LARC: Latent Communication with Rationale Anchoring for Efficient Small-Large Model Reasoning

Anonymous ACL submission

Abstract

Chain-of-thought reasoning improves large language model performance, but explicit textual traces are expensive to generate and transmit. Latent soft-thought methods reduce this overhead by moving reasoning into continuous hidden states, yet they leave open what reasoning content compact latent messages should preserve, especially when a frozen sender model communicates with a frozen receiver through a trainable bridge.

We introduce **LARC**, a rationale-anchored latent communication framework for small-large model collaboration. LARC uses explicit rationale steps as training-time communication contracts that specify what compact sender-to-receiver messages should cover, rather than as inference-time text to generate. The framework combines channel calibration across heterogeneous frozen models, rationale anchoring for reasoning coverage, and receiver-side reasoning supervision to ensure that latent messages remain useful for answering.

Across LLaMA and Qwen model families, LARC improves average accuracy over both SoftCoT and an unanchored latent communication baseline. Controlled corruption experiments show that rationale anchors are meaningful contracts: corrupting the anchor degrades performance even when the receiver-side target remains correct. Compared with explicit assistant-CoT communication, LARC transmits only four latent-message slots and achieves roughly $1.5\times$ faster inference. These results show that latent communication between language models benefits from explicit reasoning contracts.

1 Introduction

Chain-of-thought reasoning improves the problem-solving ability of large language models by externalizing intermediate reasoning steps in natural language (Wei et al., 2022; Kojima et al., 2022). This explicitness is useful, but it is also expensive: a model must generate and attend to long reasoning traces before producing a final answer. Recent latent reasoning methods attempt to reduce this cost by moving part of the reasoning process from text tokens to continuous hidden states (Hao et al., 2024; Cheng and Van Durme, 2024; Xu et al., 2025). In parallel, small-large collaboration methods ask whether a smaller model can guide a larger model more efficiently than the larger model reasoning alone.

In assistant-guided soft-thought reasoning, a small model sends continuous messages to a larger one. This setting changes the central challenge. The question is not only how to align two hidden spaces, but what reasoning information the small model should communicate. A latent message trained only through downstream likelihood or geometric alignment is under-specified: these signals can make the message compatible with the receiver, but they do not explicitly state which reasoning structure the message should carry.

We therefore frame this task as a **latent communication problem** between a frozen sender language model and a frozen receiver language model. In this framing, the sender's latent messages are not merely implicit reasoning steps or soft prompts; they are **compact messages** sent through a trainable bridge. This reframing reveals two fundamental gaps that existing methods do not fully address:

- **Channel gap.** Messages from the sender may not be readable by the receiver because their representation spaces are heterogeneous. Direct projection can accumulate mismatch and drift, degrading the information that reaches the receiver.
- **Content gap.** Even if messages are geometrically aligned and readable, their semantic content remains weakly specified. The sender is not told what reasoning structure the mes-

sages should carry, and training signals from downstream likelihood alone do not enforce coverage of the reasoning trajectory.

Our approach. We propose **LARC**, short for *Latent Communication with Rationale Anchoring*, a framework that addresses both gaps simultaneously. LARC keeps the sender and receiver frozen and trains only a compact latent bridge. The framework separates three communication roles:

1. **Channel calibration** ensures that sender messages are readable by the receiver. We use bidirectional alignment and cycle consistency over a residual bridge to calibrate the communication channel across heterogeneous model spaces.
2. **Rationale anchoring** ensures that messages carry useful reasoning structure. During training, gold rationale steps serve as communication contracts: the set of latent messages is softly aligned to the set of rationale-step representations, encouraging coverage of the reasoning trajectory through set-wise alignment.
3. **Receiver-side reasoning** verifies that the calibrated and anchored messages are useful for producing the target rationale and answer, through standard language modeling supervision on the receiver.

At inference, rationale anchors are removed. The sender produces compact latent messages from the question and placeholder slots. The bridge maps them into receiver space, and the receiver generates the answer. The assistant never emits a textual chain of thought. The full framework is illustrated in Figure 1.

Our contributions are:

- We formulate small-to-large soft-thought reasoning as a latent communication problem between a frozen sender LM and a frozen receiver LM, identifying channel readability and content specification as the two core challenges.
- We propose LARC, which unifies channel calibration with rationale-anchored coverage matching to learn compact receiver-readable messages that carry reasoning structure.
- We organize the training objective around communication roles rather than loss stacking, making the design of each component interpretable.
- Across Qwen and LLaMA model families, LARC improves average accuracy over Soft-CoT and an unanchored latent communication bridge. Controlled experiments show that corrupted rationale anchors degrade performance, and the resulting model communicates through four latent-message slots at inference while remaining faster than explicit assistant-CoT communication.

2 Related Work

Explicit and latent chain-of-thought. Explicit chain-of-thought methods elicit intermediate reasoning traces in natural language (Wei et al., 2022; Kojima et al., 2022), with follow-up work improving sampling, selection, and deliberation over reasoning paths (Fu et al., 2022; Wang et al., 2022; Yao et al., 2023). Latent reasoning methods reduce or replace these text traces with continuous representations, including continuous thought recycling, compressed dense thoughts, soft thinking, chain-of-embedding, and budget-adaptive latent computation (Hao et al., 2024; Cheng and Van Durme, 2024; Shen et al., 2025; Zhang et al., 2026; Wang et al., 2025; Mohtashami et al., 2025). Recent analyses and surveys further study when latent thinking helps, when it collapses, and how implicit reasoning states should be supervised (Wu et al., 2025; Chen et al., 2025). Supervised implicit-CoT work such as SIM-CoT shows that explicit rationale steps can be useful training signals for latent reasoning states (Wei et al., 2025). LARC follows the broader intuition that rationales are valuable supervision, but studies their role in cross-model communication rather than single-model implicit reasoning.

Small-large model collaboration. Small models can assist large models through draft generation, speculative decoding, self-speculative decoding, weak-to-strong training, task decomposition, or soft latent guidance (Kim et al., 2023; Leviathan et al., 2023; Chen et al., 2023, 2024; Song et al., 2023; Xu et al., 2025). This line is motivated by the cost of scaling and serving large language models (Brown et al., 2020; OpenAI, 2023; Hoffmann et al., 2022; Narayanan et al., 2021). SoftCoT is the closest prior assistant-based baseline: a small model produces soft thoughts that are projected into a larger model's input space. LARC addresses a different aspect of the same collaboration setting: the content of the assistant-to-base latent message. Instead of relying only on final-task supervision, LARC uses rationale steps as training-time con-

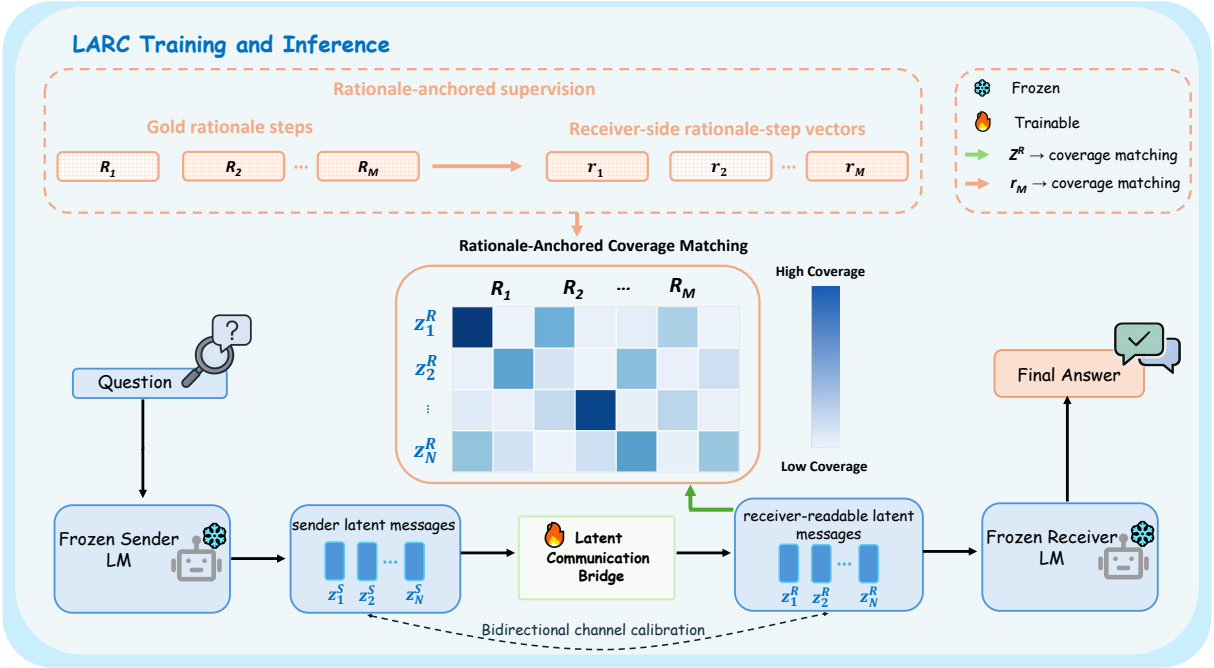


Figure 1: LARC framework. A frozen sender produces sender-side latent messages Z^S, which are mapped by a trainable Latent Communication Bridge into receiver-readable messages Z^R. Channel calibration improves readability across model spaces, while rationale-anchored supervision aligns Z^R with receiver-side rationale-step vectors r_m. At inference, rationale anchors are removed and only compact latent messages are transmitted.

tracts for what the message should communicate.

Latent interfaces and prompt bridges. Continuous prompts, prefixes, and low-rank adapters provide compact, trainable interfaces for steering frozen language models (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2022; Hu et al., 2022). Cross-model latent communication additionally requires a bridge between heterogeneous representation spaces. This relates to feature-level knowledge transfer, mutual learning, bidirectional projection, and cycle-consistent alignment (Hinton et al., 2006; Jiao et al., 2020; Sun et al., 2019; Zhang et al., 2018; Wang et al., 2020; Zhu et al., 2017; Li et al., 2023). LARC uses channel calibration to keep the bridge readable and stable, but treats this as infrastructure rather than the central contribution. The central contribution is rationale anchoring: specifying the reasoning content of compact latent messages.

3 Methodology

LARC operates between a frozen small sender language model $\mathcal{S}$ and a frozen large receiver language model $\mathcal{R}$. Given a question Q, the sender receives an instruction, the question, and N latent-message placeholders. Let

$$Z^S = [z_1^S, \ldots, z_N^S] \in \mathbb{R}^{N \times d_S} \quad (1)$$

denote the sender hidden states at the placeholder positions. A trainable bridge $\Phi_{S \to R}$ maps these messages into the receiver embedding space:

$$Z^R = \Phi_{S \to R}(Z^S) \in \mathbb{R}^{N \times d_R}. \quad (2)$$

The receiver consumes Z^R as compact latent messages and is trained to produce the gold rationale and answer. Both language models remain frozen; only the bridge parameters are updated.

The full training objective organizes around three communication roles, each answering a distinct question about the latent messages:

Component	Question	Objective
Channel Calibration	Are the messages readable?	$\mathcal{L}_{\text{channel}}$
Rationale Anchoring	What should they carry?	$\mathcal{L}_{\text{anchor}}$
Receiver-Side Reasoning	Are they useful?	$\mathcal{L}_{\text{receiver}}$

3.1 Channel Calibration

The bridge must make sender-side messages readable to the receiver across heterogeneous representation spaces. We therefore include calibration losses over the latent message span.

Bidirectional Alignment. First, a bidirectional alignment term maps sender messages into receiver space and receiver-side thought states back into sender space:

$$\mathcal{L}_{\text{align}} = \frac{1}{2}\left\|\Phi_{S\to R}(Z^S) - H^R\right\|_2^2 + \frac{1}{2}\left\|\Phi_{R\to S}(H^R) - Z^S\right\|_2^2, \tag{3}$$

where H^R denotes receiver hidden states on the corresponding latent-message span. Each projector is implemented as a two-layer MLP with a residual connection and Layer Normalization. When $d_S \neq d_R$, a linear projection aligns the skip connection.

Cycle Calibration. Second, a cycle calibration term discourages degenerate projections by requiring the sender message to be recoverable after a round trip. Let $\tilde{Z}^S = \Phi_{R\to S}(\Phi_{S\to R}(Z^S))$:

$$\mathcal{L}_{\text{cycle}} = \left\|Z^S - \tilde{Z}^S\right\|_2^2 + \left(1 - \cos\left(Z^S, \tilde{Z}^S\right)\right). \tag{4}$$

The channel objective combines these terms:

$$\mathcal{L}_{\text{channel}} = \mathcal{L}_{\text{align}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cycle}}. \tag{5}$$

Channel calibration makes compact messages readable to the receiver.

3.2 Rationale-Anchored Message Learning

Rationale anchoring is the core content objective. Let a gold rationale be segmented into M ordered steps. For each step m, we mean-pool the receiver hidden states over the step's token span to obtain a rationale-step vector $r_m \in \mathbb{R}^{d_R}$. Let $Z^R = [z_1^R, \ldots, z_N^R]$ denote the projected latent messages.

Rationale anchoring encourages the latent message set to cover the structure of the rationale trajectory through soft alignment to rationale-step vectors. We compute cosine scores between normalized latent messages and normalized rationale-step vectors:

$$a_{i,m} = \frac{\cos(z_i^R, r_m)}{\tau} - \beta\left|\frac{i}{N} - \frac{m}{M}\right|, \tag{6}$$

where τ is a temperature and β is an optional monotonic position prior. The soft alignment from message i to rationale step m is:

$$\alpha_{i,m} = \frac{\exp(a_{i,m})}{\sum_{m'=1}^{M}\exp(a_{i,m'})}. \tag{7}$$

Each latent message receives a soft rationale target:

$$\bar{r}_i = \sum_{m=1}^{M}\alpha_{i,m} r_m. \tag{8}$$

The rationale-anchored coverage objective combines message-to-rationale matching with a coverage regularizer:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{z}_i^R - \hat{\bar{r}}_i\right\|_2^2 + \lambda_{\text{cov}}\left\|\frac{1}{N}\sum_{i=1}^{N}\alpha_{i,:} - \frac{1}{M}\mathbf{1}\right\|_2^2, \tag{9}$$

where hats denote normalized vectors. The coverage term discourages all messages from collapsing onto a single rationale step.

3.3 Receiver-Side Reasoning Objective

The receiver must use latent messages to produce the desired reasoning output. We therefore train with a receiver-side language modeling objective on the gold rationale and final answer:

$$\mathcal{L}_{\text{receiver}} = -\sum_{t\in\mathcal{I}_R\cup\mathcal{I}_A}\log p_{\mathcal{R}}(y_t \mid y_{<t}, Q, Z^R). \tag{10}$$

This objective tests whether the anchored and calibrated messages are useful to the receiver.

3.4 Overall Training Objective and Inference

The full objective is:

$$\mathcal{L} = \mathcal{L}_{\text{receiver}} + \lambda_{\text{cal}}\mathcal{L}_{\text{channel}} + \lambda_{\text{ra}}\mathcal{L}_{\text{anchor}}. \tag{11}$$

This presentation avoids treating the method as a flat stack of losses: each term plays a distinct communication role.

Inference. At inference, the sender produces N latent messages from the question and placeholder slots. The learned bridge maps them into receiver space, and the receiver generates the answer through the learned latent channel.

4 Experimental Evaluation

We address three research questions:

- **RQ1:** Does rationale-anchored latent communication improve over both prior baselines and an unanchored latent bridge?
- **RQ2:** Do rationale anchors behave as meaningful communication contracts rather than generic extra supervision?
- **RQ3:** How compact is inference-time communication compared with explicit textual CoT?

Table 1: LLaMA results. Rows marked † follow Xu et al. (2025): reported numbers are copied where available, and unreported dataset entries are filled with our matched reproductions; all other results are our 3-run averages. Remaining unavailable entries are marked –. LM-Guided CoT is an explicit textual-rationale communication baseline. SoftCoT, Unanchored Bridge, and LARC use the matched reproduced latent-communication setting.

Method	GSM8K	SVAMP	ASDiv	AQuA	MathQA	CSQA	ARC-C	DU	Avg.
LoRA Fine-Tuning†	75.66	69.78	86.67	52.36	54.71	71.20	75.91	52.39	67.33
Coconut†	76.88	–	86.24	52.98	–	73.42	–	–	–
Zero-Shot CoT†	79.88	**71.22**	86.84	54.85	57.09	73.42	78.41	**54.98**	69.59
LM-Guided CoT	80.79	70.67	87.09	56.17	57.79	73.19	78.30	54.29	69.79
SoftCoT	80.69	70.89	86.99	56.43	58.12	**77.64**	**80.83**	53.66	70.66
Unanchored Bridge	81.35	68.44	87.86	69.42	58.43	76.90	73.66	50.41	70.81
LARC	**83.42**	69.56	**88.73**	**71.78**	**60.23**	74.94	78.53	47.97	**71.89**

Table 2: Qwen results. Notation and baseline definitions follow Table 1.

Method	GSM8K	ASDiv	SVAMP	AQuA	MathQA	CSQA	ARC-C	DU	Avg.
LoRA Fine-Tuning†	81.80	86.80	82.89	62.60	68.81	70.90	79.10	62.87	74.47
Coconut†	82.45	87.22	–	63.61	–	73.74	–	–	–
Zero-Shot CoT†	82.46	88.09	84.22	65.09	71.19	72.96	82.31	65.41	76.47
LM-Guided CoT	83.40	88.70	84.78	67.06	72.60	72.40	82.20	65.04	77.02
SoftCoT	83.65	90.46	84.89	76.51	75.31	67.57	84.41	62.60	78.17
Unanchored Bridge	89.31	**91.04**	84.22	83.46	81.34	73.71	84.64	60.98	81.09
LARC	**89.76**	**91.04**	**85.67**	**84.65**	**81.94**	**76.66**	**86.46**	**65.85**	**82.75**

4.1 Experimental Setup

Model Configuration. We evaluate two sender–receiver model families. In the Qwen setting, the sender is Qwen2.5-1.5B-Instruct and the receiver is Qwen2.5-7B-Instruct (Qwen Team, 2025). In the LLaMA setting, the sender is Llama-3.2-1B-Instruct and the receiver is Llama-3.1-8B-Instruct (Grattafiori et al., 2024). Both sender and receiver are frozen; only the bridge is trained.

Benchmark Datasets. The evaluation covers mathematical reasoning (GSM8K (Cobbe et al., 2021), ASDiv (Xu et al., 2025), SVAMP, AQuA (Ling et al., 2017), MathQA), commonsense reasoning (CommonsenseQA (Talmor et al., 2019)), science questions (ARC-C), and symbolic reasoning (Date Understanding (Srivastava et al., 2023)).

Baseline Methods. We compare two groups of methods:

- **Prior and textual baselines**: LoRA Fine-Tuning (Hu et al., 2022), Coconut (Hao et al., 2024), and Zero-Shot CoT are reported from Xu et al. (2025) and marked with † in Tables 1 and 2. LM-Guided CoT (Lee et al., 2024) provides an explicit textual-rationale communication baseline: the assistant first generates a natural-language rationale, and the receiver predicts from the question plus that rationale.
- **Matched reproduced methods**: SoftCoT (Xu et al., 2025), Unanchored Bridge, and LARC. SoftCoT is the prior assistant soft-thought baseline. The Unanchored Bridge uses the same frozen sender–receiver setup and channel calibration as LARC but without rationale anchoring. LARC is the full rationale-anchored latent communication model.

Training Protocol. Unless otherwise stated, the latent communication models use 4 latent-message slots. The bridge is instantiated as residual MLP blocks with bottleneck size 1024, ReLU activation, and LayerNorm. We train with AdamW using learning rate 2×10^{-5} for up to 10 epochs, with batch size 8 on most datasets and batch size 4 on AQuA, and apply early stopping on the validation trajectory. Hyperparameters are tuned by model family: for LLaMA, we use $\lambda_{\text{cal}} = 0.15$, $\lambda_{\text{cyc}} = 0.08$, and $\lambda_{\text{ra}} = 0.10$; for Qwen, we use $\lambda_{\text{cal}} = 0.10$, $\lambda_{\text{cyc}} = 0.05$, and $\lambda_{\text{ra}} = 0.08$. Across both settings, the coverage weight is $\lambda_{\text{cov}} = 0.05$, the rationale-matching temperature is $\tau = 0.1$, and the position prior is $\beta = 0.5$.

Evaluation Protocol. We report task accuracy on each benchmark and average over the reported tasks within each model family. Tables 1 and 2 separate prior reported numbers (†) from the matched

Table 3: Anchor corruption on the seven reported reasoning tasks. The LM target remains unchanged; only the rationale anchor is corrupted.

Variant	GSM8K	ASDiv	SVAMP	AQuA	MathQA	CSQA	ARC-C	Avg.
LARC full	83.41	88.73	67.00	71.76	60.24	74.94	78.52	74.94
LARC + shuffled rationale anchors	82.21	86.73	63.60	65.96	55.74	73.44	77.62	72.19
LARC + random step order	82.91	87.83	65.30	68.66	58.04	74.24	78.22	73.60

reproduced methods that are directly compared under the frozen sender–receiver setting. For the compactness analysis, we compare against an explicit assistant-CoT path by measuring assistant-side payload and end-to-end latency on GSM8K, AQuA, and ARC-C. For anchor corruption, we keep the receiver-side rationale and answer targets fixed and modify only the rationale-anchor target, so the degradation isolates the quality of the communication contract.

4.2 Main Results

We first examine whether rationale-anchored latent communication improves over prior baselines and an unanchored latent bridge. Across both model families, LARC achieves the best average accuracy among the matched reproduced methods (Tables 1 and 2). The gain is larger on Qwen and smaller but still positive on LLaMA, suggesting that rationale anchoring generalizes across model families. This matters because the comparison is made against the Unanchored Bridge, which uses the same frozen sender–receiver setup and channel calibration as LARC but removes rationale anchoring. The resulting gap shows that message-content specification provides additional gains beyond channel calibration alone: rationale contracts help the bridge communicate the reasoning content the receiver needs.

The comparison with LM-Guided CoT highlights a complementary distinction in how rationales are used. In LM-Guided CoT, the assistant communicates by producing a textual rationale at inference time, so the rationale itself becomes the payload passed to the receiver. In LARC, rationales define the communication contract that shapes what the compact latent messages cover. The fact that LARC improves over LM-Guided CoT while communicating through latent slots suggests that explicit rationales are especially effective when used to supervise message content. Together, these comparisons show that rationale anchoring improves performance over both prior baselines and an unanchored latent bridge.

4.3 Rationale Anchors as Communication Contracts

We next examine whether rationale anchors function as communication contracts for latent messages. We first perturb the anchor signal during training and then inspect what the learned channel retains and how the receiver uses it.

Anchor Corruption. The controlled corruption experiment changes only the rationale-anchor target used by $\mathcal{L}_{\text{anchor}}$, while the receiver-side LM target remains the correct rationale and answer. Any degradation therefore measures sensitivity to corrupted training-time anchors.

Table 3 shows that both corruption variants reduce accuracy across all seven reported tasks. Cross-sample anchor shuffling causes the largest drops on AQuA (5.80 points) and MathQA (4.50 points), while random step order is consistently less destructive but still lowers the seven-task average by 1.34 points. This pattern shows that LARC is driven by correct rationale content and set-level coverage of the rationale trajectory, with step order playing a secondary role in the anchor signal.

The relative ordering of the two corruption types is also informative. Random step order preserves the rationale content but perturbs its sequence, whereas cross-sample shuffling replaces the content of the contract itself. Because LARC uses set-level coverage, it is more sensitive to the correctness of rationale content than to mild order perturbations. The larger degradation from shuffled anchors is consistent with this design: what matters most is that the latent messages are anchored to the right reasoning content for the current question.

Mechanistic Evidence. To complement the corruption results, we examine the learned communication channel and the content retained by the latent messages.

The mechanism story has three pieces. Figure 2 first isolates communication space quality after channel calibration: sender messages become readable in the receiver space, but readability alone does not specify what they should communicate.

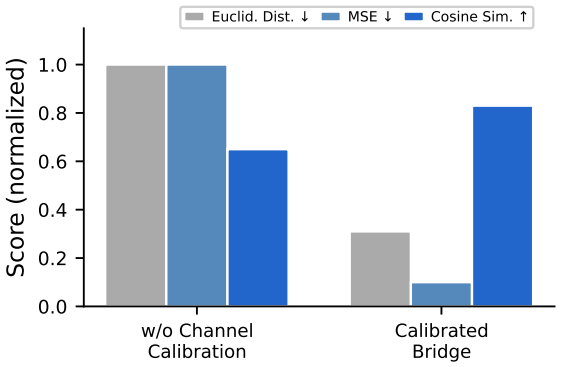


Figure 2: Channel calibration improves communication space quality. After bidirectional alignment and cycle calibration, sender messages align much more closely with receiver-side latent states in Euclidean distance, MSE, and cosine similarity.

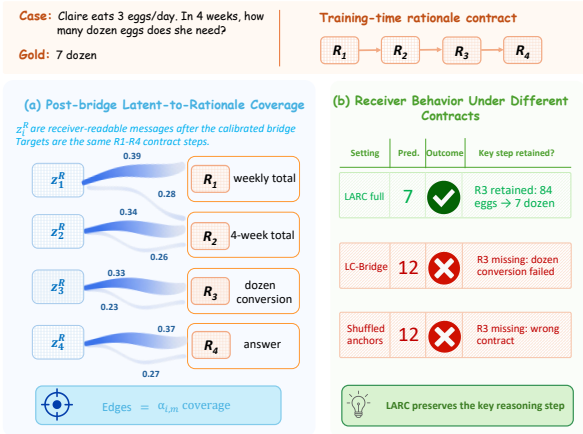


Figure 3: Rationale-anchored communication in action on a GSM8K case study. **(a)** The case-specific training-time rationale contract is shown together with the post-bridge latent-to-rationale coverage view, where later latent messages concentrate on the conversion and answer steps. **(b)** Receiver behavior under different contracts shows that the full LARC model predicts the correct answer 7, while the Unanchored Bridge baseline (labeled "LC-Bridge" inside the figure) and the shuffled-anchor variant both stop at 12, indicating that the final unit-conversion step is not reliably preserved without the correct rationale contract.

Figure 3 then gives a concrete end-to-end example of how a rationale contract changes receiver behavior on one problem. Figure 4 finally aggregates the broader rationale-anchoring evidence through coverage structure and anchor corruption.

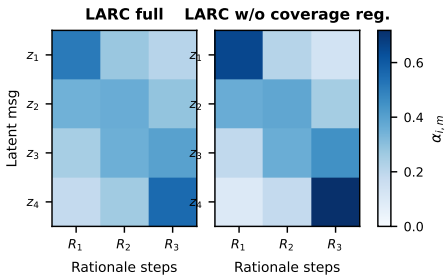
Channel Readability. Figure 2 quantifies representation mismatch between projected sender messages and receiver hidden states on the aligned message span. Channel calibration—bidirectional alignment and cycle consistency—reduces Euclidean distance by 69%, MSE by 90%, and improves cosine similarity by 19% compared to uncalibrated projection. The calibrated bridge achieves substantially better alignment across all three metrics, establishing the readability gains provided by channel calibration.

Figure 3 turns the mechanism story into a single end-to-end example. The gold rationale supplies a compact training-time contract; the learned latent slots distribute coverage over intermediate arithmetic steps rather than collapsing onto the first step; and the receiver-side behavior changes in a way that matches that contract. In this case, the full model preserves the final unit-conversion step from 84 eggs to 7 dozen, whereas both the unanchored bridge and the shuffled-anchor variant stop at 12. This makes the mechanism claim visually concrete: rationale anchoring provides a content-specifying signal that helps the latent channel retain the right intermediate structure.

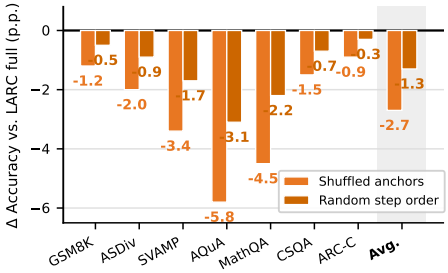
Rationale-Anchored Coverage. Figure 4a visualizes the soft alignment matrix $\alpha_{i,m}$ between latent message slots $(z_1, \ldots, z_4)$ and rationale steps $(R_1, \ldots, R_M)$. In the full LARC model, alignment mass distributes across multiple rationale steps, indicating that the anchor objective encourages set-level coverage over rationale steps. The figure makes the mechanism visible at the level of set-wise alignment between latent messages and rationale steps. Without coverage regularization, the alignment collapses onto fewer steps—a pattern consistent with the 2.07-point average drop in downstream accuracy (Table 5).

Anchor Corruption Effect. Figure 4b reports the controlled anchor corruption experiment as negative delta accuracy relative to the full LARC model. When rationale anchors are shuffled across samples or their step order is randomized (while keeping receiver-side LM targets unchanged), performance degrades across all seven reported tasks. Cross-sample shuffling is consistently more destructive than within-rationale step randomization, which is consistent with LARC's set-level coverage design: replacing the rationale content is more damaging than perturbing the order of steps. This pattern confirms that the correctness of the rationale contract matters.

Together, the corruption test, case study, and coverage analysis show that rationale anchors shape what the latent messages retain and how the receiver uses them. These results establish rationale



(a) Coverage distributes over rationale steps; removing coverage regularization concentrates mass.



(b) Anchor corruption lowers accuracy, with shuffled anchors causing larger drops.

Figure 4: Rationale anchoring shapes message content: coverage prevents concentrated alignment, and corrupting anchors lowers downstream accuracy.

Table 4: Assistant-side compactness. LARC communicates four latent-message slots at inference rather than a full assistant-generated textual rationale.

Dataset	Explicit rationale tokens	Median	LARC slots	Token saving
GSM8K	136.71	127	4	97.07%
AQuA	82.65	69	4	95.16%
ARC-C	28.01	28	4	85.72%

anchors as communication contracts for latent communication.

4.4 Compactness Against Explicit Assistant-CoT Communication

We finally examine how compact inference-time communication can be relative to explicit assistant-CoT communication. The comparison uses a setting where the small model first generates a textual rationale and the receiver then consumes that text. SoftCoT, the Unanchored Bridge, and LARC all operate through compact latent messages at inference.

Table 4 shows that the assistant-side communication can be reduced from tens or hundreds of text tokens to four latent-message slots. In the latency evaluation, the latent-message path runs at 1.80, 1.90, and 1.91 seconds per example on GSM8K, AQuA, and ARC-C, respectively. The explicit assistant-CoT path takes 2.76, 2.87, and 2.89 seconds per example, yielding roughly 1.5× speedup. These measurements close the compactness question: LARC preserves strong task performance while communicating through four latent-message slots at inference.

Table 5: Ablations for the rationale-anchor objective. These numbers come from a separate ablation run and are not from the same runs as the main results tables.

Variant	GSM8K	AQuA	ARC-C	Avg.
LARC full	80.91	67.06	76.47	74.81
LARC w/o coverage regularization	82.27	60.00	75.96	72.74
LARC w/o position bias	80.91	71.76	77.75	76.81

4.5 Ablation Studies

Finally, we ablate components of the rationale-anchor objective.

Table 5 shows that removing coverage regularization lowers the average from 74.81 to 72.74, with the largest drop on AQuA. This suggests that coverage helps prevent the latent messages from collapsing onto an incomplete rationale signal and provides the more consistent contribution across tasks. By contrast, removing the position bias improves AQuA and ARC-C in this diagnostic run, indicating that strict monotonic alignment to rationale-step order is not uniformly necessary. Correct reasoning content is therefore the stronger signal. Local step order appears to play a secondary role, with the main benefit coming from covering the right reasoning trajectory.

5 Conclusion

We presented LARC, a rationale-anchored latent communication framework for efficient small-large language model reasoning. Across Qwen and LLaMA model families, LARC improves over both SoftCoT and an unanchored latent bridge, showing that rationale anchoring strengthens assistant-to-receiver communication. Corruption and mechanism analyses further show that rationale anchors specify what the latent channel retains and how the receiver uses it. At inference, LARC communicates through four latent-message slots, reducing latency relative to explicit assistant-CoT communication while preserving strong task performance. More broadly, the results suggest that latent reasoning interfaces need explicit supervision over message content, not just cross-model readability. This makes rationale anchoring a practical design principle for future small-large collaboration.

Limitations

Our method relies on supervision with explicit reasoning traces and fixed prompting templates; its effectiveness on tasks without CoT annotations or when rationales must be generated automatically remains unclear. Moreover, the latent messages operate in a continuous space and are not directly interpretable; despite improved alignment metrics, they do not guarantee faithful or safe reasoning, and error patterns can be difficult to diagnose.

Ethics Considerations

This work adheres to ethical principles in AI research. We exclusively utilize publicly available and widely accepted benchmark datasets that do not contain personally identifiable information or sensitive content. We acknowledge that the underlying large language models used in this research may inherit biases from their training data, and our methodology does not specifically address or mitigate these inherent biases. We advocate for responsible deployment practices, including human oversight and careful validation of AI-generated content, when applying our framework or similar technologies in real-world scenarios. Furthermore, we are committed to transparency and reproducibility by detailing our methods and intend to make our code publicly available to facilitate further research and scrutiny. While we acknowledge the environmental impact of training large AI models, we prioritize computational efficiency where feasible and encourage continued efforts to reduce the carbon footprint of AI research.

Reproducibility Statement

To ensure full reproducibility of our findings, all source code for the LARC framework, trained bridge checkpoints, and configuration files used for all experiments will be made publicly available upon acceptance. Our experiments are conducted on publicly accessible benchmarks, including GSM8K, ASDiv, AQuA, CommonsenseQA, ARC-C, and Date Understanding, and build upon standard open-source models such as LLaMA-3.1-8B-Instruct, LLaMA-3.2-1B-Instruct, Qwen2.5-7B-Instruct, and Qwen2.5-1.5B-Instruct, all available via the Hugging Face Hub. A comprehensive list of hyperparameters, compute requirements, and dataset pre-processing steps are detailed in Appendix. All released artifacts will comply with their original open-source licenses, and data sources will be clearly acknowledged.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes

Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174.

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W Mahoney, Amir Gholami, and Kurt Keutzer. 2023. Speculative decoding with big little decoder. *Advances in Neural Information Processing Systems*, 36:39236–39256.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jooyoung Lee, Fan Yang, Thanh Tran, Qian Hu, Emre Barut, and Kai-Wei Chang. 2024. Can small language models help large language models reason better?: Lm-guided chain-of-thought. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2835–2843.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Jingzhi Li, Zidong Guo, Hui Li, Seungju Han, Ji-Won Baek, Min Yang, Ran Yang, and Sungjoo Suh. 2023. Rethinking feature-based knowledge distillation for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20156–20165.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 158–167.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Amirkeivan Mohtashami, Matteo Pagliardini, and Martin Jaggi. 2025. Cotformer: A chain of thought driven architecture with budget-adaptive computation cost at inference. In *International Conference on Learning Representations*, volume 2025, pages 11503–11520.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, and 1 others. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–15.

OpenAI. 2023. Gpt-4 technical report. https://arxiv.org/abs/2303.08774. ArXiv:2303.08774.

Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4323–4332.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Chien-Yi Wang, Ya-Liang Chang, Shang-Ta Yang, Dong Chen, and Shang-Hong Lai. 2020. Unified representation learning for cross model compatibility. *arXiv preprint arXiv:2008.04821*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yiming Wang, Pei Zhang, Baosong Yang, Derek Wong, and Rui Wang. 2025. Latent space chain-of-embedding enables output-free llm self-evaluation. In *International Conference on Learning Representations*, volume 2025, pages 70938–70970.

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.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. 2025. Sim-cot: Supervised implicit chain-of-thought. *arXiv preprint arXiv:2509.20317*.

Junhong Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. 2025. Llms are single-threaded reasoners: Demystifying the working mechanism of soft thinking. *arXiv preprint arXiv:2508.03440*.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23336–23351.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. 2026. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *Advances in Neural Information Processing Systems*, 38:168990–169012.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

A Appendix

A.1 Training Algorithm

Algorithm 1 summarizes the full LARC training procedure, which combines channel calibration, rationale anchoring, and receiver-side reasoning in a single pass over each training batch.

Algorithm 1: LARC training

Input: $\mathcal{S}, \mathcal{R}, \Phi_{S\to R}, \Phi_{R\to S}, \mathcal{B}$
Output: $\Phi_{S\to R}, \Phi_{R\to S}$
foreach $(Q, \{R_m\}_{m=1}^{M}, A) \in \mathcal{B}$ **do**
  $(x_S, x_R) \leftarrow \text{construct}(Q, N)$
  $H_S \leftarrow \mathcal{S}(x_S),\ Z_S \leftarrow H_S[s_a : e_a]$
  $Z_R \leftarrow \Phi_{S\to R}(Z_S)$
  $\tilde{x}_R \leftarrow \text{inject}(x_R, Z_R)$
  $\text{out} \leftarrow \mathcal{R}(\tilde{x}_R, \text{gold} = \{R_m\}_{m=1}^{M}, A)$
  $L_{\text{lm}} \leftarrow \text{NLL}(\text{out.logits}, \{R_m\}_{m=1}^{M}, A)$
  $H_B \leftarrow \text{out.hidden_states}[-1]$
  $r_m \leftarrow \text{meanpool}(H_B[\text{span}(R_m)])$
  $L_{\text{align}} \leftarrow \|\Phi_{S\to R}(Z_S) - H_B[s_b : e_b]\|_F^2$
  $Z'_S \leftarrow \Phi_{R\to S}(Z_R)$
  $L_{\text{cycle}} \leftarrow \|Z'_S - Z_S\|_F^2 + (1 - \cos(Z'_S, Z_S))$
  $\alpha_{i,:} \leftarrow \text{softmax}\Big(\frac{\cos(z_i^R, r_m)}{\tau} - \beta|i/N - m/M|\Big)$
  $\bar{r}_i \leftarrow \sum_m \alpha_{i,m} r_m$
  $L_{\text{anchor}} \leftarrow \frac{1}{N}\sum_i \|z_i^R - \bar{r}_i\|_2^2 + \lambda_{\text{cov}}\mathcal{C}(\alpha)$
  $L \leftarrow L_{\text{lm}} + \lambda_{\text{align}} L_{\text{align}}$
    $L \leftarrow L + \lambda_{\text{cycle}} L_{\text{cycle}} + \lambda_{\text{anchor}} L_{\text{anchor}}$
  $L \rightarrow$ backprop on $\Phi_{S\to R}, \Phi_{R\to S}$
end

A.2 Implementation and Training Details

Section 4.1 summarizes the standard training protocol used in the main experiments. Here we record the exact family-specific weights and hardware configuration used across the final runs.

For LLaMA models, we use channel loss weight $\lambda_{\text{cal}} = 0.15$ with cycle weight $\lambda_{\text{cyc}} = 0.08$ and rationale-anchor weight $\lambda_{\text{ra}} = 0.10$. For Qwen models, we use $\lambda_{\text{cal}} = 0.10$, $\lambda_{\text{cyc}} = 0.05$, and $\lambda_{\text{ra}} = 0.08$. The coverage regularizer weight is $\lambda_{\text{cov}} = 0.05$ across all configurations. The temperature in rationale-step matching is $\tau = 0.1$ and the position prior is $\beta = 0.5$. The default batch size is 8, with batch size 4 for AQuA, and training runs for up to 10 epochs with early stopping.

The main training experiments are performed on an NVIDIA H800 GPU, while some supplementary appendix experiments are run on a separate server. Inference is conducted on an NVIDIA RTX 5090 GPU with mixed precision (bfloat16). During inference, the sender produces 4 latent messages which are projected through the bridge to the receiver.

A.3 Additional Cross-Dataset Results

To test whether rationale-anchored latent communication transfers across nearby mathematical reasoning datasets, we train on ASDiv and evaluate on AQuA, and vice versa. Table 6 shows that the transferred models remain close to their in-domain counterparts while substantially outperforming zero-shot baselines, especially on the more difficult AQuA target.

Table 6: Cross-dataset transfer for LARC. Zero-shot and in-domain references come from the corresponding family-specific main results.

Family	Training	Testing	Accuracy	vs Zero-shot	vs In-domain
LLaMA	ASDiv	AQuA	70.92	+16.07	-0.84
LLaMA	AQuA	ASDiv	88.21	+1.37	-0.52
Qwen	ASDiv	AQuA	82.88	+17.79	-0.65
Qwen	AQuA	ASDiv	90.34	+2.25	-0.70

A.4 Additional LLaMA Ablation Results

Table 7 complements the rationale-anchor ablation in the main text with a LLaMA-family diagnostic run. The same pattern holds: removing coverage regularization causes the larger drop, while removing the position bias has only a minor effect.

Table 7: Supplementary LLaMA ablation for the rationale-anchor objective. These numbers come from a separate diagnostic run, analogous to the main-text ablation table.

Variant	GSM8K	AQuA	ARC-C	Avg.
LARC full	82.60	70.90	78.10	77.20
LARC w/o coverage regularization	81.80	67.90	77.30	75.67
LARC w/o position bias	82.40	71.60	78.00	77.33

A.5 Prompt Snippets

We include the three prompt artifacts that are most central to LARC: the sender-side instruction that shapes compact latent messages, the receiver-side prompt that consumes those messages, and the serialized rationale format that defines the training-time contract. We present them as colored full-width boxes for readability rather than dense inline prose.

Artifact 1: Core sender instruction template used to generate compact latent messages for quantitative reasoning tasks.

```
Sender-side user prompt:
You are given N latent-message
    placeholders that will be used to
produce hidden-state messages for
    another language model solving
    the
```

```
following math reasoning task efficiently and clearly.

Requirements:
- include useful numbers, operations, or key concepts
- start from the most important clues
- do not rely on long free-form explanations or serial-number style cues
- place informative problem cues near the latent-message placeholders

Problem:
{question}

Sender-side placeholder region:
[LATENT_MESSAGE_START]
[SOFT_SLOT_1] [SOFT_SLOT_2] ... [SOFT_SLOT_N]
[LATENT_MESSAGE_END]
```

Artifact 2: Core receiver prompt template that injects sender-produced latent slots as optional hints.

```
Receiver-side user prompt:
Solve the problem efficiently and clearly.

For simple problems:
- provide a concise solution with minimal calculation

For complex problems:
- use the format
  ## Step 1: ...
  ## Step 2: ...
  ...

Always conclude with:
Therefore, the final answer is: \boxed{answer}.

Optional latent-message block:
There are compact latent-message placeholders produced from a weaker assistant model.
These hidden-state messages may provide useful cues.
If they are helpful, use them as reference.
If they are not helpful, ignore them and focus on the problem.

Receiver-side placeholder region:
[LATENT_MESSAGE_START] [SOFT_SLOT_1] ... [SOFT_SLOT_N] [LATENT_MESSAGE_END]
```

Artifact 3: Receiver-side supervised rationale format used to derive ordered rationale-step spans for anchoring.

```
Training-time receiver target:
## Step 1: {rationale step 1}
## Step 2: {rationale step 2}
...
## Step M: {rationale step M}
Therefore, the final answer is: \boxed{answer}.

Rationale-contract role:
- each step is mapped to an ordered token span on the receiver side
- projected latent slots are softly aligned to these step vectors
- the final answer line remains part of the receiver-side LM target
```

A.6 LLM Usage Statement

During manuscript preparation, we used a large language model for limited language polishing and copyediting. All technical content, experimental design, results, and final wording decisions were verified and approved by the authors.