XBRIDGE: Entity-Grounded Latent Bridge for Heterogeneous LLM Communication

Wooseong Yang[1] **Wei-Chieh Huang**[1] **Weizhi Zhang**[1] **Yu Wang**[2*]
Philip S. Yu[1,†] **Junhyun Lee**[3,4,†]
[1]University of Illinois Chicago [2]Capital One AI Foundations
[3]Hankuk University of Foreign Studies [4]Noah's Farm
[†]Corresponding authors

Abstract

Heterogeneous multi-agent LLM systems, where agents are powered by different model families, can outperform homogeneous configurations by reducing redundant reasoning patterns. Yet existing communication protocols either operate through text, discarding the sender's internal representations, or require architectural homogeneity, such as a shared tokenizer or matching KV cache layouts, for latent-level transfer. We identify the *entity grounding problem* in cross-architecture communication: cross-attention bridges that transfer continuous representations across different LLM families suffer from rare-token compression collapse, where entity identity is lost in the continuous bottleneck (bridge-only F1 $\sim$30%). We propose XBRIDGE, a decode-free protocol for latent communication between heterogeneous LLMs, which addresses this through two mechanisms. Lexical Anchor Mapping (LAM) maps the sender's original context tokens to the receiver's vocabulary, providing discrete entity anchors. A Latent Enrichment Bridge (LEB) lets the receiver query the sender's hidden states for contextual enrichment. The entity anchors ground the bridge's contextual signals to specific entities through the receiver's own self-attention. Across three model families (Llama, Qwen, and Mistral), seven long-context and multi-hop QA benchmarks, and both communication directions, XBRIDGE outperforms text-based communication on all seven tasks for each model pair while achieving 11$\times$ lower latency, and in a same-architecture setting it also exceeds a KV-sharing baseline on six of seven tasks. Both models stay frozen, and LEB requires only 264M trainable parameters (3.8% of the receiver), is trained on a small balanced sample set, and adds negligible inference overhead. Our implementation is available at `https://github.com/WooseongYang/XBridge`.

1 Introduction

Multi-agent systems (MAS) can improve the reasoning capabilities of large language models (LLMs) by distributing computation across multiple inference passes [1]. Recent findings show that when interacting agents share identical parameters and inductive biases, their interactions may degenerate into redundant majority voting rather than surfacing complementary reasoning [2, 3]. A natural way to mitigate this redundancy is to use heterogeneous ensembles: configurations comprising distinct model families with diverse tokenizers, training corpora, architectures, and alignment procedures. Yet this representational diversity introduces a severe communication bottleneck: agents must exchange information across structurally incompatible latent spaces.

Existing communication paradigms face an efficiency-fidelity trade-off. Text-based methods, such as natural language communication (NLComm) [4], are architecture-agnostic but incur autoregressive

*Work done prior to joining Capital One.

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

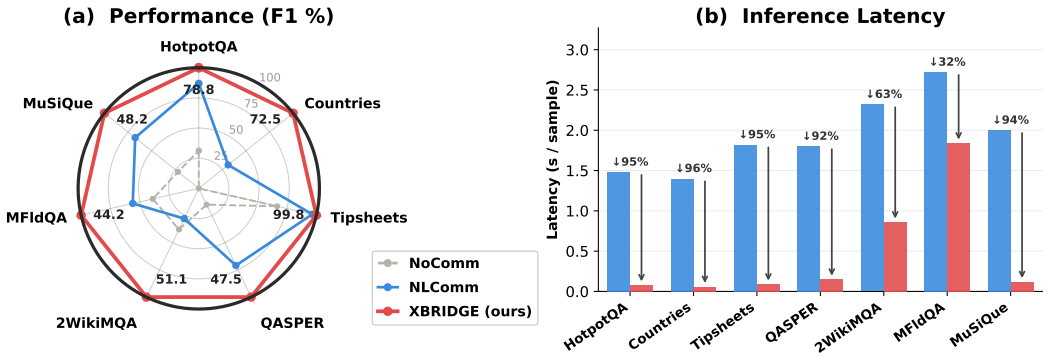


Figure 1: XBRIDGE performance and latency (Llama-3.1-8B → Qwen2.5-7B). **(a)** F1 scores across seven benchmarks. NoComm: receiver only (no communication). NLComm: heterogeneous text-based communication [4]. **(b)** Per-sample inference latency on H200 GPU. Speedup labels indicate XBRIDGE vs. NLComm.

decoding latency and compress the sender's internal state into a textual summary. Latent-level methods avoid sender-side decoding, but existing approaches such as KV cache sharing [5] and hidden-state injection [6] typically assume architectural compatibility, including aligned hidden dimensions, attention layouts, positional encodings, or tokenizers. Learned continuous projections offer a possible route across model families [7]. However, we find that continuous projections alone degrade in a specific and practically important way: they can transfer contextual information without reliably preserving the entities to which that context refers.

We define this degradation as the *entity grounding problem*. In heterogeneous LLM communication, a continuous bridge can transfer contextual information from the sender, but it struggles to preserve the discrete entity identities, such as names, numbers, or rare tokens, to which that context refers. We use *entity fidelity* to refer to preserving the correct lexical identity, and *entity grounding* to refer to binding the sender's contextual signal to the corresponding receiver-native lexical anchor. In bridge-only experiments, the receiver recovers aspects of the context but often loses the exact identities of the participating entities. Homogeneous communication settings are less exposed to this problem because shared tokenizers and representational spaces provide implicit entity anchors. Heterogeneous communication therefore creates a demanding setting in which continuous contextual transfer and exact discrete entity fidelity must be maintained simultaneously.

To address this bottleneck, we propose **XBRIDGE**, a decode-free communication protocol that mitigates the entity grounding problem via dual channels. First, **Lexical Anchor Mapping (LAM)** deterministically translates the sender's original context tokens into the receiver's vocabulary, providing discrete entity anchors without autoregressive decoding. Second, a **Latent Enrichment Bridge (LEB)** uses gated cross-attention to let the receiver query the sender's hidden states, supplying continuous contextual information that can be bound to these lexical anchors. By decoupling the preservation of discrete entity identity (the *who* and *what*) from the transfer of continuous contextual knowledge (the *how* and *why*), XBRIDGE enables heterogeneous models to communicate without requiring textual decoding or direct latent-space compatibility. The bridge is lightweight, requiring only 264M trainable parameters (3.8% of the receiver), training on a small balanced sample set in under 10 minutes, and adding negligible inference overhead.

In summary, our core contributions are:

- We formulate the entity grounding problem in heterogeneous LLM communication and demonstrate rare-token compression collapse, a failure mode in which continuous bridges preserve contextual signals but lose discrete entity identity.
- We propose XBRIDGE, a decode-free dual-channel protocol that combines Lexical Anchor Mapping for receiver-native entity anchors with Latent Enrichment Bridge for sender-side contextual enrichment. To our knowledge, this is the first protocol to pair an explicit discrete-anchor channel with latent transfer for heterogeneous communication.
- We show that XBRIDGE improves heterogeneous communication across three model families and seven benchmarks, achieving 11× lower latency than text-based communication, and that in a same-architecture setting it exceeds KV-cache sharing on 6 of 7 tasks.

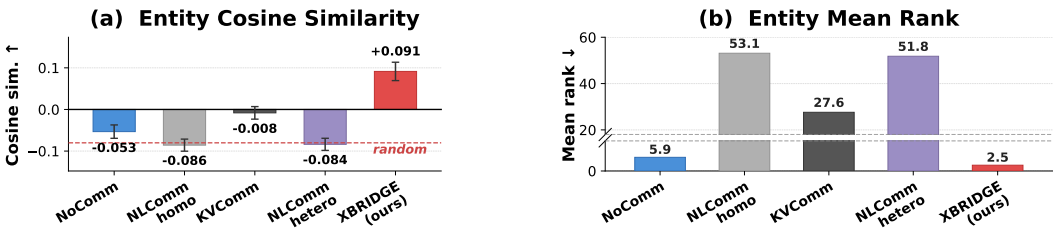


Figure 2: Entity fidelity across five communication methods on HotpotQA (Llama→Qwen, 196 samples). NoComm receives no sender signal. NLComm [4] relays text summaries within the same family (homo) or across families (hetero). KVComm [5] shares KV caches within the same architecture. **(a)** Cosine similarity between generation and answer entity representations. Only XBRIDGE achieves positive alignment. **(b)** Mean rank of the correct answer token (lower is better). All methods degrade rank beyond the no-communication baseline, while XBRIDGE improves it.

2 The Entity Grounding Problem

Heterogeneous communication requires satisfying three requirements. First, the receiver must recover receiver-native lexical identities for entities that appear in the sender context. Second, it must use contextual information encoded in the sender's hidden states. Third, it must consume this information without treating sender activations as native receiver activations. We refer to these requirements as *entity fidelity*, *entity grounding*, and *representation compatibility*, respectively.

We formalize heterogeneous LLM communication as information transfer between models with incompatible tokenizers, vocabularies, and hidden spaces. Let

$$M_i = (\tau_i, V_i, E_i, F_i, G_i), \qquad i \in \{S, R\}, \tag{1}$$

denote a sender M_S and receiver M_R, where τ_i is the tokenizer, V_i the vocabulary, E_i the embedding map, F_i the transformer, and G_i the language-model head. The models are heterogeneous when

$$V_S \neq V_R, \qquad d_S \neq d_R, \qquad F_S \not\equiv F_R, \tag{2}$$

where architectural mismatch may include differences in depth, attention layout, normalization, positional encoding, tokenizer design, or training distribution.

The sender observes a context C, while the receiver observes a question Q and must generate an answer A. The sender performs a single forward pass:

$$c_S = \tau_S(C) = (c_1, \ldots, c_{T_C}), \qquad H_S = F_S(E_S(c_S)) \in \mathbb{R}^{T_C \times d_S}. \tag{3}$$

Unless otherwise stated, H_S denotes the sender's last-layer hidden states. A communication protocol produces a sender-side message $m_S = \pi_S(c_S, H_S)$, which the receiver consumes to generate $p_\pi(A \mid C, Q) = p_R(A \mid Q; \Gamma_\theta(Q, m_S))$. Here, Γ_θ denotes the receiver-side mechanism for consuming the message. Text protocols communicate strings or token sequences, latent protocols communicate continuous states, and hybrid protocols communicate both $m_S = \big(m_{\text{disc}}(C), m_{\text{cont}}(C)\big)$.

2.1 Entity Fidelity

Let $\mathcal{E}(C)$ denote the set of entity mentions in C, including names, numbers, dates, and other rare or semantically specific spans. For an entity $e \in \mathcal{E}(C)$, define its receiver-native lexical realization as

$$a_R(e) = \tau_R(e) \in V_R^*, \tag{4}$$

where V_R^* is the set of finite token sequences over the receiver vocabulary. We call $a_R(e)$ an *entity anchor*. Entity fidelity is the requirement that the receiver assigns high probability to the correct receiver-native anchor.

For a gold answer entity $e^\star$, we measure entity fidelity by the receiver-side rank

$$\text{rank}_\pi(e^\star; C, Q) = \text{rank}_{v \in V_R} \left[\ell_R(v \mid Q; \Gamma_\theta(Q, m_S))\right]_{v \in a_R(e^\star)}, \tag{5}$$

where ℓ_R is the receiver logit induced by G_R. For multi-token entities, we use the first answer token as a diagnostic proxy. A protocol has high entity fidelity when this rank is low.

Entity fidelity can also be characterized interventionally. Let $C^{e \to e'}$ denote the context obtained by replacing every occurrence of e with another entity e' of the same semantic type. A protocol is entity-sensitive if

$$\hat{A}_\pi(C, Q) = e \quad \Longrightarrow \quad \hat{A}_\pi(C^{e \to e'}, Q) = e'. \tag{6}$$

This condition separates entity identity from relational structure. If the relation is preserved but the answer does not follow the substituted entity, the protocol does not preserve entity fidelity.

2.2 Entity Grounding Failure

Entity fidelity concerns whether the receiver outputs the correct entity. Entity grounding concerns whether the sender's contextual signal is bound to the correct receiver-native entity anchor. A protocol may transfer contextual information while still leaving that information ungrounded to the correct entity in V_R. A continuous bridge can be written abstractly as $r_\theta^{(\ell)} = B_\theta^{(\ell)}(H_S, h_R^{(\ell)}) \in \mathbb{R}^{T_R \times d_R}$, where $h_R^{(\ell)} \in \mathbb{R}^{T_R \times d_R}$ is the receiver hidden-state sequence at layer ℓ with the receiver sequence length T_R. This abstraction covers projections, adapters, hidden-state fusion, and cross-attention modules. Such a bridge may transmit contextual information encoded in H_S, but by itself it does not expose receiver-native anchors $a_R(e)$.

The resulting failure is an identifiability problem. Consider two contexts C_e and $C_{e'}$ that differ primarily by replacing entity e with another entity e' while preserving the same relational structure. A continuous-only protocol is vulnerable when $d\big(m_{\text{cont}}(C_e), m_{\text{cont}}(C_{e'})\big) \leq \epsilon$, while the required receiver-native anchors remain distinct: $a_R(e) \neq a_R(e')$. In this regime, the receiver may recover the context but struggles to determine which discrete receiver-vocabulary entity instantiates the answer. We refer to this as *rare-token compression collapse*: continuous representations preserve contextual semantics but lose reliable access to rare or specific lexical identities.

2.3 Representation Mismatch

Even if entity identity is preserved, sender activations cannot be directly treated as receiver activations. Let $\mu_R^{(\ell)}$ denote the distribution of token-level hidden states encountered by receiver layer ℓ. For a projection $P : \mathbb{R}^{d_S} \to \mathbb{R}^{d_R}$, dimension matching only ensures that PH_S lies in $\mathbb{R}^{d_R}$. It does not imply that projected sender states match the receiver's native hidden-state distribution: $PH_S \not\sim \mu_R^{(\ell)}$ in general, where $\mu_R^{(\ell)}$ denotes the distribution of hidden states at the ℓ-th layer of the receiver model. Directly injecting PH_S into the receiver residual stream or key-value cache can therefore introduce out-of-distribution activations.

A representation-compatible protocol should instead let the receiver consume sender information through its own hidden state:

$$h_R'^{(\ell)} = h_R^{(\ell)} + g_\theta^{(\ell)}(h_R^{(\ell)}, H_S), \tag{7}$$

where $g_\theta^{(\ell)}$ is a learned receiver-conditioned retrieval operator. This formulation preserves the receiver's native computation while allowing access to sender-side contextual information.

3 XBRIDGE: Entity-Grounded Latent Communication

The analysis in Section 2 identifies two challenges for heterogeneous architecture communication: continuous bridges struggle to preserve discrete entity identity (Section 2.2), and sender activations lie outside the receiver's representation space (Section 2.3). We therefore decompose heterogeneous communication into two complementary mechanisms: **Lexical Anchor Mapping** (LAM, Section 3.2) provides discrete entity anchors, and a **Latent Enrichment Bridge** (LEB, Section 3.3) provides contextual enrichment. Both signals are produced by a single sender forward pass with no autoregressive decoding; sender and receiver remain frozen throughout.

3.1 XBRIDGE Overview

We adopt the asymmetric communication setting formalized in Section 2, following the protocol established by Shi et al. [5], in which the sender observes a context and the receiver observes a question: the sender M_S processes a context document C through a single forward pass, producing

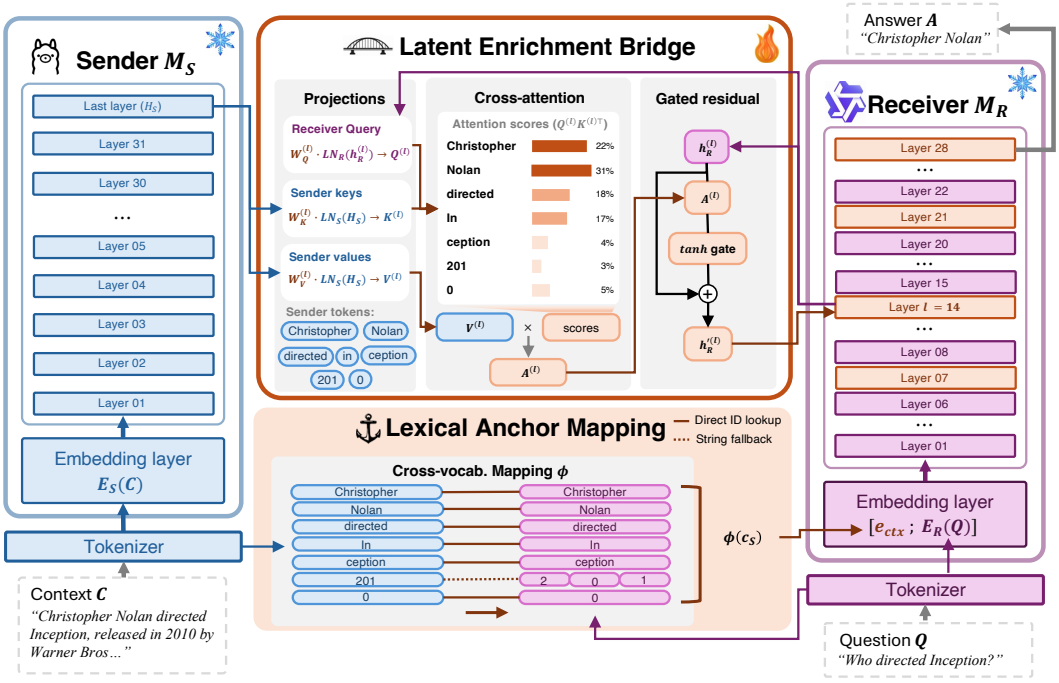


Figure 3: XBRIDGE overview. A single sender forward pass produces hidden states H_S and context tokens c_S. LAM maps c_S to receiver embeddings (entity grounding). LEB lets the receiver query H_S at four layers (contextual enrichment). The receiver generates the answer conditioned on both signals.

last-layer hidden states $H_S \in \mathbb{R}^{T_C \times d_S}$ and context token sequence $c_S = (c_1, \ldots, c_{T_C})$. The receiver M_R holds only the question Q and must generate an answer A from whatever signal the sender communicates. XBRIDGE transmits both H_S and c_S: LAM (Section 3.2) uses c_S, and LEB (Section 3.3) uses H_S.

3.2 Lexical Anchor Mapping (LAM) for Discrete Grounding

LAM addresses the entity grounding failure (Section 2.2) by placing the sender's original context tokens, including entity mentions, as receiver-native lexical inputs. These lexical anchors provide the discrete basis that LEB requires to ground its contextual enrichment to specific entities (Section 3.4).

Cross-vocabulary mapping. Given the sender's context tokens $c_S = (c_1, \ldots, c_{T_C})$ obtained by tokenizing C with M_S's tokenizer, each token $c_i \in \mathcal{V}_S$ is mapped to the receiver's vocabulary $\mathcal{V}_R$ via a deterministic, training-free mapping function $\phi : \mathcal{V}_S \to \mathcal{V}_R^*$, precomputed offline once per sender–receiver tokenizer pair. For tokens whose surface string appears in both vocabularies, ϕ performs a direct ID-to-ID lookup. For the remainder, a string fallback decodes the token to its surface string and re-tokenizes it with the receiver's tokenizer, producing one or more receiver tokens expanded in place. The mapping is lossless and adds negligible latency (<1 ms); detailed statistics and examples are in Appendix A.1.

Receiver-native lexical embeddings. The mapped token IDs are converted to receiver embeddings $e_{\text{ctx}} = (E_R[\phi(c_1)], \ldots, E_R[\phi(c_{T_C})])$ via the receiver's frozen embedding matrix E_R, omitting expansion from re-tokenization for clarity. These embeddings are prepended to the receiver's question input, following the input prepend convention used in soft-token communication [8]:

$$x_R = [e_{\text{ctx}} \; ; \; E_R(Q)] \tag{8}$$

By using E_R as a lookup table rather than training a separate projection, transmitted tokens are guaranteed to lie in the same space as the receiver's own input tokens, ensuring that pretrained self-attention patterns apply directly to the transferred context. Unlike prior methods that require the

sender to autoregressively decode a summary [4, 8], LAM transfers the full context without decoding, eliminating both latency and information loss from fixed-length compression.

3.3 Latent Enrichment Bridge (LEB) for Context Adaptation

LAM preserves entity identity but transmits only the sender's raw tokens, not the sender's *contextual understanding* of those tokens. The sender's last-layer hidden states H_S encode information integrated across all context positions through the sender's full transformer stack, information that raw token embeddings cannot convey. We use the last layer because it produces the most globally-integrated representations, as confirmed by a layer sweep in Appendix B.1. However, H_S lies in the sender's representation space, which the receiver has never encountered during training (Section 2.3). Directly injecting these representations, whether by prepending hidden states or replacing KV caches, places out-of-distribution vectors into the receiver's processing pipeline and degrades performance. *Cross-attention* provides a natural solution: rather than injecting sender representations into the receiver, the receiver queries the sender in its own native space. This receiver-driven mechanism has three stages, illustrated in the bridge panel of Figure 3.

Projections. We insert $M{=}4$ gated cross-attention modules into the receiver, one after each of layers $\ell \in \{7, 14, 21, 28\}$ of the receiver's 28 layers, following the insertion density of Flamingo [9]. Keeping the backbone frozen and training only lightweight inserted modules follows the parameter-efficient adaptation practice used across domains [10]. We find four modules optimal; fewer provide insufficient capacity while more overfit with limited training data (Appendix B.1). Each module has its own learnable parameters ($\sim$66M each, $\sim$264M total, 3.8% of the receiver). Both models remain frozen; only the bridge modules are trained. At each insertion layer ℓ, the receiver's hidden state $h_R^{(\ell)} \in \mathbb{R}^{d_R}$ is projected into a query, while the sender's hidden states H_S are projected into keys and values:

$$Q^{(\ell)} = W_Q^{(\ell)} \cdot \mathrm{LN}_R^{(\ell)}(h_R^{(\ell)}), \quad K^{(\ell)} = W_K^{(\ell)} \cdot \mathrm{LN}_S^{(\ell)}(H_S), \quad V^{(\ell)} = W_V^{(\ell)} \cdot \mathrm{LN}_S^{(\ell)}(H_S) \tag{9}$$

where $W_Q^{(\ell)} \in \mathbb{R}^{d_R \times d_R}$, $W_K^{(\ell)}, W_V^{(\ell)} \in \mathbb{R}^{d_R \times d_S}$, and $\mathrm{LN}_R^{(\ell)}$, $\mathrm{LN}_S^{(\ell)}$ are separate layer normalizations for the receiver and sender. The projection matrices simultaneously handle the dimensional mismatch ($d_S \to d_R$), serving as the learned translation between representation spaces. All four modules query the same H_S, but each has its own parameters, enabling different layers to extract different aspects of the sender's representation.

Cross-attention. The receiver's query selects relevant sender positions through scaled dot-product attention:

$$A^{(\ell)} = \mathrm{softmax}\Big(Q^{(\ell)} K^{(\ell)\top} / \sqrt{d_k}\Big) \cdot V^{(\ell)} \tag{10}$$

This is the key advantage over injection-based alternatives: the receiver decides what information to retrieve based on its current processing state at layer ℓ, rather than receiving a fixed representation regardless of what it needs.

Gated residual. The cross-attention output is added to the receiver's hidden state via a gated residual connection, preserving the receiver's pretrained representations while incorporating bridge information:

$$h_R'^{(\ell)} = h_R^{(\ell)} + \tanh(\alpha^{(\ell)}) \cdot A^{(\ell)} \tag{11}$$

where $h_R'^{(\ell)}$ is the updated hidden state that continues through the receiver's subsequent layers. We initialize $\alpha^{(\ell)} = 1.0$ ($\tanh(1.0) \approx 0.76$) so that the bridge contributes immediately, unlike Flamingo's zero initialization which requires the model to discover the bridge signal during training. With our limited training budget, warm initialization converges faster (Appendix B.1).

The bridge is trained with standard next-token prediction loss on the answer tokens:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{|A|} \log P_R(a_t \mid a_{<t}, e_{\text{ctx}}, Q; \theta_{\text{bridge}}) \tag{12}$$

where e_{ctx} is computed via LAM (Section 3.2). Since the sender is frozen, its outputs (H_S and context token IDs) are precomputed and cached, reducing training to under 10 minutes on a single GPU

Table 1: Heterogeneous communication results (F1 %). The best score per task within each model pair is bolded and the Improv. row shows the gain of XBRIDGE over the second-best method, both excluding FullComm.

Method	Countries	Tipsheets	HotpotQA	QASPER	MuSiQue	MFldQA	2Wiki	Avg
M_S: Llama-3.1-8B-Instruct; M_R: Qwen2.5-7B-Instruct								
NoComm	0.0	66.5	24.6	7.1	10.6	17.3	19.3	20.8
FullComm	50.8	98.5	78.4	31.3	31.5	48.2	47.6	55.2
NLComm$_{\text{hetero}}$	22.6	96.3	68.5	33.7	32.5	24.8	14.2	41.8
XBRIDGE	**72.5**	**99.8**	**78.8**	**47.5**	**48.2**	**44.2**	**51.1**	**63.2**
Improv.	+49.9	+3.5	+10.3	+13.8	+15.7	+19.4	+36.9	+21.4
M_S: Qwen2.5-7B-Instruct; M_R: Llama-3.1-8B-Instruct								
NoComm	0.0	11.0	31.4	3.8	2.6	11.2	12.9	10.4
FullComm	67.0	95.0	82.9	36.7	41.3	53.3	51.8	61.2
NLComm$_{\text{hetero}}$	56.2	94.8	76.8	34.1	33.5	21.7	16.3	47.6
XBRIDGE	**62.0**	**99.0**	**80.6**	**46.6**	**62.8**	**40.3**	**42.1**	**61.9**
Improv.	+5.8	+4.2	+3.8	+12.5	+29.3	+18.6	+25.8	+14.3
M_S: Mistral-7B-Instruct-v0.3; M_R: Qwen2.5-7B-Instruct								
NoComm	0.0	66.5	24.6	7.1	10.6	17.3	19.3	20.8
FullComm	50.8	98.5	78.4	31.3	31.5	48.2	47.6	55.2
NLComm$_{\text{hetero}}$	43.6	96.0	66.1	31.8	28.3	27.0	15.9	44.1
XBRIDGE	**70.0**	**100.0**	**84.9**	**50.6**	**65.0**	**35.8**	**48.7**	**65.0**
Improv.	+26.4	+4.0	+18.8	+18.8	+36.7	+8.8	+32.8	+20.9

on a balanced set of 587 samples drawn across the seven task domains. Once trained, the bridge is fixed and applied to all tasks at inference time without further adaptation (Section 4). Training data composition is analyzed in Appendix B.2. The bridge adapts to different model pairs by adjusting only the projection dimensions; we train separate instances for each pair using the same protocol.

3.4 Entity-Grounded Bridge Integration

The two mechanisms are integrated through the receiver's own processing, without any explicit fusion layer. LAM places entity mentions (e.g., "Christopher Nolan") in the receiver's input as discrete embeddings. As these propagate through the receiver's layers via self-attention, they shape the receiver's hidden states $h_R^{(\ell)}$ at every depth, embedding entity information into the representations that LEB modules use as queries. This is the grounding mechanism: when LEB at layer ℓ computes $Q^{(\ell)} = W_Q^{(\ell)} \cdot \mathrm{LN}_R^{(\ell)}(h_R^{(\ell)})$, the query already encodes which entities are present and where they appear, enabling the cross-attention to focus on the sender positions that carry relevant contextual information about those entities. Without LAM, $h_R^{(\ell)}$ reflects only the question, producing queries too vague to extract entity-specific signals from the sender, so the LEB conveys contextual signals while entity identity is lost in the continuous bottleneck, which is why LEB-only performance drops to $\sim$30% F1 (Section 4.3). Without LEB, the receiver has the sender's context tokens but lacks the sender's contextual understanding of their relationships. Together, LEB's contextual enrichment is anchored to specific entities rather than floating in continuous space. We verify this complementarity quantitatively in Section 4.3.

4 Experiments

4.1 Experimental Setup

Models. We evaluate communication across three model families to test cross-architecture generality: Llama-3.1-8B-Instruct [11] (Llama), Qwen2.5-7B-Instruct [12] (Qwen), and Mistral-7B-Instruct-v0.3 (Mistral). These families differ in architecture (hidden dimension, layer count, attention structure), tokenizer design, and training data, ensuring that improvements are not an artifact of a single architecture pairing. We evaluate three heterogeneous pairs: Llama→Qwen, Qwen→Llama, and Mistral→Qwen. We also verify that the framework generalizes to same-architecture settings (Ap-

Table 2: Entity perturbation experiment on 100 HotpotQA samples. Each sample's answer entity is replaced with an unrelated entity (e.g., "Christopher Nolan" → "Bong Joon Ho"). LAM and LEB inputs are independently swapped to test which mechanism determines entity identity. Output is classified by which entity appears in the generated answer.

Condition	LAM	LEB	→ original	→ swapped	→ neither
A: Original	original	original	45	0	55
B: Both swapped	swapped	swapped	4	37	59
C: LAM swapped only	swapped	original	5	37	58
D: LEB swapped only	original	swapped	45	0	55

pendix C.1). To test whether larger senders produce richer representations for LEB, we additionally evaluate sender size scaling with Qwen2.5-{1.5B, 7B, 14B}-Instruct→Llama.

Tasks and protocol. We evaluate on seven benchmarks: HotpotQA [13], MuSiQue [14], QASPER [15], 2WikiMultihopQA (2Wiki) and MultiFieldQA-en (MFldQA) from LongBench [16], Countries and Tipsheets [5]. The sender receives the context document only and the receiver receives the question only, the asymmetric protocol established by Shi et al. [5], which we apply identically to every method compared here. For NLComm baselines, the sender generates a 128-token summary via greedy decoding. We report F1 score. Task details are in Appendix E.

Compared methods. We compare against natural language communication (NLComm) [4], where the sender generates a text summary and relays it to the receiver. We distinguish $\text{NLComm}_{\text{hetero}}$ (cross-family) and $\text{NLComm}_{\text{homo}}$ (same family). We also report two single-model reference points measured with the receiver alone. NoComm, where the receiver sees only the question, is a lower reference indicating what the receiver achieves without communication. FullComm, where the receiver reads the full context directly, removes the communication problem rather than solving it, and serves as an upper reference in the sense of the Skyline used by Shi et al. [5]. Neither is a competing communication method, and both are excluded from the best-per-task comparison. Same-architecture methods, including $\text{NLComm}_{\text{homo}}$ and KVComm [5] which shares KV caches between models of the same family, are compared in Appendix C.1.

4.2 Main Results

Table 1 presents heterogeneous communication results across all three model pairs and seven benchmarks. XBRIDGE outperforms $\text{NLComm}_{\text{hetero}}$ on all 7 tasks across all three model pairs, with average gains of +21.4 pp (Llama→Qwen), +14.3 pp (Qwen→Llama), and +20.9 pp (Mistral→Qwen). The improvements are largest on tasks requiring multi-hop reasoning or world knowledge (Countries, 2WikiMQA, MuSiQue), where LEB provides contextual enrichment that text summaries cannot convey. On average, XBRIDGE surpasses even FullComm across all three model pairs (63.2 vs. 55.2 for Llama→Qwen, 61.9 vs. 61.2 for Qwen→Llama, 65.0 vs. 55.2 for Mistral→Qwen). Since LAM places the full mapped context in the receiver's input, XBRIDGE sees what FullComm sees and adds the sender's hidden states through LEB, so the comparison is not one of indirect against direct access. The same-architecture setting isolates this: with Qwen2.5-7B sending to itself, LAM reduces to an identity mapping, so the receiver's input matches FullComm's and only LEB differs, and $\text{XBRIDGE}_{\text{homo}}$ still leads by 7.9 pp on average (Appendix C.1). With identical models and identical input, the gain comes from LEB carrying the sender's integrated representation that the receiver would otherwise recompute. The optimal sender varies by task: Mistral→Qwen achieves the highest scores on HotpotQA, QASPER, and MuSiQue, while Llama→Qwen leads on MFldQA and 2WikiMQA. This complementarity suggests that selecting the sender based on task characteristics can further improve performance. In the same-architecture setting, XBRIDGE also exceeds KVComm [5] on 6 of 7 tasks (Appendix C.1).

4.3 Ablation: Both Anchors and Enrichment Are Necessary

The gains reported above rely on both LAM and LEB working together. To understand how much each contributes, we remove them individually on HotpotQA (Table 3). XBRIDGE without LAM ($\text{XBRIDGE}_{\text{-LAM}}$) drops to 30.3% F1, close to NoComm: LEB receives continuous signals but without

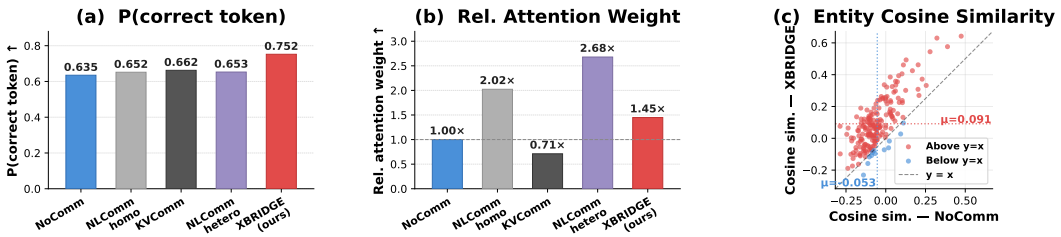


Figure 4: Bridge mechanism analysis on HotpotQA (Llama→Qwen, 196 samples). **(a)** Probability assigned to the correct answer token. **(b)** Relative attention weight on answer positions. NLComm attends most heavily yet predicts worst (cf. Figure 2b), revealing an attention-prediction disconnect. **(c)** Per-sample entity cosine similarity comparing NoComm and XBRIDGE. Points above $y{=}x$ indicate bridge improvement.

Table 3: Entity grounding ablation on HotpotQA, Llama→Qwen (F1 %).

Config.	F1	Δ
NoComm	24.6	–
XBRIDGE-LAM	30.3	+5.7
XBRIDGE-LEB	56.5	+31.9
XBRIDGE	**78.8**	+54.2

Table 4: Sender size scaling: Qwen → Llama, XBRIDGE (F1 %, 7-task avg).

Sender	Avg
Qwen2.5-1.5B	59.5
Qwen2.5-7B	61.9
Qwen2.5-14B	**62.1**

Table 5: Inference latency per sample on HotpotQA.

Method	s/sample	Rel. Latency
NoComm	0.09	1.0×
FullComm	0.13	1.4×
NLComm$_{\text{homo}}$	1.70	18.9×
NLComm$_{\text{hetero}}$	1.70	18.9×
KVComm	0.13	1.4×
XBRIDGE	**0.15**	**1.7**×

entity anchors these signals are not bound to specific entities, confirming the entity grounding failure identified in Section 2.2. XBRIDGE without LEB (XBRIDGE$_{\text{-LEB}}$) reaches 56.5%, providing entity grounding but no contextual enrichment beyond what the tokens themselves contain. The full XBRIDGE achieves 78.8%, exceeding the FullComm. The +22.3 pp gain from adding LEB to LAM contrasts with the modest +5.7 pp from LEB-only over NoComm, demonstrating that grounding is the enabling condition for LEB effectiveness. LEB contribution scales with task reasoning complexity: from +35.8 pp on factual inference requiring world knowledge (Countries) to +2.0 pp on structured extraction (Tipsheets), with per-task breakdown in Appendix C.2.

4.4 Bridge Mechanism and Role Separation

The ablation confirms that both mechanisms are necessary. A deeper question follows: does each carry a distinct type of information, or do they redundantly encode the same signal? To test this, we run an entity perturbation experiment on 100 HotpotQA samples, replacing every occurrence of the answer entity in the context with an unrelated entity of the same type (e.g., "Christopher Nolan" → "Bong Joon Ho"), then independently control which version each mechanism receives (Table 2).

Swapping LEB alone does not change the output entity: conditions A and D produce identical outputs (45 original, 0 swapped, 55 neither). The "neither" category contains outputs that mention neither the original nor the substituted entity, such as a date, another entity, or an unrelated span; we therefore interpret the experiment by comparing the original-versus-swapped direction among target-entity outputs. Under this interpretation, swapping LAM shifts the output toward the substituted entity regardless of which LEB version is used; conditions B and C produce nearly identical counts (37 swapped, 4–5 original, 58–59 neither). These results indicate a clear role separation: LAM supplies *which* entities appear in the output, while LEB supplies *how* the receiver reasons about them.

Figure 4 examines LEB's effect on the receiver's internal state across five communication conditions on 196 HotpotQA samples. NLComm methods attend most heavily to answer positions (2.0–2.7× above average) yet achieve the worst token rank because paraphrased text replaces exact entity mentions. XBRIDGE is the only method that produces positive cosine alignment between generation and answer entity representations, translating to correct-token probability of 0.752 vs. 0.635–0.662 for all other conditions (94% of individual samples, Appendix D.1).

4.5 Beyond Accuracy: Latency, Composability, and Sender Scaling

XBRIDGE achieves 11× lower latency than NLComm (0.15 s vs. 1.70 s per sample on HotpotQA) by eliminating autoregressive decoding: the sender performs a single forward pass (∼0.05 s) with negligible bridge overhead (<0.01 s), as reported in Table 5. The modular bridge design also enables zero-shot composability: two independently trained bridges (Llama and Mistral as senders) combine at inference without retraining, achieving 70.4% F1 on HotpotQA compared to 67.0% for single-sender full context and 56.8% for dual NLComm (Appendix C.3). Finally, LAM is sender-size invariant since it depends only on the vocabulary mapping, not the sender's capacity: even a 1.5B sender achieves 59.5%, approaching FullComm, while the 7B and 14B senders surpass it (Table 4).

5 Conclusion

In this work, we identified the entity grounding problem in heterogeneous LLM communication, where continuous bridges lose entity identity in the latent bottleneck. We proposed XBRIDGE, a decode-free protocol that addresses this by combining LAM with LEB, decomposing communication into discrete entity anchors and continuous contextual enrichment. Extensive experiments across three model families, seven benchmarks, and both communication directions demonstrated that XBRIDGE outperforms NLComm on every task with 11× lower latency. We highlight the complementarity of the two mechanisms: LAM determines which entities appear in the output, while LEB determines how the receiver reasons about them, as confirmed by the entity perturbation experiment. Our work shows that heterogeneous LLMs can communicate at the latent level while preserving entity fidelity, and we see this as an early step toward cross-architecture multi-agent collaboration.

Limitations

XBridge is trained for a single sender to receiver direction, so one bridge carries information one way over one exchange. A two-way setup would add a second bridge for the reverse direction, and since each bridge is lightweight and decode-free, the cost per exchange would stay low, but multi-turn latent dialogue, where the two models alternate and each turn conditions on what the other sent, is beyond what we evaluate here. The same holds for the number of agents: we have tested composition of two independently trained bridges, leaving open how bridge signals interact when more agents participate. Extending the protocol along both directions, bidirectional exchange and many-agent composition, is what we see as the path toward a cross-architecture communication protocol suited to full multi-agent systems.

Acknowledgments and Disclosure of Funding

This work used the Delta system at the National Center for Supercomputing Applications through allocation CIS251025 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program. The work of Junhyun Lee was supported by the Hankuk University of Foreign Studies Research Fund (2026) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25604335).

Competing Interests. The authors declare no competing interests.

References

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

[2] Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for LLM judges with adaptive stability detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=Vusd1Hw2D9`.

[3] Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iUjGNJzrF1.

[4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*, 2024.

[5] Xiangyu Shi, Marco Chiesa, Gerald Q. Maguire Jr., and Dejan Kostic. KVComm: Enabling efficient LLM communication through selective KV sharing. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=F7rUng23nw.

[6] Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=W6RPXUUFic.

[7] 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. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LeatkxrBCi.

[8] Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A. Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sehRvaIPQQ.

[9] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[10] Junhyun Lee, Wooseong Yang, and Jaewoo Kang. Subgraph-level universal prompt tuning. *Information Sciences*, page 123516, 2026.

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

[12] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[13] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[14] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[15] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, 2021.

[16] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

[17] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: a survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/890. URL https://doi.org/10.24963/ijcai.2024/890.

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

[19] Wei-Chieh Huang, Henry Peng Zou, Yaozu Wu, Dongyuan Li, Yankai Chen, Weizhi Zhang, Yangning Li, Angelo Zangari, Jizhou Guo, Chunyu Miao, Liancheng Fang, Langzhou He, Yinghui Li, Renhe Jiang, and Philip S. Yu. Deep research with open-domain evaluation and multi-stage guardrails for safety. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 43403–43446, 2026.

[20] Wei-Chieh Huang, Weizhi Zhang, Yueqing Liang, Yuanchen Bei, Yankai Chen, Tao Feng, Zhen Tan, Yu Wang, Tianxin Wei, Shanglin Wu, et al. A survey of agent memory in the second half: Towards self-evolving and long-horizon agents. *Transactions on Machine Learning Research*, 2026.

[21] Wei-Chieh Huang, Weizhi Zhang, Yuchen Wu, Yankai Chen, Eric Hanchen Jiang, Wooseong Yang, Yiwei Yang, Henry Peng Zou, Hanrong Zhang, Ying Nian Wu, et al. Harness the memory: A holistic evaluation of memory substrates in memory agents. *arXiv preprint arXiv:2608.15008*, 2026.

[22] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904, 2024.

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

[24] 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.

[25] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186, 2024.

[26] Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Guanting Dong, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026.

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

A LAM Details

A.1 Cross-Vocabulary Mapping

Table 6 summarizes the vocabulary overlap and token transfer statistics for the two sender–receiver pairs used in our experiments.

Table 7 shows representative examples of tokenization mismatches and how the string-based fallback resolves them.

The mismatch patterns reflect fundamental differences in tokenizer design. Llama-3.1 uses a 128K BPE vocabulary that overlaps substantially with Qwen2.5's 152K vocabulary, resulting in mismatches confined almost entirely to multi-digit numbers. Mistral-7B uses a smaller 32K SentencePiece vocabulary, causing common English words and proper nouns to fall back to string-based re-tokenization. In both cases, the fallback is lossless: the surface string content is fully preserved, with only positional expansion as a side effect.

Table 6: Cross-vocabulary mapping statistics on HotpotQA (100 validation samples). Direct-mapped tokens use ID-to-ID lookup; the remainder use string-based fallback.

	Llama-3.1 → Qwen2.5	Mistral-7B → Qwen2.5
Sender vocab size	128,256	32,768
Receiver vocab size	151,665	151,665
Shared tokens	109,566 (85.4%)	10,566 (32.2%)
Direct-mapped	97.1%	87.8%
String fallback	2.9%	12.2%
Dominant mismatch type	Multi-digit numbers	Words + proper nouns

Table 7: Examples of cross-vocabulary mapping between different tokenizer pairs. Direct lookup preserves the token as-is; string fallback re-tokenizes the surface string, producing one-to-many expansions.

Source text	Sender tokens	Receiver tokens	Type
Llama-3.1 → Qwen2.5			
"the"	[the] (1)	[the] (1)	Direct
"transformer"	[transform, er] (2)	[transform, er] (2)	Direct
"199"	[199] (1)	[1, 9, 9] (3)	Fallback
"COVID-19"	[COVID, -, 19] (3)	[COVID, -, 1, 9] (4)	Mixed
Mistral-7B → Qwen2.5			
"directed"	[directed] (1)	[direct, ed] (2)	Fallback
"Angeles"	[Angeles] (1)	[An, ge, les] (3)	Fallback
"century"	[century] (1)	[cent, ury] (2)	Fallback

A.2 Grounding Format Independence

A critical question is whether LEB's effectiveness depends on the specific grounding format (embedding prepend vs. text prompt) or merely on the *presence* of entity anchors.

LEB consistently improves all grounding formats ($+12.5$ to $+22.3$ pp), demonstrating that entity presence, not format, is what enables LEB. Whether entity anchors arrive as prepended embeddings, as text within a chat template, or as argmax-decoded tokens, LEB extracts contextual enrichment from the sender's H_S via cross-attention regardless of how those anchors enter the receiver.

The smaller Δ for NLComm ($+12.5$ pp) reflects its higher base: NLComm text already contains sender reasoning, leaving less room for LEB improvement. Notably, NLComm with LEB (81.0%) surpasses LAM with LEB (78.8%), because the sender's autoregressive decoding performs reasoning compression that benefits the receiver. However, LAM remains the preferred grounding format because it eliminates decoding latency entirely ($14\times$ faster than NLComm) while achieving comparable performance with LEB.

B LEB Design

B.1 Bridge Architecture Ablations

We ablate bridge design decisions on HotpotQA (Llama→Qwen). All variants use balanced training (587 samples).

Number of bridge modules. We vary the number of cross-attention modules inserted into the receiver (Qwen2.5-7B, 28 layers). 4 modules (264M parameters) is optimal. 7 modules (462M) overfits due to the higher parameter-to-data ratio with balanced training. 2 modules (132M) provides insufficient capacity for cross-architecture translation. This follows Flamingo's [9] finding that sparser insertion is preferred for larger models.

Table 8: Grounding format comparison on HotpotQA, Llama→Qwen (F1 %). Each grounding format is evaluated with and without LEB to isolate the bridge contribution (Δ).

Grounding format	Without LEB	With LEB	LEB Δ
LAM	56.5	78.8	+22.3
NLComm text (chat prompt)	68.5	81.0	+12.5
Argmax token map	44.5	65.5	+21.0

Table 9: Bridge architecture ablations on HotpotQA, Llama→Qwen (F1 %). All variants use balanced training (587 samples).

Ablation	Variant	F1
Modules	1 module (66M)	66.3
	2 modules (132M)	74.6
	4 modules (264M)	**78.8**
	7 modules (462M)	75.8
Sender layer	Layer 8	64.6
	Layer 16	66.7
	Layer 24	70.2
	Layer 32 (last)	**78.8**
Gate init	$\alpha = 0.0$ (Flamingo)	72.7
	$\alpha = 1.0$ (ours)	**78.8**
Loss	NTP only (ours)	**78.8**
	NTP + contrastive	78.7

Sender layer selection. Bridge performance increases monotonically from early to final sender layers, with layers before the midpoint falling below the no-bridge baseline. The last layer is optimal because it integrates information across all context positions through the sender's full attention stack, producing the most globally-aware representations.

Gate initialization. We initialize the $\tanh$ gate at $\alpha = 1.0$ ($\tanh(1.0) \approx 0.76$), differing from Flamingo's zero initialization. With $\alpha = 0$, the bridge is initially invisible to the receiver, requiring training to "discover" the bridge signal. With $\alpha = 1.0$, the bridge immediately contributes cross-architecture context. In our setting with limited training data (587 samples), warm initialization converges faster and achieves higher final performance.

Auxiliary losses. Adding a contrastive loss (encouraging the bridge to distinguish correct sender context from random contexts) does not improve performance beyond NTP loss alone. The NTP objective already implicitly requires the bridge to extract useful information from the sender's hidden states.

B.2 Bridge Training Data Composition

The composition of bridge training data significantly affects per-task performance.

HotpotQA-only bridge (20K samples) is inferior even on its own training domain (76.1% vs. 78.8% for the balanced bridge), and catastrophically degrades on unseen task types: Countries drops to 27.7% and QASPER to 36.2%. Overfitting to a single task's context-length distribution and reasoning pattern prevents the bridge from generalizing.

Unbalanced universal bridge (42K samples, 94% HotpotQA+MuSiQue) improves over the single-task bridge across all tasks but remains below the balanced bridge on average (59.3% vs. 63.2%). The heavy skew toward multi-hop QA patterns limits generalization to dissimilar tasks such as Countries and QASPER.

Balanced universal bridge (587 samples, ∼100 per task) achieves the best performance on every metric despite using 30× less training data than the unbalanced variant. The balanced bridge learns task-appropriate gating: it remains near-closed for near-saturated factual tasks (Tipsheets 99.8%)

Table 10: Effect of training data composition on LEB performance (Llama→Qwen, F1 %).

Bridge training	HotpotQA	Tipsheets	Countries	QASPER	Avg (7 tasks)
HotpotQA only (20K)	76.1	88.3	27.7	36.2	49.5
Unbalanced (42K)	77.1	100.0	66.0	45.3	59.3
Balanced (587)	78.8	99.8	72.5	47.5	63.2

Table 11: Same-architecture comparison (F1 %). XBRIDGE$_{\text{homo}}$ uses Qwen2.5-7B as both sender and receiver with a separately trained bridge. FullComm = receiver reads full context directly (identical to Table 1). **Bold** = best per task (excluding FullComm). Improv. = improvement of XBRIDGE$_{\text{homo}}$ over the second-best method (excluding FullComm).

Method	**Countries**	**Tipsheets**	**HotpotQA**	**QASPER**	**MuSiQue**	**MFldQA**	**2Wiki**	**Avg**
			M_S: Qwen2.5-7B-Instruct; M_R: Qwen2.5-7B-Instruct					
FullComm	50.8	98.5	78.4	31.3	31.5	48.2	47.6	55.2
NLComm$_{\text{homo}}$	37.2	96.0	64.9	30.8	29.1	23.9	18.2	42.9
KVComm	35.8	99.0	65.3	35.5	33.1	**48.3**	40.9	51.1
XBRIDGE$_{\text{homo}}$	**72.7**	**99.8**	**87.1**	**49.0**	**38.7**	45.6	**48.9**	**63.1**
Improv.	+35.5	+0.8	+21.8	+13.5	+5.6	−2.7	+8.0	+12.0

while actively contributing on complex reasoning tasks (HotpotQA 78.8%, Countries 72.5%). The average across all 7 tasks (63.2%) is the highest among all bridge variants.

Implication for sender size scaling. The balanced training set (587 samples) is sufficient for 7B senders but becomes a bottleneck for larger models. The 14B sender (hidden dim 5120) produces richer representations that require more training data to learn effective projections. Within this data budget, the 14B bridge plateaus at 62.1%, matching the 7B sender (61.9%) despite its larger capacity. We expect that scaling the balanced training set would unlock the 14B sender's advantage, but leave this to future work.

C Additional Results

C.1 Homogeneous Comparison

To verify that XBRIDGE generalizes beyond heterogeneous settings, we train a same-architecture bridge using Qwen2.5-7B as both sender and receiver. LAM reduces to an identity mapping in this setting, so the comparison isolates LEB's contribution on top of full context reading.

Table 11 compares XBRIDGE$_{\text{homo}}$ against NLComm$_{\text{homo}}$ and KVComm [5]. XBRIDGE$_{\text{homo}}$ outperforms NLComm$_{\text{homo}}$ on all 7 tasks and KVComm on 6 of 7 tasks, with an average improvement of +12.0 pp over the stronger baseline. Notably, XBRIDGE$_{\text{homo}}$ surpasses even FullComm on 6 of 7 tasks (+7.9 pp on average), demonstrating that LEB adds value beyond what the receiver extracts from reading the context alone. The sole exception is MultiFieldQA, consistent with the pattern observed in heterogeneous settings (Section 4.3).

C.2 Bridge Contribution per Task

Table 12 breaks down LEB's contribution across all seven tasks for the Llama→Qwen pair. LEB contribution scales with the reasoning complexity of the task.

On Countries, the context states a person's location at a landmark (e.g., "Eve is at the Uppland Runic Inscription 896") and the question asks which country the person is in, but the answer ("Sweden") appears nowhere in the context. Only LEB can transmit the sender's implicit landmark-to-country inference, yielding the largest contribution (+35.8 pp). On multi-hop tasks (HotpotQA, 2WikiMQA, MuSiQue), LEB transmits reasoning chains integrated across documents (+19 to +22 pp).

On MultiFieldQA, the question asks for a specific fact (e.g., "What is the advantage of decorrelating the data before running the PLS algorithm?") and the answer is a single extractable sentence located

Table 12: Bridge contribution per task (Llama→Qwen, F1 %). XBRIDGE$_{\text{-LEB}}$ = LAM only. Δ = XBRIDGE − XBRIDGE$_{\text{-LEB}}$.

Task	Type	XBRIDGE$_{\text{-LEB}}$	XBRIDGE	Δ
Countries	factual inference	36.7	72.5	+35.8
HotpotQA	multi-hop QA	56.5	78.8	+22.3
2WikiMQA	multi-hop QA	29.5	51.1	+21.6
MuSiQue	compositional QA	28.4	48.2	+19.8
QASPER	scientific QA	33.0	47.5	+14.5
MFldQA	long-doc retrieval	37.3	44.2	+7.0
Tipsheets	structured extract.	97.8	99.8	+2.0

Table 13: Multi-agent communication on HotpotQA, Llama→Qwen (F1 %). Two senders each process one supporting document; the receiver integrates both.

Configuration	F1
Single sender, full context (Llama)	67.0
Dual NLComm (Llama + Mistral)	56.8
Dual XBRIDGE (Llama + Mistral)	**70.4**

deep within a ∼7K-token document. No cross-entity reasoning is needed, and LEB adds only +7.0 pp. This is also the sole task where KVComm [5] outperforms XBRIDGE (48.3 vs. 44.2): same-architecture KV sharing preserves position-level attention patterns that directly encode answer location, an advantage specific to retrieval tasks that disappears when reasoning is required (e.g., HotpotQA +13.5 pp, MuSiQue +15.1 pp over KVComm).

On Tipsheets, the answer is directly extractable from structured text, leaving minimal room for LEB improvement (+2.0 pp).

C.3 Multi-Agent Communication

We test composability on HotpotQA, where each question requires reasoning over two supporting documents. We assign each document to a separate sender, Llama for the first and Mistral for the second, and let the receiver (Qwen) integrate both via their respective bridges. Each bridge was trained independently for its own model pair; no joint training or adaptation is performed.

Dual XBRIDGE achieves 70.4% F1 (Table 13), outperforming both the single-sender full-context baseline (67.0%, +3.4 pp) and dual NLComm (56.8%, +13.6 pp). The single-sender baseline gives one Llama sender both documents, so the dual setup's improvement demonstrates that distributing documents across specialized senders is more effective than compressing everything through a single sender. This zero-shot composability, combining independently trained bridges without additional training, suggests that the framework scales naturally to multi-party agent settings.

D Additional Analysis

D.1 Per-Sample Consistency

The bridge mechanism analysis in Section 4.4 reports aggregate metrics across 196 HotpotQA samples (Llama→Qwen). Here we verify that these improvements are consistent at the individual sample level rather than driven by a few outliers.

Figure 4(c) plots the mean-centered cosine similarity between the generation representation and the answer entity representation for each sample, comparing NoComm and XBRIDGE. Points above the $y = x$ line indicate samples where LEB improves alignment. Of 196 samples, 184 (94%) fall above $y = x$ ($p \approx 0$, paired t-test). The population mean shifts from $\mu = -0.053$ (NoComm) to $\mu = +0.091$ (XBRIDGE), a sign change indicating that without LEB the receiver's generation state moves away from the answer entity, while with LEB it moves toward it.

The 12 samples (6%) where LEB does not improve alignment tend to involve short, unambiguous contexts where LAM alone already provides sufficient grounding. Even in these cases, LEB does not substantially degrade performance, indicating that the gating mechanism (Equation 11) successfully attenuates the bridge signal when it is not needed.

D.2 Entity Perturbation Examples

Table 14 shows representative examples from the entity perturbation experiment (Section 4.4). In each case, swapping LEB input alone does not change the output entity, while swapping LAM changes it entirely.

Table 14: Representative entity perturbation examples on HotpotQA (Llama→Qwen). The answer entity in the context is replaced with an unrelated entity of the same type. Condition A: both original. Condition D: LEB swapped only. Condition C: LAM swapped only.

Question	Original	Swapped	A	D	C
Which port city lies ∼25 km north of the Lingnan Fine Arts Museum?	Keelung	Majuro	Keelung	Keelung	Majuro
Who did Neleus of Scepsis study under in addition to Theophrastus?	Aristotle	Moroni	Aristotle	Aristotle	Moroni
Who will Billy Howle be seen opposite in the upcoming British drama film directed by Dominic Cooke?	Saoirse Ronan	Kenji Mizoguchi	Saoirse Ronan	Saoirse Ronan	Kenji Mizoguchi

In all three examples, conditions A and D produce identical outputs, confirming that LEB does not influence entity selection. Condition C shifts the output to the swapped entity every time. LEB adapts its contextual enrichment accordingly: contextual signals such as "port city 25 km north of the museum" or "studied under in the Peripatetic school" bind to whichever entity LAM provides.

E Task Details

Table 15 summarizes the seven benchmarks used in our evaluation. Training and evaluation splits are strictly separated with zero overlap.

Table 15: Task details. Context length statistics are measured across evaluation samples using the Llama-3.1 tokenizer.

Task	Type	Eval N	Median tok	Mean tok	Source
HotpotQA	multi-hop QA	200	74	80	Yang et al. [13]
MuSiQue	compositional QA	200	∼1K*	∼1.2K*	Trivedi et al. [14]
QASPER	scientific QA	200	4,118	4,514	Dasigi et al. [15]
2WikiMQA	multi-hop QA	100	6,532	7,267	LongBench [16]
MultiFieldQA-en	long-doc retrieval	75	6,979	6,878	LongBench [16]
Countries	factual inference	100	11	11	Shi et al. [5]
Tipsheets	structured extract.	200	63	62	Shi et al. [5]

*MuSiQue statistics shown are for supporting paragraphs only; full context with distractors is approximately 4–5× longer. All evaluations use the full context.

F Related Work

Text-based multi-agent communication. LLMs are increasingly deployed in multi-agent systems where agents collaborate to solve tasks beyond individual capability [17, 18, 19]. A parallel line asks what an agent should retain across interactions [20, 21], whereas we focus on what is transferred within a single exchange. Multi-agent debate improves factuality by having models critique each other's reasoning [4, 22], while frameworks such as CAMEL [23], AutoGen [24], and ChatDev [25] coordinate agents through structured natural language exchanges. Natural language communication (NLComm) [4] generalizes this pattern: one model generates a text summary and relays it to another. These text-based protocols are architecture-agnostic but require autoregressive decoding at every

communication step, incurring high latency. They also discard the sender's internal representations, compressing context into a short summary that loses both the sender's contextual understanding and relational structure among entities.

Latent multi-agent communication. To avoid the cost and information loss of text-based communication, recent work transmits internal model representations directly [26]. CIPHER [8] communicates via soft tokens across models but requires a shared tokenizer. AC [6] shows that activation-space interventions transfer between LLMs of the same family. C2C [7] projects and fuses KV caches across model families through learned linear projections, achieving cross-architecture transfer but operating entirely in continuous space. KVComm [5] shares selectively chosen KV cache layers, achieving near-FullComm performance with reduced communication cost, but assumes matching hidden dimensions, attention heads, and positional encodings. These methods differ not only in what they transmit but in what they assume the two models share. C2C fuses caches position-wise and therefore has the sharer and receiver process the same context, and LatentMAS [1] concatenates layer-wise KV caches across agents under a common architecture. MoT [27] routes a query to a subset of frozen heterogeneous experts and lets a designated primary expert cross-attend to its peers' projected hidden states at uniformly placed interaction layers; all experts encode the same query with their own tokenizers, so the sequences stay aligned and the primary expert already holds the input. All three are symmetric in the sense that the receiver already holds the context. XBridge operates in the asymmetric setting of KVComm, where the sender observes a context the receiver does not, and it is this asymmetry that gives rise to the entity grounding problem: once the receiver can no longer see the entities directly, a discrete channel aligned to its own vocabulary becomes necessary rather than supplementary. Existing latent protocols therefore either assume architectural homogeneity, require a shared tokenizer, or place the two models on the same context, and none preserves discrete entity identity across different vocabularies. XBridge addresses this gap by enabling latent communication across different LLM architectures while preserving entity fidelity.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state three contributions (entity grounding problem formulation, XBridge protocol, empirical results across three model families and seven benchmarks) that are supported by experimental results in Section 4 and Appendix C.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed throughout the paper: the 14B sender bottleneck due to limited training data (Appendix B.2), MultiFieldQA as the sole task where KVComm outperforms XBridge (Section 4.3), and the current evaluation scope of 7B-scale models across three families.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not include formal theorems or proofs. Section 2 provides a formal problem setup and design principles but does not claim theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All model names, hyperparameters, training details (587 balanced samples, NTP loss, bridge architecture), and evaluation protocol (F1 score, 7 benchmarks) are specified in Sections 3–4 and Appendix B. Code is provided in our public repository, `https://github.com/WooseongYang/XBridge`.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Code is available at `https://github.com/WooseongYang/XBridge`. All datasets used are publicly available (HotpotQA, MuSiQue, QASPER, LongBench, Countries, Tipsheets).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Training details (Section 3.3, Appendix B), model specifications (Section 4.1), task details (Appendix E), and bridge architecture ablations (Appendix B.1) are fully described.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We do not report error bars because bridge training is deterministic given the fixed precomputed sender outputs and balanced training set. Per-sample consistency is analyzed in Appendix D.1 (94% of samples improve, $p \approx 0$ paired t-test).

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

Justification: Bridge training requires $\sim$10 minutes on a single H200 GPU with 587 samples (Section 3.3). Inference latency is reported per sample in Table 5. All experiments were conducted on NVIDIA H200 GPUs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics. All models used are publicly released, and the work focuses on improving communication efficiency between LLMs.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This work is foundational research on inter-LLM communication protocols. It does not introduce new generative capabilities or enable applications beyond what the underlying frozen LLMs already support.

Guidelines:

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

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release pretrained language models or scraped datasets. The bridge modules are lightweight adapters (264M parameters) that function only in conjunction with existing frozen LLMs.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All models (Llama-3.1, Qwen2.5, Mistral-7B) and datasets (HotpotQA, MuSiQue, QASPER, LongBench) are publicly available and properly cited in Sections 4.1 and Appendix E.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

Justification: We release the XBridge codebase including training and evaluation scripts in our public repository, `https://github.com/WooseongYang/XBridge`. The README provides setup and reproduction instructions.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

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

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

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

Answer: [N/A]

Justification: This work does not involve human subjects research.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLMs (Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, Mistral-7B-Instruct-v0.3) are core components of the proposed method, serving as frozen sender and receiver agents. All models are publicly available and fully specified in Section 4.1.

Guidelines:

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