1] https://chat.openai.com/

(1) Actor cross-entropy utility. We require the generated message to be *useful* for the frozen actor:

$$\underbrace{\mathcal{L}_{\text{task}} \;=\; \frac{1}{|S|}\sum_{t\in S}\big(-\log p_\theta(y_t \mid C_t, H_K)\big)}_{\text{(computed under Path A).}} \tag{3}$$

This term enforces that the compressed latents H_K still drive correct next-token predictions, directly penalizing information loss due to shortening ($K \ll L$). It prevents degenerate "over-compression" that would be efficient but useless to the actor. Practically, it anchors training on task utility, encouraging compression gain does not come at the cost of downstream performance.

(2) Uncertainty-weighted agreement. We further encourage *behavioral agreement* between using full-length latent communication (Path D) and generated compressed latent communication (Path A), with per-token weights that reflect how much any latent reduces uncertainty relative to the no-latent baseline (Path B). Let the entropies be

$$\underbrace{H^{(q)}(t) \;=\; -\sum_{v} p_t^{(q)}(v)\,\log p_t^{(q)}(v),.}_{q\in\{A,D,B\}} \tag{4}$$

Define raw weights $w_t^\star = \max\big(H^{(B)}(t) - H^{(D)}(t),\, 0\big)$ and optionally clip $w_t^\star$ to $[0, \tau]$ to suppress outliers. Normalize to unit mean:

$$w_t \;=\; \frac{w_t^\star}{\frac{1}{|S|}\sum_{u\in S} w_u^\star + \varepsilon}. \tag{5}$$

The agreement term is a temperature-scaled KL:

$$\begin{aligned}\mathcal{L}_{\text{pref}} &= \frac{1}{\sum_{t\in S} w_t}\sum_{t\in S} w_t\, T^2\, \text{KL}\Big(p_t^{(D)} \,\big\|\, p_t^{(A)}\Big) \\ &= \frac{T^2}{\sum_{t\in S} w_t}\sum_{t\in S} w_t \sum_v p_t^{(D)}(v)\log\frac{p_t^{(D)}(v)}{p_t^{(A)}(v)}.\end{aligned} \tag{6}$$

By matching $p^{(A)}$ to $p^{(D)}$ where full latents actually reduce uncertainty (weights w_t), this term teaches H_K to reproduce the *informative* behavioral effects of H_L while ignoring positions where latents are unhelpful. Unlike reconstruction-based or contrastive objectives, this formulation aligns compressed latents directly through the actor's induced behavior, avoiding assumptions about latent invertibility or instance-level correspondence.

This allows compressed communication to focus on functional equivalence rather than representational similarity. This is particularly important for reasoning latents, which are over-parameterized, temporally misaligned under compression, and lack a natural one-to-one mapping across steps. By aligning compressed and full communications through induced behavior, our formulation supports variable-length latent messages, enables abstraction across multiple reasoning steps, and yields more stable and transferable training signals.

(3) Latent direction alignment. To stabilize compression, we align the *global direction* of actor-side latent features induced by generated vs. data latents. Let $Z_k^{(q)} \in \mathbb{R}^{d_z}$ be the actor-side features (after $g(\cdot)$ and the actor's input stack) at latent step k under path $q \in \{A, D\}$. When H_L has length $L \neq K$, apply a fixed resampling operator ρ_K (*e.g.,* uniform down/up-sampling) and write $Z_{1:K}^{(D)} = \rho_K\big(Z_{1:L}^{(D)}\big)$. Define step-averaged directions $\bar{z}^{(q)} = \frac{1}{K}\sum_{k=1}^{K} Z_k^{(q)}$ and the cosine penalty

$$\begin{aligned}\mathcal{L}_{\text{geom}} &= 1 - \cos\big(\bar{z}^{(A)},\, \bar{z}^{(D)}\big) \\ &= 1 - \frac{\langle \bar{z}^{(A)},\, \bar{z}^{(D)} \rangle}{\|\bar{z}^{(A)}\|_2 \, \|\bar{z}^{(D)}\|_2}.\end{aligned} \tag{7}$$

This term preserves the *geometry* of the actor-side representations, preventing the compressed latents from drifting to directions that the actor interprets differently. Empirically, it improves stability and mitigates mode collapse when K is small by retaining the global semantic orientation of H_L.

Overall objective. The compression objective for M_ϕ (with M_θ frozen) is

$$\begin{aligned}\mathcal{L}_{\text{compress}} = \lambda_{\text{task}}\, \mathcal{L}_{\text{task}} \\ + \lambda_{\text{pref}}\, \mathcal{L}_{\text{pref}} \\ + \lambda_{\text{geom}}\, \mathcal{L}_{\text{geom}}.\end{aligned} \tag{8}$$

In practice, all terms are computed over $t \in S$ with teacher forcing; gradients propagate only to ϕ.

B.1 Baselines and settings in Interlat.

We consider two external baselines, which do not rely on latent communication at all. All baselines are trained using the same base models (Qwen2.5-7B/0.5B-Base, LLaMA3.1-8B-Base) as Interlat, differing only in whether and how inter-agent communication is provided.

1. **CoT (full).** We use complete Chain-of-Thought (CoT) traces produced by a related instruction-tuned model (Qwen2.5-7B-Instruct, Qwen2.5-0.5B-Instruct, and LLaMA3.1-8B-Instruct) to perform full-parameter supervised fine-tuning. In inference, the model receives a complete CoT plan before generating answers.

   Rationale. This baseline serves as a strong upper bound for language-based communication: it evaluates whether latent communication can surpass explicit human-readable planning, and controls for the supervision quality provided by an instruction-tuned teacher.

2. **No-CoT.** The language model is trained to produce the final answer directly, without receiving any plan from other agents.

   Rationale. This baseline isolates the contribution of any communication signal. It tests whether inter-agent exchange, latent or linguistic, is necessary for solving multi-step tasks.

In addition, we evaluate controlled **variants** of Interlat to diagnose what information is encoded in the latents.

1. **Text.** Instead of latent communication, we feed the corresponding CoT plan (in language space) to the actor.

   Rationale. This variant keeps the interaction protocol unchanged while varying only the communication channel. It enables a direct comparison between language-space and latent-space communication under matched training conditions, disentangling architectural factors from representational ones.

2. **No-Comm.** We remove any communication from the actor's input. This variant quantifies the intrinsic benefit of communication in our framework and verifies that performance improvements do not arise solely from modifications to the underlying model parameters.

3. **CrossTask.** We replace the current task's latent communication with one sampled from a different task.

   Rationale. This variant examines whether the actor is genuinely interpreting task-specific latent content. A substantial degradation indicates reliance on meaningful information

encoded in the latents, rather than superficial distributional shortcuts.

4. **Noised.** We add perturbations to the latent communication H: (a) **CovNoise-0.5×/1.0×**: covariance-shaped noise $\varepsilon_t \sim \mathcal{N}(0, \hat{\Sigma})$ with optional strength $\lambda \in \{0.5, 1.0\}$, where $\hat{\Sigma}$ is the sample covariance of the original H; (b) **WhiteNoise**: a control drawn from $\mathcal{N}(0, I)$ with the same length.

   Rationale. Noise-based perturbations interrogate the robustness and locality of latent-space semantics. Covariance-shaped noise preserves global second-order structure, whereas white noise does not, allowing us to assess whether the actor relies on fine-grained geometric relations within true latent trajectories.

5. **CovGauss.** We replace the entire H with *i.i.d.* samples $H_t \sim \mathcal{N}(0, \hat{\Sigma})$ (0μ) and report a robustness check with $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ (μ). These preserve first-second order moments while removing higher-order structure and temporal alignment.

   Rationale. This variant preserves the mean and covariance of the original latent distribution while discarding all higher-order statistics and temporal correlations. It tests whether latent communication conveys information beyond global moments (first-order and second-order moments), *e.g.,* structured reasoning paths or non-Gaussian manifold geometry.

6. **RandomRot.** We apply a structure-preserving but information-scrambling transform $H' = \hat{\mu} + (H - \hat{\mu})\,\hat{\Sigma}^{-1/2}\,Q\,\hat{\Sigma}^{1/2}$, where Q is a Haar-random orthogonal matrix (Mezzadri, 2006).

   Rationale. This preserves the mean/covariance exactly while disrupting higher-order structure. Random rotation strictly preserves the first two moments of the latent distribution while scrambling its geometric orientation and higher-order structure. This constitutes a strong diagnostic of whether the actor depends on directional semantics or sequential organization within the latent manifold, rather than mere distributional similarity.

7. **Cross-Family.** We evaluate Interlat under a cross-model-family setting, where the sender and actor belong to different pretrained model families. Specifically, latent communications are generated by a sender model from one family (In this work, we use Qwen2.5-7B-Instruct) and consumed by an actor model from another family (LLaMA3.1-8B-Base), without sharing parameters or tokenizer vocabularies.

   Rationale. This setting tests whether latent communication encodes task-relevant information in a model-agnostic manner, rather than exploiting family-specific activation conventions or implicit alignment.

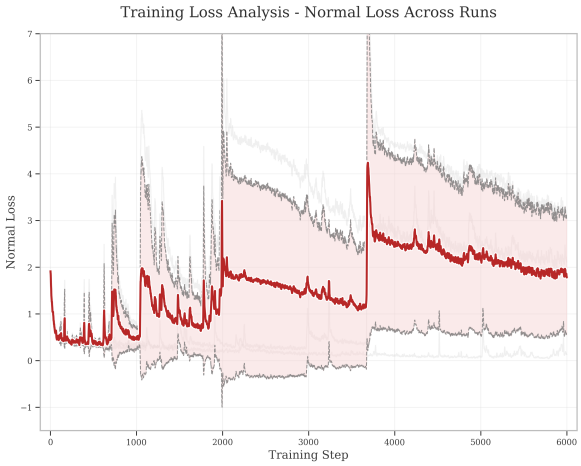


Figure 7: Training dynamics of the cross-entropy loss when curriculum learning is removed, illustrating highly unstable optimization behavior.

C Benchmark

C.1 Alfworld

Alfworld (Shridhar et al., 2020) is a text-only benchmark that simulates embodied household tasks while keeping interaction purely in natural language. Agents observe textual descriptions of the scene and issue high-level commands from a constrained action set (*e.g.,* go to, open, close, take, put, toggle on/off, heat, cool, examine). Tasks are long-horizon and compositional, requiring perception, planning, and execution over multiple steps under partial observability. The benchmark provides official train/seen/unseen splits and a standard success metric under a fixed step budget (*e.g.,* 20 steps in our setup), enabling systematic and reproducible evaluation of sequential decision-making.

D Ablations and Step Analysis

We present ablation studies for both the actor and reasoning models, reporting the average number

of steps for successful trials versus all trials (success/all).

Effect of curriculum learning. For the actor model, removing curriculum learning forces the agent to interpret latent communications from scratch. As shown in Figure 7, this leads to highly unstable training dynamics and substantially degraded latent comprehension, preventing the model from consistently leveraging the communicated information.

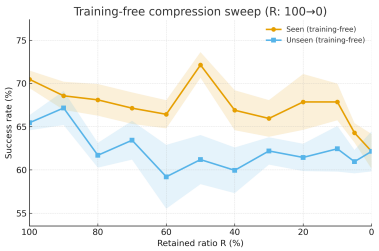
Step count versus performance. Table 6 reveals a nuanced but systematic relationship between step count and task performance. On seen tasks, ablating key components results in a lower overall success rate. Although these ablated models often take fewer steps on the trials they complete, their high failure rate indicates an inability to reliably interpret latent communication and solve tasks. In contrast, the full model achieves both higher success rates and longer trajectories, suggesting that additional steps correspond to productive exploration rather than inefficiency.

On unseen tasks, several ablations (*e.g.,* removing curriculum learning or the communication adapter) exhibit the opposite pattern: the agent takes more steps while achieving a lower success rate. This demonstrates that longer trajectories alone do not imply effective exploration. Without these critical components, the policy exhibits unstructured search behavior that fails to form coherent task-solving strategies. Together, these observations underscore the importance of evaluating step count jointly with success rate, and support our central claim that information-rich latent communication enables structured and effective exploration rather than random wandering.

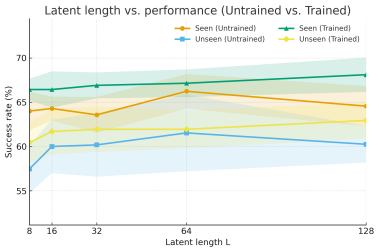
E Compression Result

In this section, we provide more detailed results on compression with average steps as success/all across tasks in Table 7 (LLaMA3.1-8B-Base) and Table 8 (Qwen2.5-7B-Base) and corresponding performance trend in Figure 8. Latency is measured on the same machine and decoding policy (if needed) across rows [2].

[2]For the untrained reasoning model, we use the standard `generate` API from Hugging Face `transformers`; see https://github.com/huggingface/transformers.



(a) Training-free sweep over retained ratio R



(b) Latent length L vs. performance (untrained & trained)

Figure 8: Result of compression on seen and unseen tasks. **Left:** Success rate under training-free compression with different retained ratios R. **Right:** Performance of untrained and trained models across latent lengths L

F Latent Parallelism Analysis

We first compared the latent communications produced by our trained reasoning model with those from an off-the-shelf Qwen2.5-7B-Instruct model in the compression-effectiveness analysis (see the Experiments section). Because our reasoning model is initialized from Qwen2.5-7B-Base, we additionally compare it with this base model, which has not been trained for generating compressed latent communication, in Figure 9. The findings are consistent with the earlier comparison: the trained model maintains stable vertical gaps between successive Top-k curves across steps and exhibits a substantially lower $P_{50}(S_{10})$, whereas the base model shows a clear convergence toward Top-1.

We further extend the parallelism analysis to a deeper horizon of 32 steps. As shown in Figure 10, the trained model exhibits stable vertical gaps between successive Top-k curves throughout these steps. This extended analysis further verifies that the trained latent representations preserve a broader set of plausible reasoning paths by sustaining a more balanced probability distribution rather than prematurely collapsing to a Top-1 hypothesis.

Method	Seen	Steps	Unseen	Steps
Actor model				
Ours Full	**70.48**$_{\pm 1.01}$	9.41/12.54	**65.42**$_{\pm 0.87}$	9.86/13.37
w/o curri	33.10$_{\pm 2.97}$	9.07/16.38	20.65$_{\pm 2.15}$	10.47/18.03
w/o $\mathcal{L}_{\text{sep}}$	<u>58.81</u>$_{\pm 1.41}$	8.07/12.98	<u>60.70</u>$_{\pm 5.50}$	9.64/13.71
w/o $\mathcal{L}_{\text{align}}$	56.90$_{\pm 1.41}$	8.16/13.26	53.98$_{\pm 3.35}$	9.56/14.36
w/o adapter	4.05$_{\pm 1.70}$	9.32/19.57	4.48$_{\pm 1.31}$	10.53/19.58
Reasoning model				
Ours Full	**68.10**$_{\pm 1.93}$	9.21/12.65	<u>62.94</u>$_{\pm 2.03}$	9.88/13.63
w/o $\mathcal{L}_{\text{task}}$	<u>65.71</u>$_{\pm 1.43}$	8.86/12.68	**63.18**$_{\pm 3.47}$	9.68/13.48
w/o $\mathcal{L}_{\text{pref}}$	64.76$_{\pm 2.97}$	8.92/12.82	60.20$_{\pm 3.13}$	9.68/13.79
w/o $\mathcal{L}_{\text{geom}}$	64.05$_{\pm 3.55}$	8.71/12.77	59.45$_{\pm 3.01}$	9.88/13.98

Table 6: Ablation of training components. "Ours Full" uses all components.

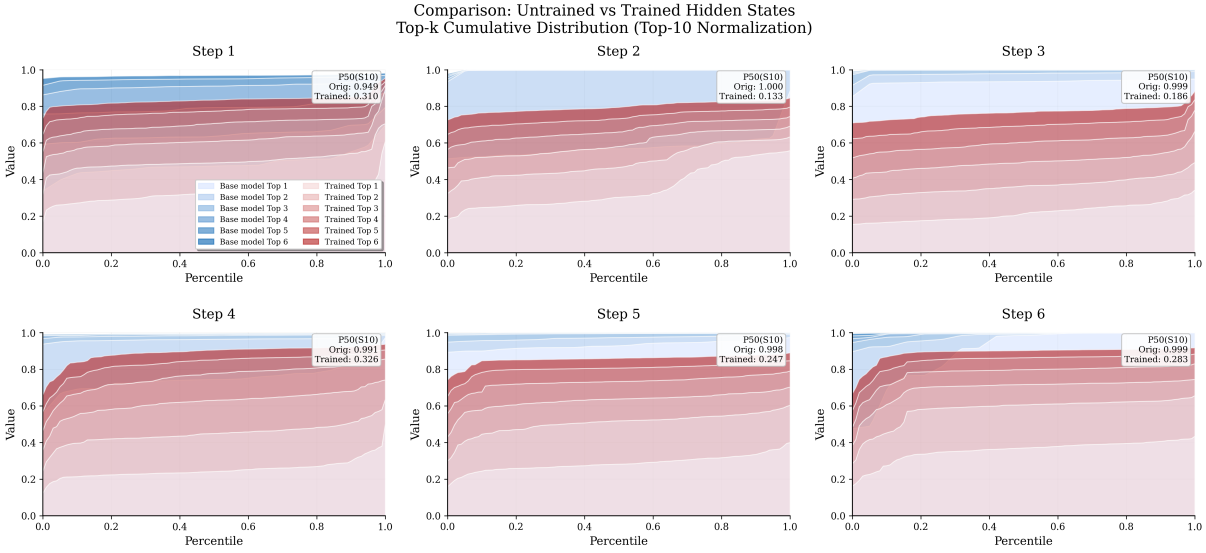


Figure 9: Parallelism in latent communication over the first six steps. Red indicates latents from the trained model, and blue indicates latents from the untrained base model. The trained latents preserve stable vertical gaps between successive Top-k bands and achieve a markedly lower $P_{50}(S_{10})$, evidencing persistent parallelism, whereas the untrained base model's latents progressively collapse toward Top-1.

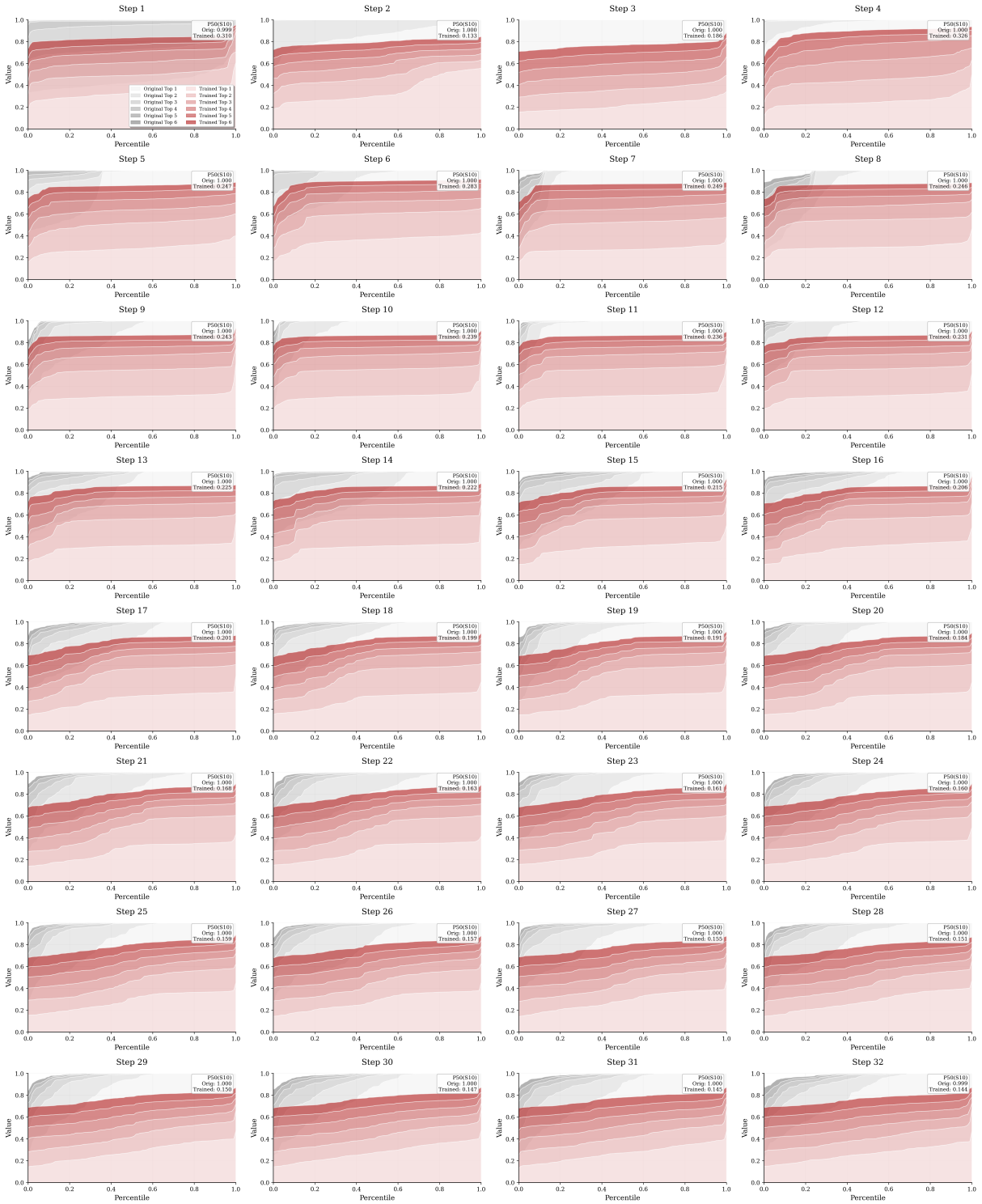


Figure 10: **Extended analysis (32 steps).** Same construction as Fig. 5, now for steps 1–32. Persistent separation among successive Top-k bands and consistently lower $\mathrm{P50}(S_{10})$ values indicate that the trained latents maintain broad, plausible reasoning branches across the entire sequence, despite compression.

LLaMA3.1-8B					
Ratio	Seen	Steps	Unseen	Steps	Time
Untrained					
Full	**70.71**$_{\pm 1.04}$	8.02/12.58	**70.90**$_{\pm 1.21}$	8.21/12.96	10.20s
90%	60.09$_{\pm 2.94}$	8.12/12.86	60.45$_{\pm 1.58}$	9.51/13.66	-
80%	<u>65.27</u>$_{\pm 1.66}$	8.23/12.32	<u>63.43</u>$_{\pm 4.84}$	9.05/13.06	-
70%	60.62$_{\pm 2.43}$	8.14/12.81	61.79$_{\pm 2.81}$	10.13/13.90	-
60%	59.82$_{\pm 2.55}$	7.90/12.76	63.44$_{\pm 2.79}$	9.39/13.27	-
50%	63.12$_{\pm 3.58}$	8.02/12.44	60.07$_{\pm 1.43}$	9.85/13.90	-
40%	65.27$_{\pm 1.08}$	8.06/12.21	59.70$_{\pm 1.83}$	9.67/13.83	-
30%	61.25$_{\pm 2.61}$	8.20/12.77	59.51$_{\pm 2.39}$	10.13/14.13	-
20%	61.79$_{\pm 3.10}$	8.41/12.84	57.84$_{\pm 1.58}$	9.48/13.92	-
10%	64.91$_{\pm 1.23}$	8.60/12.60	60.45$_{\pm 4.22}$	9.48/13.64	-
5%	63.68$_{\pm 2.57}$	8.42/12.62	60.95$_{\pm 1.35}$	9.90/13.84	-
0%	63.57$_{\pm 2.44}$	8.35/12.59	58.40$_{\pm 2.76}$	9.47/13.85	-
128L	61.59$_{\pm 2.34}$	8.24/12.75	62.39$_{\pm 2.82}$	10.01/13.77	4.00s
64L	62.54$_{\pm 2.90}$	8.38/12.73	60.52$_{\pm 3.08}$	9.86/13.86	2.10s
32L	62.30$_{\pm 2.08}$	8.33/12.73	57.46$_{\pm 2.28}$	9.98/14.24	1.20s
16L	63.89$_{\pm 2.70}$	8.53/12.67	59.61$_{\pm 3.53}$	9.50/13.74	0.70s
8L	61.79$_{\pm 2.44}$	8.09/12.64	58.77$_{\pm 2.76}$	9.99/14.12	0.45s
Trained					
128L	66.46$_{\pm 1.98}$	8.18/12.54	66.35$_{\pm 1.86}$	8.96/13.12	2.80s
64L	66.21$_{\pm 1.72}$	8.12/12.58	65.42$_{\pm 1.94}$	9.02/13.20	1.40s
32L	65.45$_{\pm 1.63}$	8.08/12.60	65.01$_{\pm 1.88}$	9.08/13.28	0.72s
16L	64.41$_{\pm 1.95}$	8.10/12.64	65.20$_{\pm 1.76}$	9.12/13.34	0.39s
8L	64.32$_{\pm 1.84}$	8.14/12.66	64.89$_{\pm 1.69}$	9.18/13.40	0.24s

Table 7: Complete compression results with seen/unseen accuracy, steps, and latency across tasks on LLaMA models.

G Qualitative Analysis of Latent Communication via PCA

To qualitatively examine the semantic structure encoded in latent communications, we perform Principal Component Analysis (PCA) on 3,119 samples from the ALFWorld training set. Each sample corresponds to the mean-pooled last-layer hidden state generated by the reasoning agent for a specific task instance. Tasks are grouped according to the official ALFWorld task templates, which define six core reasoning patterns: `pick_and_place`, `pick_clean_then_place`, `pick_heat_then_place`, `pick_cool_then_place`, `look_in_recep`, and `look_at_obj`. Figure 11 visualizes the projection of latent communications onto the first two principal components, which capture the dominant axes of variance across task instances.

The resulting PCA visualization reveals clear task-dependent organization in the latent space. Action-centric templates such as `pick_and_place` form a dense central cluster, while templates involving additional procedural constraints—such as thermal manipulation in `pick_heat_then_place` and `pick_cool_then_place`—occupy adjacent yet separable regions. Perception-oriented tasks (`look_in_recep` and `look_at_obj`), although less frequent, also exhibit localized concentrations distinct from execution-heavy templates. As PCA preserves global variance structure rather than emphasizing local neighborhoods, this separation indicates that task-specific semantics are encoded in the dominant latent dimensions, rather than arising from projection artifacts. Moreover, intra-cluster dispersion (e.g., within `pick_and_place`) suggests that latent representations retain fine-grained variations across task instances, rather than collapsing to a single prototype per template.

Qwen2.5-7B					
Ratio	Seen	Steps	Unseen	Steps	Time
Untrained					
Full	$\underline{70.48}_{\pm 1.01}$	9.41/12.54	$\underline{65.42}_{\pm 0.87}$	9.86/13.37	9.19s
90%	$68.57_{\pm 1.63}$	8.77/12.30	$\mathbf{67.16}_{\pm 1.97}$	9.27/12.79	-
80%	$68.10_{\pm 1.83}$	8.56/12.21	$61.69_{\pm 1.43}$	9.10/13.28	-
70%	$67.14_{\pm 1.82}$	8.68/12.40	$63.43_{\pm 2.24}$	9.42/13.29	-
60%	$66.43_{\pm 1.63}$	8.52/12.37	$59.20_{\pm 3.69}$	9.90/14.02	-
50%	$\mathbf{72.14}_{\pm 1.48}$	9.03/12.09	$61.19_{\pm 2.84}$	9.37/13.50	-
40%	$66.90_{\pm 2.31}$	8.88/12.56	$59.95_{\pm 2.64}$	9.52/13.72	-
30%	$65.95_{\pm 2.12}$	8.80/12.61	$62.19_{\pm 1.58}$	10.11/13.85	-
20%	$67.86_{\pm 3.23}$	8.97/12.52	$61.44_{\pm 1.58}$	9.98/13.84	-
10%	$67.86_{\pm 2.12}$	8.76/12.37	$62.44_{\pm 2.64}$	9.72/13.58	-
5%	$64.52_{\pm 1.12}$	9.19/13.02	$60.95_{\pm 1.35}$	9.90/13.84	-
0%	$62.14_{\pm 2.01}$	10.19/13.90	$62.19_{\pm 2.32}$	10.23/13.92	-
128L	$64.52_{\pm 2.26}$	8.68/12.70	$60.20_{\pm 2.06}$	9.69/13.79	3.55s
64L	$66.19_{\pm 1.95}$	8.76/12.56	$61.44_{\pm 4.32}$	9.85/13.76	1.83s
32L	$63.57_{\pm 2.01}$	8.66/12.79	$60.20_{\pm 3.58}$	9.87/13.90	1.03s
16L	$64.29_{\pm 1.34}$	8.64/12.70	$59.95_{\pm 3.01}$	10.07/14.05	0.62s
8L	$64.05_{\pm 2.18}$	8.80/12.83	$57.46_{\pm 2.69}$	10.29/14.42	0.39s
Trained					
128L	$68.10_{\pm 1.93}$	9.21/12.65	$62.94_{\pm 2.03}$	9.88/13.63	2.25s
64L	$67.14_{\pm 1.56}$	9.15/12.72	$61.94_{\pm 2.13}$	9.92/13.76	1.16s
32L	$66.90_{\pm 1.46}$	9.02/12.65	$61.94_{\pm 2.56}$	9.96/13.78	0.60s
16L	$66.43_{\pm 2.05}$	9.08/12.75	$61.69_{\pm 2.56}$	9.98/13.82	0.33s
8L	$66.43_{\pm 1.22}$	9.11/12.77	$60.45_{\pm 2.23}$	9.90/13.89	0.20s

Table 8: Complete compression results with seen/unseen accuracy, steps, and latency across tasks on Qwen models.

Beyond static clustering, we further analyze how these task-level structures are affected by the latent communication adapter. As shown in Figure 12 and quantified in Table 9, different task templates exhibit consistent but template-dependent centroid shifts before and after transformation. While the absolute magnitudes of these shifts are moderate relative to within-template dispersion, their directions are highly structured, indicating selective reorganization rather than global rescaling of the latent space. Execution-heavy templates such as `pick_and_place` and `pick_heat_then_place` undergo larger relative shifts, whereas observation-oriented templates remain more stable. Together, these results suggest that Interlat preserves the overall geometry of latent communications while inducing task-aware semantic alignment, enabling the actor agent to differentiate diverse reasoning paradigms without relying on explicit natural language communication.

H Training Template

We present an example in Figure H to illustrate how agents explore and solve tasks in Alfworld. After perceiving the environment, the agent executes an action, receives feedback from the environment, and then proceeds to the next step until the goal is accomplished. Figure H shows an example of how training data is structured for the actor agent. We append either the latent communication or the natural-language plan after the instruction to facilitate inter-agent communication.

Task template	#Samples	L	R (median)	$\rho = L/R$
pick_and_place	848	1.742	8.347	0.209
heat_then_place	578	2.403	11.506	0.209
cool_then_place	429	1.329	13.111	0.101
look_at_obj	187	0.794	8.498	0.093
clean_then_place	632	0.989	17.839	0.055
look_in_recep	431	0.250	23.393	0.011
other	14	0.581	32.346	0.018

Table 9: Quantitative centroid shift analysis for each task template.

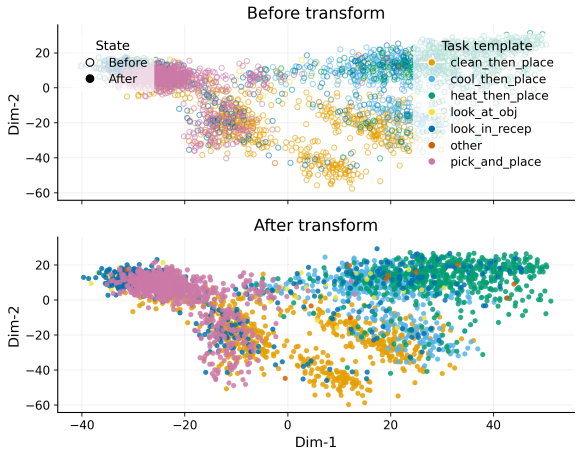


Figure 11: PCA visualization of latent communications grouped by ALFWorld task templates ($N = 3{,}119$). Each point corresponds to the mean-pooled last-layer hidden state of the reasoning agent's plan for a specific task instance. Colors indicate six core task templates: `pick_and_place`, `pick_clean_then_place`, `pick_heat_then_place`, `pick_cool_then_place`, `look_in_recep`, and `look_at_obj`. The emergence of distinct clusters suggests that Interlat's latent communication captures task-specific semantic structure, enabling the actor agent to differentiate diverse reasoning paradigms without relying on natural language.

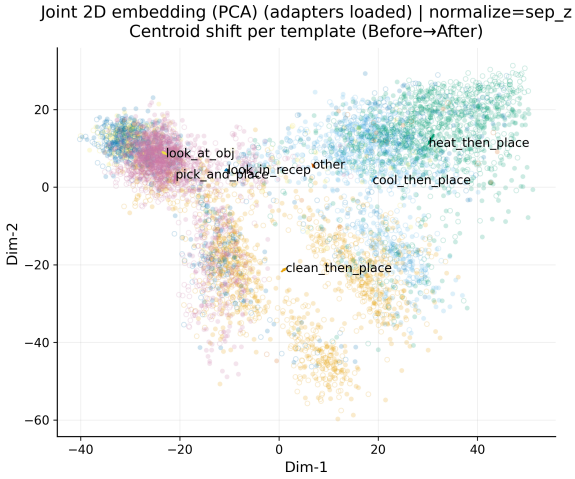


Figure 12: Joint PCA visualization of latent communications grouped by ALFWorld task templates. Each point represents the mean-pooled last-layer hidden state of the reasoning agent for a single task instance ($N = 3{,}119$). Colors indicate different task templates. Solid points correspond to latent representations after applying the communication adapter, while hollow points denote the original latents before transformation. Arrows depict the centroid shift for each template from before to after. Although the centroid shifts are moderate in magnitude, their directions are consistently template-dependent, indicating structured semantic reorganization rather than global rescaling of the latent space.

Conversations

Env
You are in the middle of a room. Looking quickly around you, you see a bathtubbasin 1, a cabinet 2, a cabinet 1, a countertop 1, a garbagecan 1, a handtowelholder 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1.
Task: put a toiletpaper in toiletpaperhanger.

Agent
Thought: I need to place a toilet paper roll on the toilet paper hanger. First, I should check if there's already a roll on the hanger or if it's empty.
Action: go to toiletpaperhanger 1

Env
Observation: On the toiletpaperhanger 1, you see nothing.

Agent
Thought: The toilet paper hanger is empty, so I need to find a new roll of toilet paper to put on it.
Action: go to toilet 1

Env
Observation: On the toilet 1, you see a soapbottle 1, and a toiletpaper 1.

Agent
Thought: I've found a toilet paper roll on the toilet. I should pick it up so I can place it on the hanger.
Action: take toiletpaper 1 from toilet 1

Env
Observation: You pick up the toiletpaper 1 from the toilet 1.

Agent
Thought: Now that I have the toilet paper roll, I need to return to the toilet paper hanger to install it.
Action: go to toiletpaperhanger 1

Env
Observation: On the toiletpaperhanger 1, you see nothing.

Agent
Thought: I'm back at the toilet paper hanger with the roll in hand. It's time to put the toilet paper on the hanger.
Action: put toiletpaper 1 in/on toiletpaperhanger 1

Training Template for Actor Agent

<im_start>system
You are a helpful assistant. <im_end>
<im_start>user
Interact with a household to solve a task. Imagine you are an intelligent agent in a household environment and your target is to perform actions to complete the task goal. At the beginning of your interactions, you will be given a detailed description of the current environment and your goal to accomplish.

For each of your turns, you will be given the observation of the last turn. You should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format:"Thought: your thoughts. Action: your next action".
The available actions are:
1. go to recep
2. task obj from recep
3. put obj in/on recep
4. open recep
5. close recep
6. toggle obj recep
7. clean obj with recep
8. heat obj with recep
9. cool obj with recep
where obj and recep correspond to objects and receptacles.
After your each turn, the environment will give you immediate feedback based on which you plan your next few steps. if the envrionment output "Nothing happened", that means the previous action is invalid and you should try more options.
Your response should use the following format:
Thought: <your thoughts>
Action: <your next action>
You are in the middle of a room. Looking quickly around you, you see a armchair 2, a armchair 1, a coffeetable 2, a coffeetable 1, a diningtable 1, a garbagecan 1, a sidetable 2, a sidetable 1, and a sofa 1.
Your task is to: find two newspaper and put them in armchair.
Now, you are given a step-by-step plan to complete this task as follow:
<bop>
Insert a latent communication or textual plan for inter-agent communication.
<eop>
<im_end>
<im_start>assistant
Thought: To complete the task, I need to locate two newspapers and place them in an armchair. I should start by searching the nearby furniture for any newspapers.
Action: go to sofa 1 <im_end>
<im_start>user
Observation: On the sofa 1, you see a box 3, a box 2, a newspaper 2, and a pillow 1. <im_end>
<im_start>assistant
Thought: I've found one newspaper on sofa 1. I should pick it up to place it in the armchair later.
Action: take newspaper 2 from sofa 1 <im_end>
<im_start>user
Observation: You pick up the newspaper 2 from the sofa 1. <im_end>
<im_start>assistant
Thought: With newspaper 2 in hand, I need to find the second newspaper. The dining table is another common place where newspapers might be found.
Action: go to dining table 1 <im_end>
<im_start>user
Observation: On the dining table 1, you see a box 1, a credit card 2, a credit card 1, a laptop 1, a newspaper 1, a plate 3, a plate 1, a remote control 1, a statue 3, a statue 1, a tissue box 3, and a tissue box 1. <im_end>
<im_start>assistant
There's the second newspaper on the dining table. I should take it so I can place both newspapers in the armchair as instructed.
Action: take newspaper 1 from dining table 1 <im_end>
<im_start>user
Observation: You pick up the newspaper 1 from the dining table 1. <im_end>
<im_start>assistant
Thought: With both newspapers now in my possession, I need to find an armchair to place them in.
Action: go to armchair 1 <im_end>
<im_start>user
Observation: On the armchair 1, you see a keychain 1. <im_end>
<im_start>assistant
Thought: The armchair has a keychain on it, but there's enough space for the newspapers. I'll place the first newspaper here.
Action: put newspaper 2 in/on armchair 1 <im_end>
<im_start>user
Observation: You put the newspaper 2 in/on the armchair 1. <im_end>

<im_start>assistant
Thought: With the first newspaper placed, I need to put the second newspaper in the armchair to complete the task.
Action: put newspaper 1 in/on armchair 1
<im_end>