NEURAL ESPERANTO: LEARNING TO COMMUNICATE ACROSS FROZEN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can different language models exchange symbols, add information, and pass the result to an unfamiliar receiver? We introduce *Neural Esperanto*, a shared communication interface for frozen language models. Sending and receiving adapters generate discrete symbols one at a time, with each symbol read back before the next is produced. Training uses a reference code derived from English. Qwen sends facts to Mistral, which adds its own facts and sends a message to Phi. Phi learns from Qwen/Llama messages only and answers questions revealed after both messages are complete. On 1,152 new spatial worlds with unseen entity–distance assignments, two runs achieve 97.05% and 97.14% accuracy, compared with 97.40% for a similarly trained text-based chain. Both runs stay within the predefined two-percentage-point margin of text. Even with access to all test labels, a predictor that only counts symbols reaches only 34.95% and 33.33%; replacing messages makes answers follow the replacement facts. Controlled comparisons show that training the receiving adapter improves both reading incoming messages and generating new ones. These results show that frozen models can communicate using a learned interface derived from English, while developing independent meanings and grammar remains an open challenge.

1 INTRODUCTION

One model knows that Eli is south of Ben; another knows that Ben is west of Cleo. Can they pass this information to a third model that will later be asked how far Eli is from Cleo? The first two cannot simply send the answer because they have not seen the question. They must preserve facts that another model can extend and use.

Models can already communicate through continuous vectors or translated attention memories (Du et al., 2026; Liu et al., 2026; Chen et al., 2026). We study discrete messages: sequences of integer symbols shared by different models. Each model has a receiving adapter, called its *Ear*, and a sending adapter, called its *Mouth*. Senders share an output layer, the *Writer*. The underlying Transformer and language-model output head remain frozen throughout. After picking a symbol, the sender reads it back through its own Ear and Transformer before choosing the next. The Ear is therefore used twice: to read another model's message and to read the symbols the model has already generated.

This dual role motivates our training comparison: updating Mouth alone versus updating both Mouth and Ear. We also separate what the Ear contributes to reading incoming messages from what it contributes to reading back generated symbols. Training gradients pass back through the frozen Transformer, while the final model produces answers through its original text output head.

Our main experiment follows Qwen $\rightarrow$ Mistral $\rightarrow$ Phi. Qwen and Mistral finish their messages before any question is revealed. Phi joins later: only its Ear is trained, using messages from Qwen and Llama without question-answer labels. At test time, Phi reads messages from Mistral for the first time. To test whether messages preserve who is related to whom, we swap which entity gets which distance while holding overall symbol counts constant. We compare the result with a predictor that can see symbol counts but not order, and with the same chain of models exchanging text. Replacing each source of information tests whether answers follow its content.

- We introduce a shared symbol interface that lets frozen models pass facts through a two-message chain and communicate with a newly trained receiver.

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

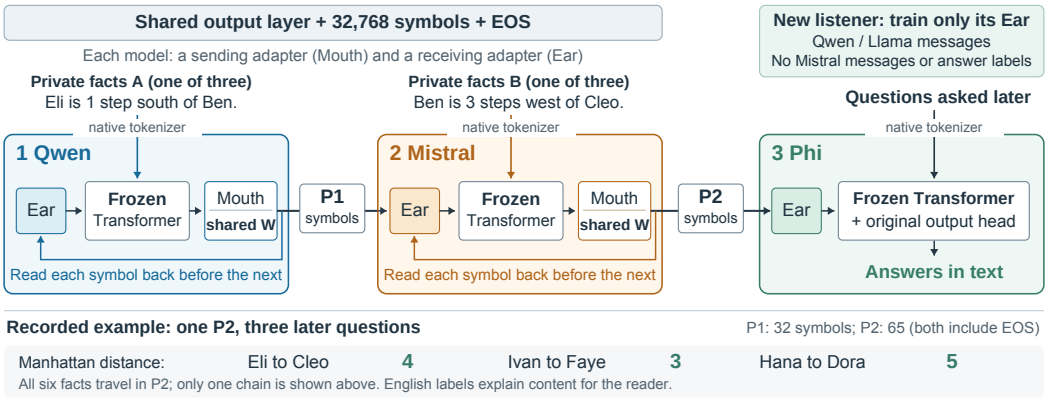


Figure 1: How three models communicate in sequence. Qwen sends message P_1; Mistral reads P_1, adds its private facts, and sends P_2. Phi receives the test questions only after both messages have finished. Each sender reads back its own generated symbols through its Ear and frozen Transformer before emitting the next symbol. Facts and questions enter through each model's native tokenizer, while machine symbols enter through the Ear adapter. English labels explain one of three spatial chains in a recorded example. Complete messages and all three recorded answers appear in Appendix G.

- Tests on new relation combinations and question types show that the messages remain useful beyond training examples. On balanced spatial worlds, accuracy approaches the text baseline and exceeds the best order-blind predictor.
- Controlled training comparisons show why the receiving adapter matters both when a model reads a message and when it generates one.

Neural Esperanto here names a shared learned interface. Its training code comes from English, so the experiments study communication across models within an existing linguistic structure. Inventing a new grammar is a separate goal.

2 How models send and receive messages

2.1 Send the facts first, ask questions later

A world contains two separate fact sets, x_A and x_B. The first model sends message P_1. The second reads it, adds x_B, and sends message P_2. Only then does the final model receive questions $q_1, \ldots, q_Q$:

$$P_1 = \mathrm{Pub}_A(x_A), \qquad P_2 = \mathrm{Pub}_B(P_1, x_B), \qquad \hat{y}_j = \mathrm{Read}_C(P_2, q_j). \tag{1}$$

Each message is a sequence of discrete integer symbols ending with an end-of-message symbol, EOS. Senders receive no questions, answers, or world or test-group identifiers. The exact same message P_2 is used to answer every question about that world. Only the generated symbol IDs travel between models: no hidden vectors, original text, or target sequences are shared.

For model i, Ear E_i converts symbols into the input vectors its Transformer normally receives. The frozen Transformer F_i processes those vectors and builds its own attention memory. Mouth M_i uses attention over multiple context states H_i, together with the current state h_t, so it can select information from across the document. Its 256-dimensional output enters a shared 512-dimensional code space:

$$v_t = \mathrm{LN}(M_i(h_t, H_i)), \quad \ell_t(z) = s\langle \mathrm{normalize}(W v_t), c_z \rangle, \quad \ell_t(\mathrm{EOS}) = w_e^\top v_t + b_e. \tag{2}$$

The embedding vector c_z for each symbol is fixed. All senders share the same Writer projection, normalization, score scale, and EOS predictor. Each model needs only one Ear and one Mouth, so adding a new partner does not require building a separate translator. During evaluation, the model

Table 1: Training stages in each run. One update corresponds to one optimizer step; examples may be used repeatedly. Original Transformer and language-model output heads remain frozen throughout.

Stage	What is trained and how	Training steps
Starting code	English-derived symbol and embedding mapping	Shared initialization
Learn to receive	Three Ears; answer loss and correct-message margin	400 updates per model
Send initial facts	Mouths and Writer; next-symbol prediction	1,000 updates; 1,024 worlds
Send combined facts	Mouths and Writer; combined-fact target messages	1,500 updates; same worlds
Refine Llama	Llama Mouth; other senders and Writer fixed	6,000 updates
Add Phi receiver	Phi Ear; input-embedding and length targets	512 messages; 4,000 updates
Compare Ear training	Mistral Mouth; Ear either updated or fixed	600 updates
Final evaluation	All parameters fixed	Final 600-update checkpoints

picks the highest-scoring symbol and feeds it through E_i and F_i before making the next choice. The final receiver answers using its original Transformer and vocabulary head, without any external classifier or graph solver.

The base models are Qwen2.5-1.5B-Instruct, Llama-3.2-1B-Instruct, and Mistral-7B-Instruct-v0.3. Their Ear adapters contain 4.72M, 6.29M, and 12.59M trainable parameters. The new Phi-4-mini-instruct Ear uses neighboring symbols and contains 24.75M parameters. Because these adapters can perform substantial computation of their own, we cannot attribute every reasoning step solely to the frozen Transformer. Implementation details appear in Appendix A.

2.2 Learning to send, receive, and read back symbols

Training begins with a reference code: an English byte-level BPE tokenizer whose 32,768 vocabulary entries are mapped one-to-one to shared symbols, along with an EOS symbol. Encoding the source facts produces target messages in their original order. The original Ears are trained with answer cross-entropy and a margin loss that favors the correct message over an unrelated one. Mouth adapters and the shared Writer learn to predict the next symbol from earlier target symbols. This supervised setup lets us test whether models can use a shared interface before studying how a new language might emerge.

Our main training comparison tests whether Mistral's Ear also benefits from training when the model learns to send combined facts. Both conditions update Mistral's Mouth for 600 steps on the same 1,024 worlds and generated first messages. One condition also updates its Ear through the frozen Transformer, while the other keeps the Ear fixed. Both minimize the same weighted next-symbol cross-entropy:

$$\mathcal{L}_Z = -\frac{\sum_{n,t} w_{z^*_{nt}} \log p(z^*_{nt} \mid P_{1,n}, x_{B,n}, z^*_{n,<t})}{\sum_{n,t} w_{z^*_{nt}}}. \tag{3}$$

Weights are computed from the training targets. When the Ear is trainable, it receives gradients both from reading P_1 and from reading earlier target symbols. The original model weights, shared Writer, and all other adapters stay fixed. Training uses discrete target IDs directly. At test time, the sender receives no target prefix and must generate the whole message itself.

We repeat task-specific training twice from the same reference-code initialization. Run 1 uses seed 37101 and Run 2 uses seed 37102. We evaluate the final checkpoint after all 600 updates in each run, with no selection based on test scores. The runs share the initial code but independently learn their task-specific adapters.

From the reference code to a trained sender. Table 1 outlines the full training sequence. The first stage teaches the original Ears to convert messages into useful model inputs. Mouth and Writer then learn to send initial facts, followed by combined facts. Additional Llama training supplies

messages for Phi's Ear. We then compare updating Mistral's Mouth alone with updating Mouth and Ear together. This 600-step comparison tests how the Ear helps an already trained model learn to send; it does not train the whole system from scratch.

2.3 Teaching a new model to receive messages

Phi trains only its Ear adapter on 512 messages generated by Qwen and Llama over 4,000 updates, learning to predict embedding sequences and lengths from reference targets. No question-answer labels, Mistral messages, or test worlds are seen during this training. Other adapters and base model weights stay fixed. Phi must then interpret Mistral's messages, testing whether it can understand a sender never encountered during training. We do not test Phi's sending ability.

One shared symbol, different model inputs. A single shared symbol can map to different numbers of input tokens across models. Phi's Ear therefore predicts both an embedding sequence and the number of token positions to occupy. A convolutional network considers neighboring symbols so each symbol's interpretation can reflect local context. Training compares these predictions with Phi's input embeddings for the tokenized reference text. At test time, the received symbols alone feed into the Ear. Phi's Transformer then builds its own attention memory and generates text answers through its original vocabulary head. In this way, models with distinct tokenizers exchange messages without matching hidden dimensions or copying another model's attention memory.

A complete exchange. Figure 1 follows the first test world in Run 1, selected by input order. Qwen reads three facts, including *Eli is 1 grid steps south of Ben*, and sends a 32-symbol P_1 beginning `Z14751 Z4 Z47` and ending in EOS. Mistral reads these symbols through its Ear and separately receives three new facts, including *Ben is 3 grid steps west of Cleo*. Its Mouth and the shared Writer produce a 65-symbol P_2 that preserves all six facts. Both lengths include EOS. Only then does Phi receive the question *What is the Manhattan distance in grid steps from Eli to Cleo?* Its answer ends with `<answer>4</answer>`. The same unchanged message also yields 3 for Ivan–Faye and 5 for Hana–Dora. Mistral passes on the combined facts, and Phi performs the final distance calculation. We decode the messages into English separately to explain the example; the learned exchange itself passes symbols directly between the adapters. Appendix G gives both full messages.

3 Experimental design

Testing who is related to whom. The main test contains 1,152 worlds in 16 groups of 72. Each world has three two-step spatial chains over nine familiar entities, with one step supplied by Qwen and the other by Mistral. Training chains lie along a single axis; test chains turn a corner. Each source uses the distances $1, 2, 3$ once. Permuting their assignments and rotating directions changes the relationships while preserving symbol counts in the reference text. The same question therefore has different answers in matched worlds. Within each group, a predictor given only the question, only either source, or only the original text's word counts can be correct at most $1/3$ of the time.

We exclude earlier development worlds and their equivalents under chain reordering, joint rotation, and cyclic reassignment of distances. This exclusion covers 1,619 groups of equivalent worlds from 22 earlier sources. The generator, checkpoints, and decision rules were fixed before producing test answers. This final test was planned in advance; earlier development was exploratory. Each second message supports three later questions, giving 3,456 answers per condition and run.

How well could a receiver do without symbol order? Because a generated message can deviate from its training target, we compute an order-blind accuracy limit using the messages actually produced. Let $b(P_2)$ count the occurrences of each symbol in P_2, ignoring order, and group examples by $g = (\text{test group}, q, b(P_2))$. Any predictor that ignores symbol order must output the same answer for all examples sharing the same g. If we give this predictor access to all test labels, its best strategy is to pick the majority label in each group:

$$U_{\text{bag}} = \frac{1}{N} \sum_{g} \max_{y} |\{n : g_n = g,\ y_n = y\}| \,. \tag{4}$$

Table 2: Accuracy on new relations and tasks (%). Each row reports 3,456 answers per run. The last column checks whether both runs meet the predefined margin of no more than 2 percentage points below text, allowing for uncertainty. This column compares with text; it does not measure overall task success. Planning reuses test worlds.

Test	Run 1	Run 2	Text	Meets text margin?
New direction combinations	81.94	82.47	83.02	No
New distance combinations	96.79	97.42	97.69	No
Reversing a relation	81.37	81.94	82.84	No
New entity–distance assignments	97.05	97.14	97.40	Yes
Action planning (same worlds)	45.66	45.52	45.69	Yes

This is the highest accuracy any fixed count-only rule can achieve on this test set. This benchmark is deliberately generous because an actual receiver has neither group labels nor true answers. Exceeding this limit shows that message structure carries useful information beyond symbol counts. It does not identify a grammar different from the English-derived reference code.

Replacing information and comparing with text. To test whether models actually rely on the communicated facts, we separately replace the first message P_1, Mistral's private facts x_B, or the final message P_2 with data from another controlled world. When P_1 or x_B is swapped, Mistral generates a fresh P_2. These tests replace content rather than shuffling symbols within a message. We then evaluate whether answers match the original facts or follow the replacement facts.

Our text baseline uses the same models, visible facts, sending stages, delayed questions, generation limits, and 600 extra updates. Its Qwen/Mistral input adapters match the corresponding Ears in size. Overall parameter counts and measured computation are not fully matched.

Which comparison answers which question? Each comparison tests a different explanation for success. The text chain compares answer accuracy under matched conditions. The single-source limit asks whether one participant's facts suffice. The count-only limit asks whether message arrangement adds information beyond symbol identities and frequencies. Replacement tests check whether answers follow changes to a source. A model could guess from question wording and appear accurate yet ignore substituted facts. Keeping questions fixed while changing facts distinguishes this shortcut from communication.

Answer scoring and statistical tests. The receiver outputs text, and a rule-based parser extracts the value inside the final answer tag to check against ground truth. Invalid answers, truncated responses, and failures to generate a message count as incorrect. Stored model outputs were rescored independently. We compute confidence intervals using 100,000 paired bootstrap resamples over the 16 test groups, keeping linked questions and conditions together. Both runs use the exact same groups. The three message-replacement comparisons in each run apply Bonferroni correction for multiple tests.

We set our success criteria and replacement thresholds before evaluating test data (Appendix B). In addition, a statistical *non-inferiority* test checks whether answer accuracy is no more than 2 percentage points (pp) below the text baseline: specifically, the lower bound of the paired difference (symbols minus text) must be at least -2pp in both runs. Other tests follow their own predefined criteria.

4 Results

4.1 Messages support new relations and new questions

We test the fixed final models on new direction combinations, reversed relations, and new entity–distance assignments. An additional planning experiment reuses existing messages. Table 2 summarizes accuracy; Figure 2 shows its uncertainty relative to text.

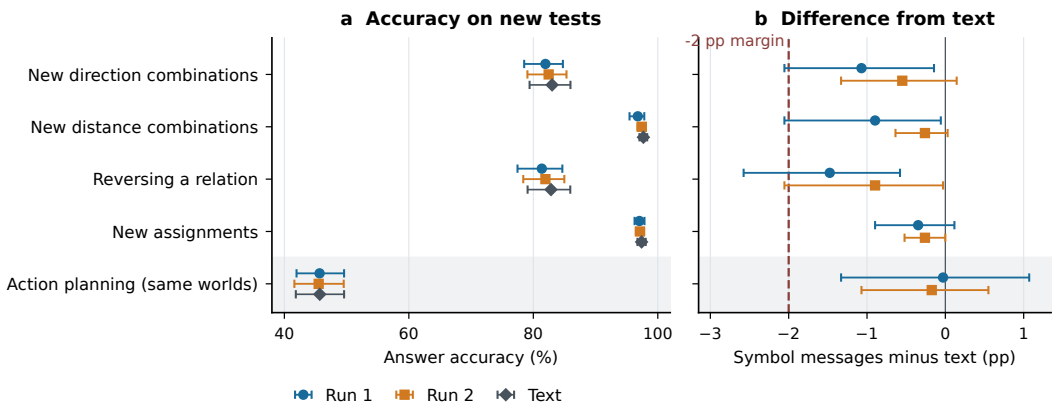


Figure 2: Accuracy compared with the text baseline. Left: symbol and text accuracy with 95% bootstrap confidence intervals. Right: paired differences (symbols minus text) against the allowed −2pp margin. Difference intervals are 95% for relation tests and 97.5% for planning. An interval extending left of the margin does not establish near-text accuracy, even when the means are close. The shaded row tests action planning on the same worlds.

Preserving which distance belongs to which entity. On the main balanced test, symbol communication achieves 97.05% and 97.14% accuracy across the two runs, compared with 97.40% for the text baseline. Both runs satisfy the predefined main requirements and the separate text-comparison margin. By comparison, count-only predictors reach only 34.95% and 33.33%. The lower confidence bounds on the improvement over these order-blind predictors exceed 60 pp in both runs, showing that the messages preserve relationships that symbol counts alone cannot capture. Detailed intervals are in Appendix E.

The 95% accuracy intervals are [96.27, 97.86]% (Run 1) and [96.47, 97.80]% (Run 2). In paired tests against text on the exact same worlds, the differences are -0.35 pp $[-0.90, 0.12]$ and -0.26 pp $[-0.52, 0.00]$. Because both lower bounds stay above the -2 pp non-inferiority margin, symbol communication performs comparably to text. We calculate these differences on matched answers from the same worlds rather than from the overlap of separate accuracy intervals. Both runs are compared with the same text answers, so the comparisons share that reference.

New combinations and reasoning operations. When spatial chains turn a corner instead of continuing along one axis, combined-direction accuracy is 81.94% and 82.47%. The lower bounds, 78.53% and 79.05%, exceed the exact 50% limit for either source alone. Each message supports eight later questions. A separate test asks three hypothetical relation-reversal questions per message, a question type absent from adapter training. Accuracy is 81.37% and 81.94%, meeting the predefined generalization and message-replacement requirements. The original pretrained models may already know how to reason about spatial relations; what transfers here is their ability to use the learned messages for those questions. Neither test establishes accuracy within 2pp of text in both runs.

The same message can support action planning. Keeping all models and adapters frozen, we give Phi a new instruction: plan a path of unit steps through a waypoint to a destination. An automated environment checks whether the steps visit the waypoint first, reach the correct endpoint, and take the shortest route. Accuracy is 45.66% and 45.52%, compared with the text baseline of 45.69%. The paired difference intervals ($[-1.33, 1.07]$ and $[-1.07, 0.55]$pp) satisfy the non-inferiority margin relative to text, but accuracy in both runs falls well below the predefined 80% target. Thus, the messages can be reused for planning, though neither symbols nor text make multi-step planning dependable. Often the model recites the correct spatial facts and then produces a faulty path (Appendix E). These results show reuse on new tasks within the same worlds, not general cross-domain transfer.

Table 3: Swapping information changes model answers. Accuracy (%) is scored against the original world facts (Run 1 / Run 2). The replacement columns swap in data from another controlled world. New-combination results include all eight questions (9,216 answers per run); other rows have 3,456. Planning reuses the main test worlds and messages.

Test	No replacement	Replace first message	Replace Mistral facts	Replace final message
New combinations	89.01 / 89.39	13.04 / 13.10	14.56 / 14.79	3.04 / 3.12
Reversing a relation	81.37 / 81.94	3.82 / 3.73	3.70 / 3.73	0.06 / 0.03
New assignments	97.05 / 97.14	0.46 / 0.41	0.12 / 0.14	0.61 / 0.67
Action planning	45.66 / 45.52	0.03 / 0.09	0.23 / 0.26	0.32 / 0.43

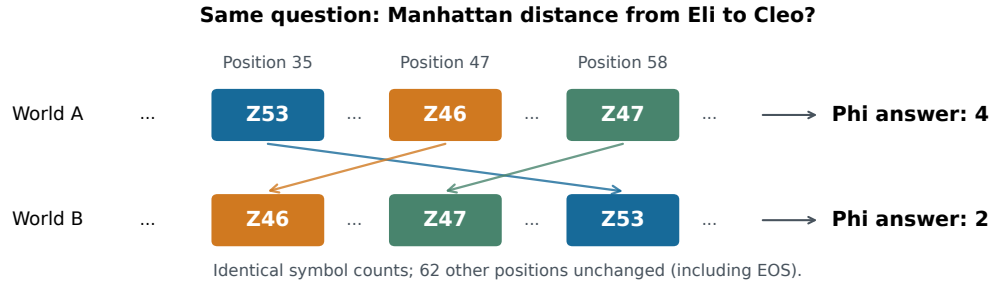


Figure 3: Identical symbol counts can express different spatial relationships. These two recorded 65-symbol messages (including EOS) differ at only three positions, while the remaining 62 positions are identical. Phi answers the same question correctly for both worlds (4 and 2, respectively). Lines connect corresponding symbols across messages; they illustrate differences between recorded worlds, not an experimental edit. The pair was picked in original file order.

4.2 Answers depend on the information in both messages

Table 3 shows what happens when information is replaced. On the main balanced test, accuracy scored against the original facts falls below 0.7% after any replacement. Yet when scored against the answers implied by the newly introduced facts, accuracy stays near 97%. The six Bonferroni-corrected lower bounds on these drops range from 95.20 to 96.18 pp. Answers in the relation-reversal test likewise track the swapped facts (81.37% and 81.94%). The answers follow the replacement content; the drop in the original-answer score is therefore not simply a failure to process an unfamiliar message.

A second analysis makes the role of message structure concrete. We find pairs of recorded examples with the same question and the same symbol counts in P_2, but different correct answers. Both answers are correct in 94.78% and 94.23% of the eligible pairs. Figure 3 shows the first pair by input order, selected without inspecting correctness. Three positions exchange symbols, and Phi answers 4 in one world and 2 in the other. This analysis reuses existing answers; it does not create new test observations or manipulate symbol order. It illustrates information in the arrangement of the symbols, while leaving its English-derived structure intact.

4.3 The receiving adapter also helps the model speak

Updating both Mouth and Ear improves direction accuracy by 8.94 pp [3.30, 16.72] and 4.95 pp [0.84, 10.04] compared with updating Mouth alone for the same number of optimizer steps. To see why, we hold the trained Mouth fixed and separately swap the incoming-message Ear and the read-back Ear between their initial and trained versions (Figure 4). Using the updated Ear in both roles raises the percentage of generated messages that exactly match the reference code from 59.72%/75.26% to 96.09%/99.83%. Both roles yield statistically significant gains with corrected confidence bounds above zero, with no evidence of an additional benefit beyond the sum of their separate effects. Training the receiver therefore helps a sender generate messages more accurately, though this does not pinpoint which internal layers carry out each calculation.

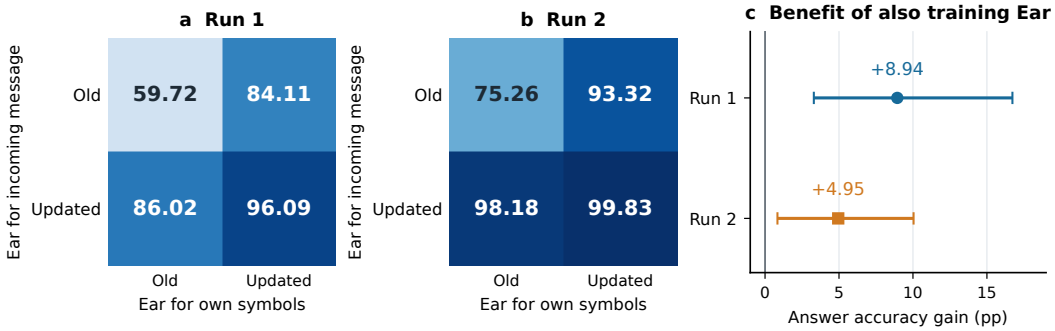


Figure 4: The Ear adapter aids both message reading and message generation. (a,b) Percentage of generated second messages that match the reference code exactly across 1,152 worlds, while keeping the trained Mouth fixed. (c) Gain in question-answering accuracy from updating Ear and Mouth together versus Mouth alone, measured on a separate test with 95% confidence intervals. Panels (a,b) measure symbol reconstruction, while panel (c) measures question answering.

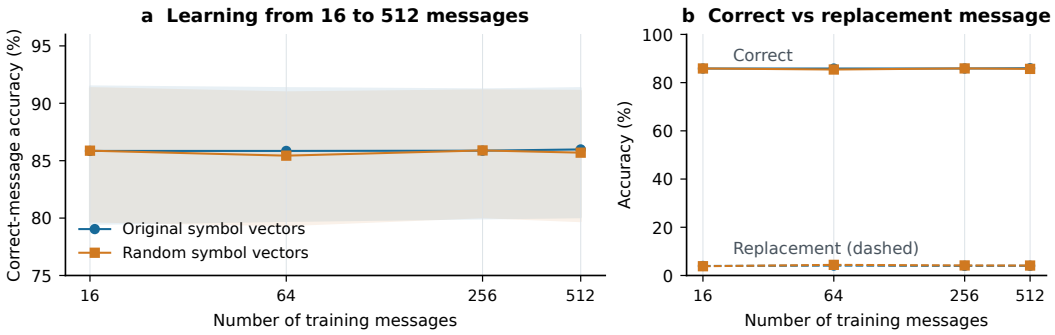


Figure 5: Teaching Phi to interpret messages from an unfamiliar sender. All runs use 4,000 updates on Qwen and Llama messages without question-answer labels. (a) Accuracy given correct messages, with pointwise 95% confidence bands. (b) Accuracy given correct versus replacement messages across the full 0–100% range. Both panels evaluate the same 576 worlds and compare original symbol embeddings against randomly assigned embeddings.

4.4 A NEW RECEIVER CAN UNDERSTAND AN UNSEEN SENDER

Phi's Ear is trained solely on messages from Qwen and Llama, and is then tested on messages from Mistral. In an experiment varying Phi's training set from 16 to 512 messages (with 4,000 updates in each setting), answer accuracy on an earlier set of 576 worlds remains stable between 85.85% and 85.98%, falling to around 4% when messages are replaced (Figure 5). Using randomly assigned symbol vectors yields similar accuracy, showing that the original vector assignment is not required for success on these worlds.

In practice, 16 messages provide 1,028 distinct local symbol contexts across 38 unique symbols, while 512 messages provide 32,928 contexts across 39 symbols. Thus, Phi learns a small task-specific subset of the 32,768-symbol vocabulary. Because total updates are fixed at 4,000, smaller datasets see the same number of gradient steps through repeated sampling.

4.5 DISCRETE SYMBOLS VERSUS VECTORS ON THE FINAL CONNECTION

We also compare two ways of transmitting the last message: selected symbols and the 512-dimensional FP16 vectors just before symbol selection. Both Phi receiving adapters have 7,970,977 parameters, identical initialization, 512 training messages, and 4,000 updates. Accuracy is 78.98% for symbols and 77.64% for vectors; replacement messages reduce it to 3.24% and 3.42%. The

paired vector-minus-symbol interval is $[-2.79, 0.05]$pp. This interval establishes neither a significant difference nor equivalence.

At fixed precision, the average second message uses 1,076.24 bits for symbols and 542,908.15 bits for vectors, a 504.45-fold ratio (Figure 6, Appendix F). This comparison excludes stored codebooks and adapters. Only the final connection changes; earlier message generation and read-back still use symbols. The ratio therefore describes this specific vector size and precision. An optimized continuous system and a full accuracy-versus-bits comparison remain to be tested.

4.6 What kind of communication has been learned?

The Ear reads symbols directly. Separately decoding them into English recovers all combined facts in 96.01% and 100% of main-test messages across the two runs. The same Phi model answers from this text at 96.99% and 97.40% accuracy, equivalent to Ear within the predefined $\pm$2pp margin. Joining and re-encoding the facts reproduces every second message in Run 2. The interface thus preserves an English-derived structure: this task requires final-model reasoning but no new deductions by the intermediate model (Appendix D).

Why compare a learned Ear with a fixed text decoder? Both methods read the same generated P_2, holding sender errors constant. The decoder supplies Phi with text; Ear supplies embeddings. Similar accuracy need not mean identical hidden states. The concatenation control shows that combined facts suffice here; testing computed results requires an intermediate deduction or state update.

5 Related work

Prior studies communicate through learned vectors (Du et al., 2026) or translated attention memories (Liu et al., 2026; Chen et al., 2026). We send integer symbols so messages can be inspected and replaced, testing whether facts survive two sending stages for questions hidden from both senders.

Learned input representations can condition frozen models (Li & Liang, 2021), including across tokenizers (Haltiuk & Smywiński-Pohl, 2025; Minixhofer et al., 2025). Our paired training comparison and separate tests measure Ear's two roles: reading incoming messages and reading back generated symbols, with the original Transformer frozen.

Iterated learning examines how compositional languages emerge (Ren et al., 2020). English-derived supervision lets us test learning an interface separately from inventing a language. Cross-model reuse can differ from jointly trained performance (Marincat, 2026), motivating Phi's new-sender test. Since accuracy alone can hide unused messages (Zhang & Emu, 2026), we separately replace each source and check whether answers follow the new facts.

6 Discussion and conclusion

Neural Esperanto lets frozen models exchange facts through learned symbol interfaces. On the balanced spatial test, a new receiver achieves near-text accuracy after two sending stages. Training Ear improves both reading incoming messages and generating new ones, while replacement tests show that answers depend on both information sources.

The evidence is strongest for spatial relations among familiar entities. Both runs share an English-derived initialization, and other relation tests do not all meet the same margin below text. Broader development tests have not established text-level transfer across domains. Phi's 24.75M-parameter Ear also has substantial computational capacity. Independent language formation, broader transfer, and a better accuracy–communication-cost trade-off remain open goals.

From a shared interface to a more general language. Future tests should train Phi to send as well as receive, require the intermediate model to transform facts before seeing final questions, and compare meanings and composition after matching symbols across codes: renaming English words alone does not create a new language. These tests would distinguish two-way participation, communication of computed changes rather than concatenation, and a structure genuinely different from the English-derived code.

REPRODUCIBILITY STATEMENT

The appendix details training, test construction, scoring, and uncertainty. Messages, model outputs, scoring checks, and development records are retained locally; an anonymized public release is pending.

ETHICS STATEMENT

The main experiments use synthetic worlds without personal records. Any release must respect model and dataset licenses. Symbol messages are difficult for people to inspect directly; high task accuracy alone does not establish that a model's communications are safe or controllable.

AI USE STATEMENT

The research idea, methodological design, and core algorithms were developed by the human authors. AI tools were used to assist with implementation, and all AI-generated code was reviewed and verified by the authors. GPT-6 Astra and Gemini 3.8 Flash were additionally used for language polishing and improving the clarity and readability of the manuscript.

REFERENCES

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

Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. *arXiv preprint arXiv:2511.09149*, 2026. URL https://arxiv.org/abs/2511.09149v5. Version 5.

Mykola Haltiuk and Aleksander Smywiński-Pohl. Model-aware tokenizer transfer. *arXiv preprint arXiv:2510.21954*, 2025. URL https://arxiv.org/abs/2510.21954v1. Version 1.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. URL https://arxiv.org/abs/2101.00190.

Jiyao Liu, Qi Zhang, Yaoyi Jia, Ziwen Kan, and Song Wang. Dual-cache latent space communication between heterogeneous language models. *arXiv preprint arXiv:2608.20617*, 2026. URL https://arxiv.org/abs/2608.20617.

Narcis Marincat. Portable semantics, private dialects: Reuse and negative transfer in latent communication between language-model cells. *arXiv preprint arXiv:2609.11365*, 2026. URL https://arxiv.org/abs/2609.11365.

Benjamin Minixhofer, Tyler Murray, Tomasz Limisiewicz, Anna Korhonen, Luke Zettlemoyer, Noah A. Smith, Edoardo M. Ponti, Luca Soldaini, and Valentin Hofmann. Bolmo: Byteifying the next generation of language models. *arXiv preprint arXiv:2512.15586*, 2025. URL https://arxiv.org/abs/2512.15586v1. Version 1.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2002.01365.

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

A Architecture and training details

Sending and receiving adapters. Mouth uses four attention heads and a 256-dimensional internal representation. Its attention query comes from the normalized current Transformer state, while keys and values come from normalized context states. A learned gate blends the attended context with a projection of the current state. Mistral keeps context from both the incoming message and its private facts. The shared Writer normalizes its 512-dimensional output, scores it against fixed symbol vectors, and clamps the score scale to $[1, 100]$. A separate head predicts EOS. Generation allows up to 128 steps, followed by a forced EOS if needed, for a maximum of 129 symbols. EOS is not allowed on the first step, and forced endings are recorded. Ear provides input embeddings, after which the receiving model builds its own attention memory.

Phi's Ear contains 24,748,193 parameters. It uses a 128-channel convolutional correction with kernel size 3 and dilations $1, 2, 4, 8$, covering 15 neighboring positions on each side. For every symbol, it predicts an ordered sequence of Phi input embeddings and its length. A predicted length may be zero. Padding is masked; an entirely empty output counts as an error. There is no fallback to the original text. Training combines normalized embedding mean-squared error and length cross-entropy. The reference tokenizer supplies these training targets offline; the learned communication path does not use it to decode messages at inference time.

Training stages. Table 1 summarizes initialization and training. Both runs start from the same pre-task reference-code parameters and independently train their task-specific adapters. Each run computes its cached Transformer states after training its own Ear. Llama supplies examples for Phi's receiver training, while Mistral is the intermediate sender in the main evaluation.

The final comparison uses AdamW with zero weight decay, batch size 32, microbatch size 8, and gradient clipping at norm 1. The Mouth learning rate is 10^{-4}; the trainable Ear uses 10^{-5}. Both conditions start from the same parameters within each run and use the same worlds. To reduce the dominance of frequent symbols, target counts n_z define weights

$$w_z = \mathrm{clip}_{[0.25,8]} \sqrt{\overline{n}_{\{z:n_z>0\}} / \max(n_z, 1)}. \tag{5}$$

Equation 3 normalizes over the full batch, including all microbatches. The frozen model runs in FP16 on V100; adapter normalization, cross-entropy, parameters, and optimizer states use FP32 where specified in the implementation. Alignment checks verify that the target symbol is absent from the state used to predict it.

B Evaluation protocol

What information each model receives. World records and questions are stored separately. Senders read only the world records, and the receiver sees a question only after the complete message exists. Mistral alone receives the second fact set. Replacing the first message or Mistral's facts requires Mistral to generate the second message again. Replacing the second message changes only Phi's input. For each replacement, we also compute the answer implied by the new facts. This additional score measures whether the model follows the replacement; the main accuracy measure always uses the original world's answer.

Scoring generated answers. The main test has three questions per world and allows 512 output tokens per answer. A fixed parser checks the final answer tag. An invalid final answer remains incorrect even if the explanation contains the right value. Run 1 has 13 invalid answers and two truncations; Run 2 has three invalid answers and two truncations. Text has three invalid answers and three truncations. These categories overlap. Neither learned run fails to generate an upstream message on this test. Output token IDs and termination flags are saved, although the record has no separate final EOS ID. Independent rescoring agrees for all 55,296 saved answer records across the original conditions. Several records refer to the same world, so they are not independent samples.

Confidence intervals and decision rules. The main test requires an accuracy lower bound above 80%, a count-only accuracy limit of at most 50%, an improvement above 15pp over that limit (lower

bound), and corrected lower bounds above 15pp for all three replacement effects. Both runs must meet these criteria.

Each test group contains 72 related worlds. Bootstrap sampling keeps all worlds, questions, models, and conditions within a group together. The main test uses 100,000 draws with seed 202609240502. Bonferroni correction covers the three primary replacement comparisons in each run. Both runs must meet the requirements, but they share the same text-baseline outputs. The text comparison separately requires the lower bound on the accuracy difference to be at least -2pp; failing to find a difference from zero would not establish that result. With only two training seeds, variation across independent training runs remains incompletely characterized.

Why symbol counts have a maximum achievable accuracy. Any fixed predictor given $g = (\text{test group}, q, b(P_2))$ must produce the same answer for every example in that group. It can correctly answer at most the number of examples carrying the most common label. Summing those counts gives Eq. 4. Randomizing predictions cannot improve the expected maximum. This bound applies to the observed test set and the specified inputs. A receiver with symbol order or other information is outside that restriction. Access to every test label deliberately gives the count-only predictor an advantage over a normally trained classifier.

What the comparisons control. The text chain matches facts, model identities, sending stages, delayed questions, generation limits, Qwen/Mistral input-adapter sizes, and additional training updates. Total gradient access and measured computation are not fully matched. Single-source limits describe a model missing one set of facts; they do not describe a model given everything at once. Replacement tests change content in whole messages. The count-only bound separately tests whether message structure adds information. The main balanced test was fixed before evaluation, while the Ear-role and fixed-decoder analyses reuse test sets already inspected.

C Learning to receive from a new sender

This experiment uses an earlier set of 576 worlds, separate from the final balanced test. Eight Phi Ears are trained for 4,000 updates each, with batch size 1,024. Smaller training sets are subsets of the larger sets, so their examples are repeated more often. Training uses neither question-answer labels nor Mistral messages. Table 4 compares original symbol vectors with randomly assigned vectors, including replacement-message controls.

Table 4: Phi answer accuracy (%) after training on Qwen/Llama messages. Every condition uses 4,000 updates. Replacement messages come from another world. Original and random refer to the vectors assigned to symbols.

	Original symbol vectors		Random symbol vectors	
Training messages	Correct message	Replacement	Correct message	Replacement
16	85.85	3.95	85.87	3.84
64	85.85	4.10	85.44	4.41
256	85.87	3.99	85.89	4.17
512	85.98	4.04	85.70	4.12

Sixteen messages contain 1,028 distinct local symbol contexts and 38 observed symbols; 512 contain 32,928 contexts and 39 symbols. The available alphabet has 32,768 entries, but this experiment teaches a small subset. Similar accuracy with random vectors shows that the original arrangement of symbol vectors is not necessary for success on these worlds.

To separate the Ear's two uses, we fix the updated Mouth and cross old and updated Ears for incoming messages and generated-symbol read-back. Figure 4 evaluates all four combinations on the same 1,152 worlds. Checking the decoded facts while allowing their order to change gives similar improvements to exact message matching. Both individual effects have positive corrected lower bounds. One run has a significantly negative interaction, giving no evidence of an extra benefit beyond the sum of the two effects. Changing the read-back Ear leaves the first symbol unchanged in all 4,608 paired checks, as expected before any symbol has been read back.

D How the learned messages relate to English

Decoding the reference code. The fixed BPE assignment maps 32,768 entries to distinct symbols, so its inverse is known without learning another mapping. All 1,152 first messages in each run exactly match the encoded source facts. Second messages exactly match the six combined facts in concatenated order for 1,106/1,152 worlds (96.01%) in Run 1 and 1,152/1,152 (100%) in Run 2. Exact text reconstruction and exact symbol matching give the same counts, without normalizing whitespace or punctuation. Including EOS, there are 55 and zero edits over 75,240 alignment positions, corresponding to 99.9269% and 100% symbol similarity. This offline analysis shows how closely the learned messages retain the English-derived reference structure.

Reading the same messages with a fixed decoder. We apply the known inverse and original BPE decoder to the generated P_2, then let the same frozen Phi read the resulting text through its own tokenizer. Only the receiver changes: messages are not repaired or regenerated. Precision, chat template, greedy decoding, the 512-token answer limit, and batches of 12 worlds with three questions each are held fixed.

Table 5: Two ways for Phi to read the same message: learned Ear versus a fixed text decoder. Accuracy (%) on 3,456 questions per run. Differences and paired 97.5% bootstrap intervals are shown in percentage points (pp). This comparison reuses the already evaluated balanced test set.

How Phi reads the message	Run 1	Run 2
Learned Ear (%)	97.05	97.14
Fixed text decoder (%)	96.99	97.40
Difference: Decoder minus Ear (pp)	-0.06	0.26
97.5% confidence interval (pp)	[-0.38,0.26]	[-0.03,0.58]

The comparison was specified before generating its new answers, but after examining the original test. It uses 100,000 paired bootstrap draws, seed 202609250121, and $\alpha = 0.025$ per run. Both intervals lie within the predefined $[-2, +2]$pp equivalence margin. Counts for both correct, decoder only correct, Ear only correct, and both wrong are $(3331, 21, 23, 81)$ and $(3339, 27, 18, 72)$. Independent rescoring agrees for all 27,648 new answers, including the replacement conditions. Invalid and truncated outputs remain in the score. The two receivers have similar accuracy on this task; their internal computations may still differ.

Does the intermediate model need to derive a new fact? A simple comparator decodes P_1, appends Mistral's private facts with newline separators, and re-encodes the result. It uses no model, question, answer, world identifier, relation parser, or distance solver. In each of four original conditions, it reproduces 1,106/1,152 second messages in Run 1 and all 1,152 in Run 2. Thus all 384 original receiver batches in Run 2 can be constructed without an intermediate LLM. A second implementation confirms that the inputs match; no new answer accuracy is measured. Mistral may carry out internal reasoning, but constructing these messages does not require it. This experiment demonstrates passing combined facts for later inference, rather than a task that needs every intermediate model to solve a subproblem.

E New relation combinations and new uses of existing messages

Table 6 gives the full intervals for the main balanced test. The summary table in the main text compares task means without pooling different test sets or changing their predefined criteria.

Combining and reversing relations. The new-combination test has 1,152 fresh worlds in 16 groups, with eight questions per world. After inspecting this test, we trained and evaluated the stronger text baseline; the original machine answers were unchanged. Run 1's lower bounds on the difference from text are -2.0544pp for both direction and distance, narrowly outside the -2pp margin. Both runs still exceed the single-source information limit and satisfy the replacement tests.

The relation-reversal test uses another 1,152 worlds in 16 groups, with three questions per message. Its predefined requirements are at least 75% accuracy, a 95% lower bound above the 50% single-

Table 6: Main balanced test results across 3,456 answers per run and condition. Brackets show paired 95% bootstrap confidence intervals. Accuracy uses percentages; differences use percentage points (pp). The same text answers serve both runs.

Metric	Run 1	Run 2
Learned symbol messages (%)	97.05 [96.27,97.86]	97.14 [96.47,97.80]
Text baseline (%)	97.40	97.40
Symbols minus text (pp)	−0.35 [−0.90,0.12]	−0.26 [−0.52,0.00]
Best count-only predictor (%)	34.95	33.33
Gain over count-only limit (pp)	62.09 [60.24,63.74]	63.80 [63.14,64.47]

source limit in both runs, and corrected lower bounds above 15pp for all three replacement effects. Both runs satisfy them. Accuracy intervals are [77.46,84.66]% and [78.39,84.98]%; differences from text are $[-2.58, -0.58]$ and $[-2.05, -0.03]$pp, outside the separate -2pp requirement. After all three replacements, scoring against the new facts gives 81.37% and 81.94%. Adapters, senders, checkpoints, questions, and generation limits were fixed before this test.

Action planning from existing messages. Models and adapters stay fixed, and the saved balanced-test messages are used without regeneration. Each world receives three requests for unit compass moves through a waypoint and on to a destination. A simulator checks visit order, endpoint, and minimal length. Prompts, greedy decoding, and the 512-token output limit match between symbols and text. Because text messages are identical across runs, the text route is evaluated once. Difference intervals use $\alpha = 0.025$ per run; six replacement intervals use $\alpha = 0.05/6$. All replacement-effect lower bounds exceed 39.8pp. Both comparisons meet the -2pp text margin, but neither reaches the 80% absolute-accuracy requirement. The 41,472 rescored outputs all concern these reused spatial worlds.

A later literal-text check finds that 3,283/3,324 machine outputs and 3,335 text outputs state both required facts correctly, out of 3,456 per route. Among those, 1,705/1,751/1,756 still produce an incorrect plan. Preserving the premises is therefore insufficient for successful action generation. This check describes the generated text, without changing the original scores or identifying the models' hidden reasoning steps.

Examples with the same counts but different answers. We pair records from the same run and test group that have identical questions and complete symbol counts, including EOS, but different correct answers. Selection does not depend on model correctness. There are 17,548/18,600 pairs; both answers are correct in 16,632/17,527, or 94.78%/94.23%. Descriptive bootstrap intervals are [93.26,96.28]%/[92.88,95.66]%. A record can occur in several pairs, so pairs are not independent samples. The figure uses the first pair in test-group, question, and world-ID order. Positions start at one and the symbol IDs are unchanged. This analysis does not regenerate answers or edit messages. An earlier replacement design found no eligible pairs with matching counts under its stricter donor rule; the descriptive analysis does not provide that missing intervention.

F Comparing symbols and vectors on the final connection

The earlier Run 1 Qwen/Mistral senders, their generated histories, and shared Writer remain fixed. Only the last transmitted representation changes: either a selected symbol, read through the fixed symbol dictionary, or the normalized 512-dimensional FP16 vector just before symbol selection. Both new Phi receiving adapters take vectors and have 7,970,977 parameters. They share initialization, 512 Qwen/Llama training messages, 4,000 Adam updates, embedding/length targets, and position sampling. Training uses no question-answer labels or Mistral examples. There is no trainable correction indexed by symbol ID. These two matched adapters are smaller than the main Phi Ear.

Evaluation reuses 1,152 worlds in 16 groups, with eight questions per world. All 36,864 answers were rescored independently. Symbols versus vectors achieve 70.86% versus 68.95% on combined direction, and 83.65% versus 83.07% on combined distance. When a message is replaced, accu-

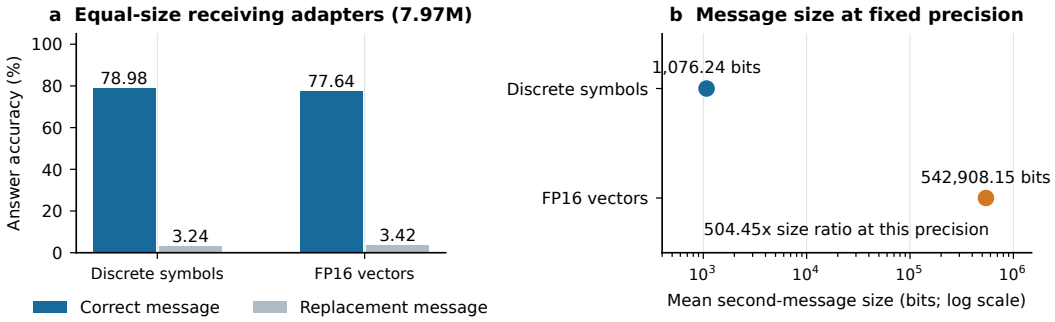


Figure 6: Comparing symbols and vectors with equally sized receiving adapters. Left: answer accuracy (%) with correct and replacement messages; the paired confidence interval for vectors minus symbols is $[-2.79, 0.05]$ pp. Right: second-message size in bits at fixed precision. This measures transmitted data at the specified precision, excluding stored dictionaries, adapter weights, and network overhead. It does not compare fully optimized communication systems.

racy measured against the new replacement facts is 79.04% for symbols and 77.72% for vectors, confirming that both representations affect answer content.

For L content symbols, the discrete message uses $16(L + 1)$ bits including EOS. The vector message uses $8193L + 1$ bits including termination indicators. Average sizes are 1,076.24/542,908.15 bits, a 504.45-fold ratio. This excludes stored dictionaries, adapter weights, serialization overhead, training, and loading. The comparison uses one vector dimension and precision, without entropy coding, throughput measurements, or end-to-end retraining of the continuous channel.

Comparison with text size. On the main balanced test, both symbol messages together use 1,568 fixed-width bits, versus 1,577.29 for the best measured lossless text encoding. Text encoding is slightly smaller on the combination and reversal tests. Thus the final-connection vector comparison does not establish greater overall efficiency than natural language. Measuring the best attainable accuracy at different message sizes remains future work.

G A RECORDED MODEL-TO-MODEL EXCHANGE

The example in Figure 1 is world `BNC-0-0-0-0-0` from Run 1, selected by input order. It is the same world as the first member of the message pair in Figure 3. The following facts, symbol IDs, and final answers come from the stored messages and model outputs; no additional model generation is used. Source wording is reproduced verbatim, including "1 grid steps."

Qwen: read private facts and send a message. The source receives these three statements:

> Ivan is 2 grid steps south of Ari.
> Hana is 3 grid steps south of Gus.
> Eli is 1 grid steps south of Ben.

Qwen's original Transformer reads this text. Its sending adapter attends to the available context, and the shared output layer selects each next symbol. The emitted symbol is read back through Qwen's Ear before the next step. The complete first message, with the `Z` prefix omitted to improve readability, is:

```
14751 4 47 27305 18572 707 3 7253 0 7 413 823 4 53 27305
18572
707 3 10042 0 7 2257 16 4 46 27305 18572 707 3 1107 0 EOS
```

There are 31 content symbols and EOS; EOS has ID 32768. The model generated this message one symbol at a time, without receiving the target sequence during evaluation.

Mistral: receive a message and add new facts. Mistral's Ear converts P_1 into model input embeddings. Mistral additionally reads the following text through its own tokenizer:

> Ben is 3 grid steps west of Cleo.
> Ari is 1 grid steps west of Faye.
> Gus is 2 grid steps west of Dora.

The original Transformer processes the received embeddings together with these private facts. Mistral's Mouth and the shared Writer generate the second message:

```
14751 4 47 27305 18572 707 3 7253 0 7 413 823 4 53 27305
18572
707 3 10042 0 7 2257 16 4 46 27305 18572 707 3 1107 0 7
7889 4 53 27305 18572 894 3 8732 9 0 7 153 819 4 46 27305
18572 894 3 18639 0 7 424 385 4 47 27305 18572 894 3 26798 0
EOS
```

This message has 64 content symbols and EOS. Decoding it separately with the reference tokenizer recovers the three Qwen facts followed by the three Mistral facts, with no symbol errors in this example. The shared code mapping lets human readers inspect this content, but the models communicate directly without an English text decoder in the loop.

Phi: answer after both messages are complete. The same P_2 is passed through Phi's Ear for all three delayed questions. Phi then processes the question text and generates a text answer:

Question	Recorded answer
What is the Manhattan distance in grid steps from Eli to Cleo?	4
What is the Manhattan distance in grid steps from Ivan to Faye?	3
What is the Manhattan distance in grid steps from Hana to Dora?	5

For the first question, the two displacements are one step south and three steps west, giving Manhattan distance $1 + 3 = 4$. The other chains give $2 + 1 = 3$ and $3 + 2 = 5$. All three recorded answers match their correct answers. Neither sender saw these questions, and the second message was not changed between questions. Llama supplies training messages for Phi's receiver training but does not participate in this particular chain of models.