MAS-BIT: A MULTI-AGENT SYSTEM FOR END-TO-END BRAIN-TO-TEXT DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speech brain-computer interfaces aim to restore communication by translating neural activity into text, yet end-to-end decoders continue to trail cascaded systems in accuracy. We introduce a **M**ulti-**A**gent **S**ystem for **B**ra**I**n-to-**T**ext (**MAS-BIT**), the first multi-agent framework for end-to-end brain-to-text decoding from intracortical recordings. MAS-BIT coordinates three specialized agents: a neural agent that interprets continuous neural representations, a phonetic agent that decodes soft phoneme posteriors, and an arbitration agent that integrates their complementary evidence with direct access to neural representations to generate the final transcription. The expert agents communicate with the arbiter through token-embedding-based latent communication, constructing continuous messages as probability-weighted combinations of token embeddings. These messages retain alternative token hypotheses and provide a differentiable interface through which sentence-level supervision jointly trains the neural encoder and specialized agents. MAS-BIT achieves state-of-the-art end-to-end decoding performance on the Brain-to-Text '24 and '25 benchmarks, with word error rates of 8.418% on '24 and 4.878% on the '25 public test set (5.288% on the private test set). These results narrow the gap to cascaded systems without ensembling independently trained models or using external language-model rescoring.

1 INTRODUCTION

Speech brain-computer interfaces (BCIs) aim to restore communication by translating neural activity associated with attempted speech into text (Willett et al., 2023; Card et al., 2024). Accurate decoding requires recovering linguistic content from noisy recordings and limited paired training data. Cascaded systems combine a neural-to-phoneme decoder with language-model-based sentence reconstruction. In contrast, end-to-end systems connect neural representations directly to a language decoder and optimize for the target transcription (Feng et al., 2024; Zhang et al., 2026). In the Brain-to-Text benchmark comparisons reported by Zhang et al. (2026), end-to-end decoding remains less accurate than cascaded decoding.

End-to-end decoding is nevertheless attractive because it jointly optimizes neural representations and sentence generation while retaining direct access to continuous neural evidence. Cascaded systems require a separate phoneme-to-language decoding stack, including large n-gram/WFST resources and language-model rescoring, which can impose hundreds of gigabytes of host-memory overhead. We therefore ask whether end-to-end decoding can approach cascaded accuracy while preserving a unified, substantially lighter inference pipeline.

A neural encoder provides two views of the same recording: continuous hidden representations and phoneme posteriors obtained from a supervised phoneme readout. The hidden representations expose learned neural features to a language decoder, while the posteriors organize this evidence around phoneme identities and their relative probabilities. Posterior-based interfaces in speech recognition illustrate how such structured distributions can condition a language model (Ma et al., 2025). For brain-to-text decoding, these two views suggest a division of labor: one expert interprets continuous neural representations, and another interprets explicit phonetic evidence. Their different inputs can lead to different transcription hypotheses, even though both originate from the same neural observation. We investigate whether these specialists can learn to provide complementary evidence for a common final transcription.

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.

The communication interface determines which parts of the experts' judgments reach the final decoder. Passing a single generated text sequence conveys the selected tokens but discards their relative support against alternative tokens at each position. The discrete selection also blocks ordinary backpropagation through the message into the expert probabilities that produced it. Prior work on differentiable communication and continuous exchange between language-model agents provides alternatives to this text interface (Foerster et al., 2016; Pham et al., 2024; Zou et al., 2026). For neural decoding, we seek an interface that conveys information about competing token hypotheses and allows the final transcription objective to shape the experts through their messages.

Here, we introduce an end-to-end **M**ulti-**A**gent **S**ystem for **Bra**I**n**-to-**T**ext (**MAS-BIT**) that separates specialist decoding from arbitration. A *neural expert* reads continuous encoder representations, a *phonetic expert* reads soft phoneme posteriors, and an *arbiter* integrates their messages with direct neural evidence. The roles share one language-model backbone with separate adapters. Rather than passing only selected text, the experts communicate through probability-weighted token embeddings that retain alternative hypotheses and allow the final transcription loss to supervise the experts through their messages. Alternating updates train the encoder and the agents in a single end-to-end system.

We evaluate participant-specific models on the Brain-to-Text '24 and '25 benchmarks, obtaining 8.418% WER on the '24 holdout and 4.878%/5.288% on the '25 public/private subsets. On the '24 holdout and '25 public subset, MAS-BIT achieves state-of-the-art end-to-end performance, surpassing BIT's single-model decoder and end-to-end ensemble without ensembling independently trained models or external language-model rescoring (Zhang et al., 2026). Controlled comparisons show that specialized experts outperform a single decoder given both views, with further gains from latent communication and staged training. The arbiter outperforms reference-aware selection of the better complete expert draft and recovers correct transcriptions when both drafts are wrong. Together, these results show that jointly trained specialists can make better use of neural and phonetic evidence than simply feeding both views to one decoder or selecting between expert drafts.

2 Related Work

Speech neuroprostheses and end-to-end neural decoding. Intracortical speech neuroprostheses commonly decode intermediate phonetic units before language-model-based sentence reconstruction (Willett et al., 2023; Card et al., 2024). More recent work directly connects neural representations to pretrained language models (Feng et al., 2024; Zhang et al., 2026). In particular, BIT combines cross-task, cross-species neural pretraining with a phoneme-adapted encoder and an audio-LLM decoder (Zhang et al., 2026). MoDAl combines parallel neural encoders through alignment and decorrelation losses (Chen & Chin, 2026), while BrainWhisperer adapts a pretrained Whisper encoder–decoder to neural inputs (Boccato et al., 2026). Other direct decoders use a BART word decoder (Olak et al., 2026), transfer Wav2Vec2 representations to character CTC decoding (Fiedler et al., 2025), or train a character-CTC Conformer (Khanday et al., 2026). In a recent preprint, Fogg et al. (2026) train a shared neural phoneme decoder across six participants using participant-specific input projections.

Posterior-based interfaces and signal-grounded refinement. CTC posteriors provide a probabilistic interface between low-level sequence representations and language models. LegoSLM uses posterior-weighted token embeddings to connect a speech encoder to an LLM without committing to a one-best intermediate sequence (Ma et al., 2025). Related deliberation systems refine first-pass hypotheses while retaining access to the original signal (Hu et al., 2020; Dekel et al., 2026). MAS-BIT combines these ideas through a neural expert that reads continuous encoder states, a phonetic expert that reads CTC posteriors, and an arbiter that integrates their messages with direct neural evidence.

Multi-agent and differentiable language-model communication. Multi-agent language-model systems improve generation by separating proposal generation from downstream aggregation (Du et al., 2024; Wang et al., 2025). Continuous communication further allows information to pass through differentiable representations rather than only discrete text (Foerster et al., 2016; Pham et al., 2024; Ramesh & Li, 2025; Zou et al., 2026). MAS-BIT adopts a fixed two-expert–one-arbiter topology with a shared frozen language-model backbone and role-specific LoRA adapters (Hu et al.,

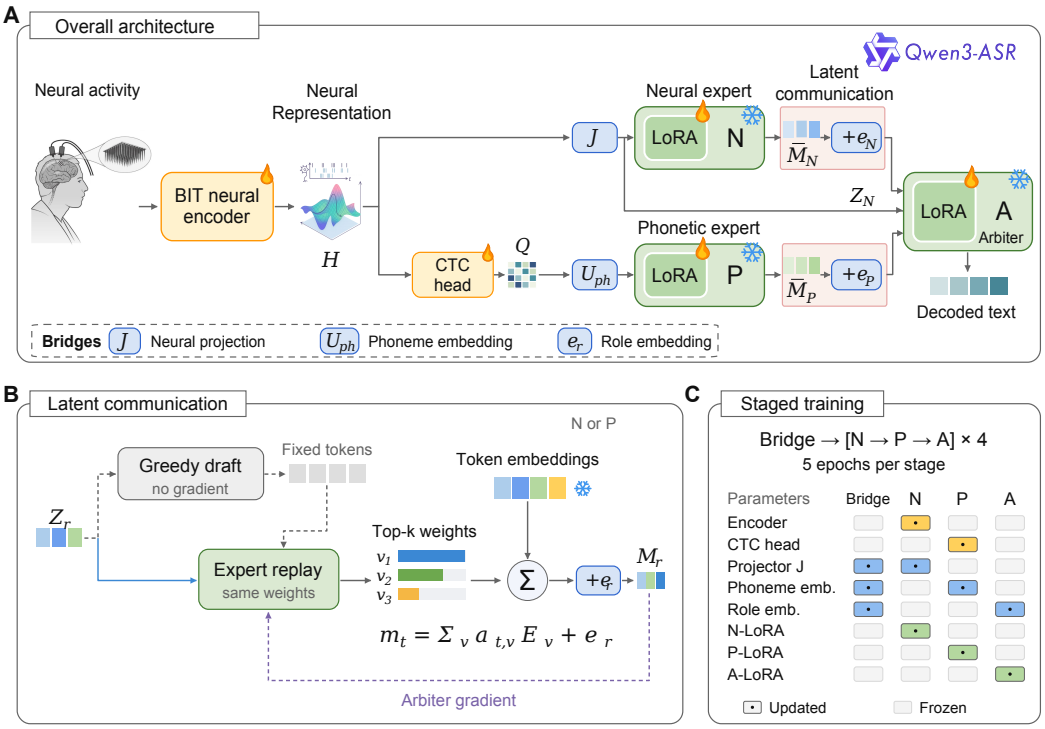


Figure 1: *MAS-BIT combines neural and phonetic evidence through specialized experts and an arbiter.* **(A)** The experts read hidden states and phoneme posteriors from a shared BIT-initialized encoder. The arbiter combines their messages with direct neural input. All three roles share a frozen text backbone with separate adapters. **(B)** Each expert replays its greedy draft to construct probability-weighted token embeddings. Z_r denotes its input, Z_N or Z_P, and e_r identifies the message source. The arbiter loss trains active upstream components through these latent messages. **(C)** Five bridge epochs precede four cycles, each consisting of five neural-expert, phonetic-expert, and arbiter epochs. Filled cells indicate parameter updates; frozen downstream weights retain input gradients.

2022). Each expert specializes in its input evidence, and its generated draft is replayed as latent token-level messages optimized via the downstream arbiter loss.

3 METHODS

MAS-BIT maps a neural recording to a sentence through a neural expert N, a phonetic expert P, and an arbiter A (Figure 1). The experts read complementary representations of the same recording and send latent token-embedding messages to the arbiter. The arbiter combines these messages with direct neural evidence to produce the final transcription. We describe the shared-backbone architecture and its cyclic training procedure.

3.1 ARCHITECTURE

Neural and phonetic evidence. Let $X \in \mathbb{R}^{T \times C}$ denote an utterance of intracortical features, $y = (y_1, \ldots, y_U)$ its text-token sequence including the end-of-sequence token, and π its phoneme target. The input combines threshold-crossing activity and spike-band power (Nason et al., 2020); we describe preprocessing in Appendix B.1. A participant-specific input projection and a neural Transformer shared by the three agents produce $H = f_\theta(X) \in \mathbb{R}^{S \times d_n}$, where $S = \lceil T/5 \rceil$ and $d_n = 384$. We initialize MAS-BIT with the released BIT encoders for T12 and T15, along with their corresponding phoneme readouts (Zhang et al., 2026).

A linear projector J maps the shared neural states into the language model's embedding space. A linear phoneme readout (CTC head) produces phoneme logits, which are normalized into posteriors

and combined with a learned phoneme embedding table:

$$Z_N = J(H) = HW_J + b_J, \qquad Q = \mathrm{softmax}(HW_C + b_C), \qquad Z_P = QU_{\mathrm{ph}}. \tag{1}$$

Here, $d = 2048$ is the language-model width, and $Q \in \mathbb{R}^{S\times 41}$ contains phoneme posteriors over 39 speech phones, silence, and the CTC blank. The learned table $U_{\mathrm{ph}} \in \mathbb{R}^{41\times d}$ converts each posterior into its expected phoneme embedding. Thus, Z_N retains continuous neural features, while Z_P expresses the evidence in terms of the supervised phoneme inventory. The phonetic interface preserves the full distribution, including blank probability, at every valid patch position.

Specialized language agents. All three agents use the Qwen3-ASR-1.7B (Shi et al., 2026) text decoder. We share and freeze its pretrained weights and token embedding table, while each role has an independent LoRA adapter (Hu et al., 2022). Appendix B.1 lists the adapter settings. The neural projector, phoneme embedding table, message-role embeddings, and role adapters are initialized for the MAS-BIT run.

The neural expert N conditions on Z_N and predicts an English transcription directly from neural representations. The phonetic expert P conditions on Z_P and also predicts English text tokens; its phonetic specialization comes from the posterior-based input. The arbiter A conditions on Z_N together with the two expert messages defined below. Each role conditions on its continuous input sequence and generates text autoregressively. Role-specific prompts describe the input evidence and request an English transcription (Figure 2A; Appendix B.1). Together, the input evidence, prompt, and separate adapter define each decoding role. Padding positions are masked.

3.2 Latent communication

Each expert $r \in \{N, P\}$ receives a continuous input Z_r: projected neural states Z_N for the neural expert, or posterior-weighted phoneme embeddings Z_P for the phonetic expert. Conditioned on this input, the expert generates a greedy draft $\hat{y}^r$. Draft generation uses the expert's previously generated tokens and stops at an end-of-sequence token or a fixed length cap. During training, this rollout runs without gradient tracking and with language-model dropout disabled. We then replay the generated draft through the same expert to compute next-token logits

$$\ell_t^r = g_r(Z_r, \hat{y}^r_{<t}). \tag{2}$$

The replay is differentiable and uses the expert's own draft as its text context. Let $\mathcal{K}_t^r$ be the indices of the 16 largest logits at position t. With temperature $\tau = 1$, the message content is a probability-weighted token embedding (Pham et al., 2024):

$$a_{t,v}^r = \frac{\exp(\ell_{t,v}^r/\tau)}{\sum_{u\in\mathcal{K}_t^r}\exp(\ell_{t,u}^r/\tau)}, \qquad \bar{m}_t^r = \sum_{v\in\mathcal{K}_t^r} a_{t,v}^r E_v, \tag{3}$$

where $E_v \in \mathbb{R}^d$ is the shared frozen embedding of text token v. The weights are renormalized over the selected support, allowing several candidate tokens to contribute to each message position. We add a learned message-role embedding e_r to each message token to indicate whether it originates from the neural or phonetic expert. At inference, this gives $m_t^r = \bar{m}_t^r + e_r$. Training applies dropout to the message content before adding the role embedding (Appendix B.2).

The arbiter's continuous prefix is the concatenation

$$Z_A = [Z_N; M_N; M_P], \qquad p_A(y \mid X) = \prod_{t=1}^{U} p_A(y_t \mid Z_A, y_{<t}). \tag{4}$$

Message masks retain valid draft positions, including the first end-of-sequence token. This defines a single round of expert-to-arbiter communication with direct neural conditioning. The arbiter loss backpropagates through the selected softmax weights and replay computation into active expert adapters, input interfaces, and the neural encoder. The discrete draft tokens and top-k membership are held fixed during this backward pass.

Inference. At inference, the model encodes the recording once, generates and replays the two expert drafts, and greedily decodes the arbiter's final sentence. Neural augmentation and dropout are disabled. The arbiter's transcription is the final system output.

3.3 Training objectives and staged training

We supervise each agent with teacher-forced cross-entropy on the target sentence, while constructing the expert messages from generated drafts. Let $\mathcal{L}_{\mathrm{CE}}^{r}$ denote the summed token loss for role r, normalized by the number of target text tokens, and $\mathcal{L}_{\mathrm{CTC}}^{(l)}$ the phoneme CTC loss (Graves et al., 2006) at encoder layer l, normalized by the number of target phones. We apply auxiliary CTC losses to intermediate encoder layers (Lee & Watanabe, 2021). The training objective is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{CE}}^{N} + \mathcal{L}_{\mathrm{CE}}^{P} + \mathcal{L}_{\mathrm{CE}}^{A} + 0.3\,\mathcal{L}_{\mathrm{CTC}}^{(7)} \\ + 0.05\,\rho\,\frac{\mathcal{L}_{\mathrm{CTC}}^{(3)} + \mathcal{L}_{\mathrm{CTC}}^{(5)}}{2}.\end{aligned} \tag{5}$$

The coefficient ρ increases linearly from zero to one over the first epoch in which the encoder is updated and remains one thereafter. All three sentence losses are computed in every training stage; the stage determines which parameters are updated. Text and phoneme losses retain their separate denominators, with the distributed normalization specified in Appendix B.4.

Inspired by the neural augmentation approach of Fogg et al. (2026), all three roles share the same augmented recording during expert generation, replay, and sentence supervision. Augmentation includes temporal and electrode perturbations, as well as patch masking (Appendix B.3).

Staged training. Training begins with five bridge epochs that update the neural projector, phoneme embedding table, and message-role embeddings while keeping the encoder, CTC head, and role adapters fixed. It then repeats five epochs each of N, P, and A updates for four cycles:

$$B_5 \;\rightarrow\; [N_5 \;\rightarrow\; P_5 \;\rightarrow\; A_5]^4. \tag{6}$$

Subscripts denote epoch counts, giving a maximum of 65 epochs. Figure 1C summarizes the stage-specific updates; Table 3 in Appendix B.4 lists the active parameter groups. Each block updates only its designated parameter groups.

Frozen downstream modules transmit input gradients, allowing the arbiter objective to supervise active upstream components, but receive no parameter or optimizer-state updates.

We use AdamW (Loshchilov & Hutter, 2019) with a block-local warmup–cosine learning-rate schedule; optimization settings are given in Appendix B.4.

4 Evaluation

4.1 Tasks

We evaluate MAS-BIT on attempted speech decoding using the Brain-to-Text '24 and '25 datasets from participants T12 and T15, respectively (Willett et al., 2023; 2024; Card et al., 2024). In this task, neural activity is recorded while participants attempt to articulate a sentence, and the decoder predicts the sentence's transcription. We measure performance by corpus word error rate (WER): the total number of word substitutions, deletions, and insertions divided by the number of reference words; lower WER indicates more accurate decoding.

Models are trained separately for each participant, using 8,800 training utterances for T12 and 8,072 for T15. We use all 880 and 1,426 provided labeled validation utterances for checkpoint selection. Our benchmark comparisons, including the three controls, report performance on the hidden holdouts of 1,200 and 1,450 utterances. T15 public and private holdout scores are reported separately. Checkpoint selection uses word-edit counts on the provided labeled sets, with decoding settings fixed before holdout evaluation. All predictions are generated without access to reference transcriptions, and every utterance in the declared split contributes to evaluation. Appendix B.5 details the checkpoint selection and decoding settings.

4.2 Baselines

We benchmark MAS-BIT against published end-to-end and cascaded decoders, examine candidate-sentence counts in our cascaded reproduction, and evaluate three ablations of the multi-agent framework.

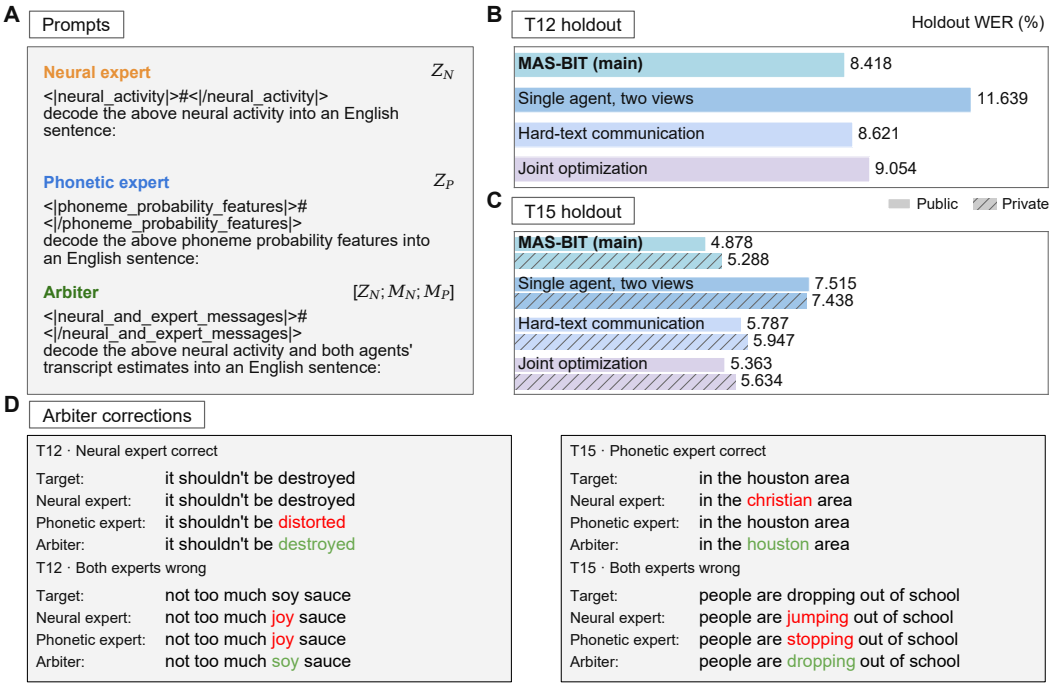


Figure 2: *MAS-BIT benefits from expert specialization, latent communication, and staged training.* **(A)** Role-specific prompts; # marks the continuous-input slot. **(B–C)** T12 and T15 holdout WER; solid and hatched bars show T15 public and private. The controls share the backbone and encoder initialization. Bars have the same zero-origin scale. **(D)** Selected validation transcripts show the arbiter matching a correct expert and recovering sentences when both drafts are wrong. Expert errors are red; corresponding correct arbiter words are green.

Published baselines. Our primary comparison uses reported corpus WER on matched participant and hidden-holdout subsets (Table 1). End-to-end decoding without ensembling generates text from neural input without an external WFST, n-gram model, pronunciation lexicon, or independent language-model reranker; phoneme pretraining, auxiliary CTC losses, and staged training are compatible with this definition. BIT End-to-End + Ensemble is listed separately, and published cascaded references appear in Appendix E. Appendix E separately summarizes published results on other evaluation sets.

Candidate-sentence analysis. We reproduce cascaded decoding with the released BIT encoder for each participant. MAS-BIT is initialized from the same encoder. Following Zhang et al. (2026), we pass CTC phoneme logits to an OpenWebText2 5-gram language model and rescore the candidate sentences with OPT-6.7B (Zhang et al., 2022). We vary the number of retained candidates, $K \in \{1, 2, 3, 4, 6, 9, 12, 18, 25, 37, 50, 75, 100\}$, while keeping the encoder, language models, and search settings fixed. Appendix D describes lattice rescoring and candidate selection.

Multi-agent ablations. We compare MAS-BIT with *Single agent, two views*, *Hard-text communication*, and *Joint optimization*. All controls share the backbone, participant-specific initialization, data, augmentation, and checkpoint-selection procedure; corresponding components start from the same weights. Appendix B.6 defines them and includes the schematic in Figure 6.

5 Experiments and Results

5.1 Specialization, communication, and staged training improve decoding

Three controls examine the proposed design choices from the same encoder initialization (Figure 2B–C).

Specialization makes better use of the same input evidence. Giving a single agent both neural representations and phoneme posteriors produces the largest degradation. Its holdout WER is higher than MAS-BIT by 3.22 percentage points on T12 and 2.64/2.15 points on T15 public/private. Specialist decoding followed by arbitration therefore makes better use of the two input views.

Latent messages improve upon discrete expert drafts. Replacing latent messages with hard text increases holdout WER by 0.20 points on T12 and 0.91/0.66 points on T15 public/private. The latent interface retains alternative token hypotheses and allows the arbiter loss to update expert probabilities through replay. The hard-text control instead passes the selected draft tokens and blocks this gradient path.

Token alternatives help resolve expert disagreements. On the validation sets, retaining alternative tokens at fixed model weights and expert drafts improves decoding on T12, particularly when the experts disagree. The aggregate effect is small on T15, and shuffling the probabilities of non-leading alternatives produces small changes on both participants (Appendix C.3).

Alternating updates improve the shared system. Joint optimization increases holdout WER by 0.64 points on T12 and 0.49/0.35 points on T15 public/private. Alternating updates improve accuracy under the matched epoch budget. Each stage retains supervision from the final transcription while restricting parameter updates to the active components.

5.2 Expert messages and direct neural evidence improve arbitration

The validation examples in Figure 2D show the arbiter resolving disagreements and recovering sentences that both experts decode incorrectly. Across the full validation sets, arbitration yields more exact matches and fewer word edits than an offline, reference-aware oracle that selects the better complete expert draft (Appendix C). Some correct transcriptions also contain words absent from both drafts, extending recovery beyond aligned word recombination (Figure 3; Appendix C.1). The arbiter can draw on both the expert messages and its direct neural input when making these corrections.

With the trained model fixed, masking either expert message reduces overall decoding accuracy, and masking both produces a larger decline. Zeroing the direct neural input while retaining both messages also reduces recovery of sentences missed by both experts. Table 8 reports input-masking results.

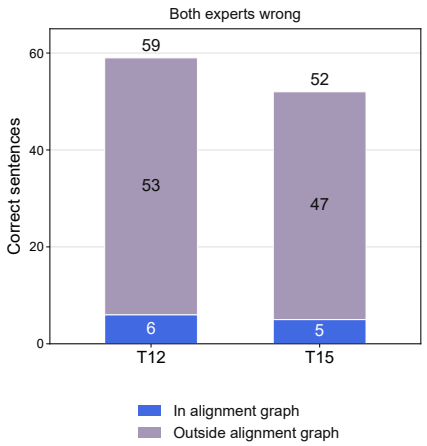


Figure 3: *Arbitration recovers sentences beyond draft recombination.* Exact recoveries when both experts are wrong, on the full validation sets.

5.3 Multi-agent decoding improves upon BIT's end-to-end ensemble

Starting from BIT's released participant-specific encoders, MAS-BIT achieves 8.418% WER on the Brain-to-Text '24 holdout and 4.878%/5.288% on the public/private subsets of the '25 holdout (Figure 4A). Relative to BIT's single-model end-to-end decoder, WER decreases by 46.3% on T12 and 55.9% on T15 public (Zhang et al., 2026). It also reduces WER relative to BIT's end-to-end ensemble by 17.6% on T12 and 37.1% on T15 public. Coordinating specialized experts within one jointly trained system thus improves upon BIT's aggregation of independently trained decoders.

5.4 MAS-BIT surpasses the reproduced cascade on T15

We next compare MAS-BIT with cascaded decoding using the same released BIT encoder checkpoint that initializes MAS-BIT. The cascade keeps this encoder fixed, while MAS-BIT adapts it

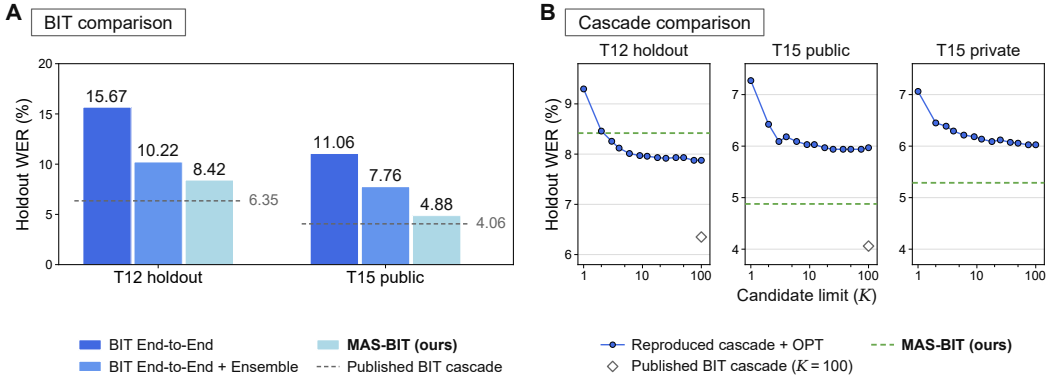


Figure 4: *MAS-BIT improves upon BIT's end-to-end ensemble and surpasses the reproduced cascade on T15.* **(A)** Published BIT end-to-end results (Zhang et al., 2026) and MAS-BIT on T12 holdout and T15 public. **(B)** Our cascade uses the released BIT encoder that initializes MAS-BIT. Blue curves show OPT rescoring at each candidate limit; green lines show MAS-BIT. Gray diamonds at $K = 100$ mark the published BIT cascade results on T12 and T15 public. All panels report corpus holdout WER (%, lower is better); candidate limits use a logarithmic axis.

Table 1: *Published end-to-end benchmark results.* Values are corpus WER (%); Feng and BIT scores follow Zhang et al. (2026). MoDAl reports mean ± 95% CI over five seeds; MoDAl-Full also uses area 44. BrainWhisperer-cross uses additional neural datasets. *Competition WER provisionally listed under T15 public; public/private is not explicitly identified for these entries. – denotes an unreported result.

	T12	T15	
Method	**Holdout**	**Public**	**Private**
End-to-end decoding without ensembling			
Feng et al. (2024)	24.69	–	–
RNN encoder + E2E LLM decoder (Zhang et al., 2026)	19.9	16.0*	–
MoDAl-1 (Chen & Chin, 2026)	18.8 ± 0.5	–	–
MoDAl-Full (Chen & Chin, 2026)	17.7 ± 0.4	–	–
BIT-TFS End-to-End (Zhang et al., 2026)	16.5	12.2*	–
BIT-Human End-to-End (Zhang et al., 2026)	15.8	11.4*	–
BIT End-to-End (Zhang et al., 2026)	15.67	11.06	–
BrainWhisperer (Boccato et al., 2026)	–	10.2*	–
BrainWhisperer-cross (Boccato et al., 2026)	–	8.7*	–
MAS-BIT (ours)	**8.418**	**4.878**	**5.288**
End-to-end decoding with ensembling			
BIT End-to-End + Ensemble (Zhang et al., 2026)	10.22	7.76	–

through end-to-end training. MAS-BIT achieves lower holdout WER on both T15 subsets across all tested candidate limits up to 100 sentences (Figure 4B). Its margins over the lowest reproduced WER are 1.06 and 0.74 percentage points on T15 public and private, respectively. On T12, MAS-BIT outperforms the cascade with one or two candidates and approaches its best WER: 8.418% compared with 7.877%, a gap of 0.541 percentage points. The published BIT cascade uses $K = 100$ and remains more accurate on T12 and T15 public, at 6.35% and 4.06% WER (Zhang et al., 2026); gray diamonds mark these results in Figure 4B.

5.5 MAS-BIT ESTABLISHES A NEW STATE OF THE ART IN END-TO-END DECODING

Across published end-to-end systems on matched benchmark subsets, MAS-BIT achieves the lowest reported WER on T12 holdout and T15 public (Table 1). It improves upon the decoder of Feng et al. (2024), both MoDAl variants (Chen & Chin, 2026), and BIT's end-to-end configurations and ensemble (Zhang et al., 2026).

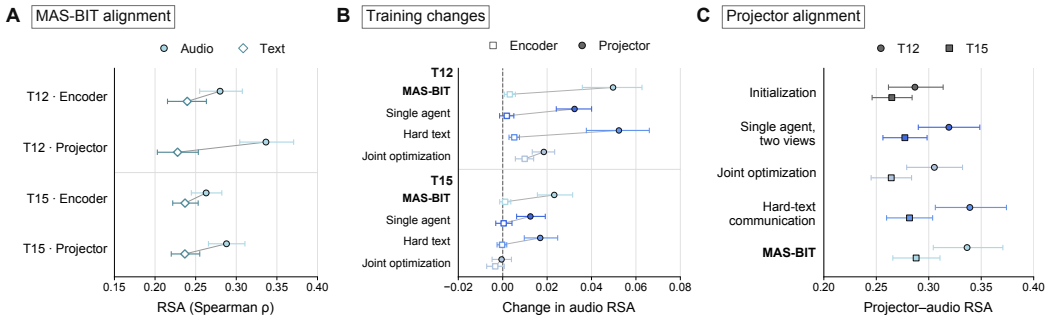


Figure 5: *Sentence-level neural alignment with Qwen audio and text representations.* **(A)** Encoder and projector RSA with audio and text references. **(B)** Change in audio-reference RSA from the shared initialization. **(C)** Projector–audio RSA across training variants. Error bars are pointwise 95% paired-bootstrap intervals; the audio reference encodes fixed-voice TTS of the reference transcripts.

5.6 Multi-agent staged training strengthens neural-to-audio alignment

We compare the sentence geometry of encoder states H and projected states Z_N with Qwen audio and text references using word-count-stratified representational similarity analysis (RSA) (Kriegeskorte et al., 2008). Representation extraction, standardization, dissimilarity construction, weighting, paired bootstrap, and length-control analyses are described in Appendix F.

MAS-BIT is more aligned with the audio than the text reference at both encoder and projector outputs for both participants (Figure 5A). At the projector, audio/text RSA is 0.337/0.228 for T12 and 0.288/0.237 for T15. Because the selected references differ in both modality and model capacity, this comparison should not be interpreted as an isolated modality effect.

Training-associated gains are concentrated at the projector output. Relative to the shared initialization, audio RSA increases by 0.0497 and 0.0232 at the projector for T12 and T15, versus 0.0032 and 0.0011 at the encoder (Figure 5B). At the projector, MAS-BIT reaches 0.337/0.288 on T12/T15, exceeding initialization (0.287/0.265), *Single agent, two views* (0.319/0.277), and *Joint optimization* (0.305/0.264) (Figure 5C). Hard-text communication yields similar RSA magnitudes with participant-dependent ordering.

6 Discussion

In this work, we introduce MAS-BIT, an end-to-end speech decoder that combines neural and phonetic experts through latent communication. Starting from released BIT encoders, the experts interpret two representations of the same recording, and an arbiter integrates their messages with direct neural evidence. The controls show that assigning these views to specialized agents improves upon a single decoder given both views, while latent messages and alternating updates further improve decoding. MAS-BIT achieves state-of-the-art end-to-end WER on the T12 holdout and T15 public subset, surpassing BIT's end-to-end ensemble (Zhang et al., 2026). The corresponding WERs are 8.418% and 4.878%, with 5.288% on the T15 private subset. Our cascade comparison starts from the same released encoder checkpoints: MAS-BIT achieves lower WER on both T15 subsets through 100 candidates, and approaches cascaded decoding on T12. Together, these results show how coordinated interpretation of neural evidence can improve end-to-end decoding within one jointly trained system.

MAS-BIT advances an end-to-end approach to restoring communication from attempted speech in people with paralysis. Its shared language-model backbone supports specialized decoding without storing a separate pretrained model for each role. Compared with the reproduced cascade, MAS-BIT reduces peak host RAM from 321.8 to 12.7 GiB on T12 and from 334.8 to 31.1 GiB on T15, lowering the memory burden for future edge deployment (Table 12). This combination of decoding accuracy and reduced host-memory demand supports more compact computing platforms for assistive communication.

AI USE STATEMENT

All research ideas come from the author(s). We used generative AI tools, including ChatGPT and Claude, for grammar checks, writing improvements, and text polishing. All research ideas, hypotheses, experimental designs, and scientific interpretations originated from the authors.

REPRODUCIBILITY STATEMENT

Detailed architecture, optimization, preprocessing, evaluation, and analysis procedures are provided in Sections 3–5 and Appendices A–F. Upon publication, we will release the source code, configurations, and model checkpoints to facilitate reproduction of the reported results.

REFERENCES

Tyler Benster, Guy Wilson, Reshef Elisha, Francis R Willett, and Shaul Druckmann. A cross-modal approach to silent speech with llm-enhanced recognition, 2024. URL https://arxiv.org/abs/2403.05583.

Tommaso Boccato, Michal Olak, and Matteo Ferrante. BrainWhisperer: Leveraging Large-Scale ASR Models for Neural Speech Decoding, 2026. arXiv:2603.13321v1.

Nicholas S. Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R. Willett, Erin M. Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R. Deo, Aparna Srinivasan, Eun Young Choi, Matthew F. Glasser, Leigh R. Hochberg, Jaimie M. Henderson, Kiarash Shahlaie, Sergey D. Stavisky, and David M. Brandman. An Accurate and Rapidly Calibrating Speech Neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024. ISSN 1533-4406. doi: 10.1056/nejmoa2314132. URL https://doi.org/10.1056/NEJMoa2314132.

Yuanhao Chen and Peter Chin. Modal: Self-supervised neural modality discovery via decorrelation for speech neuroprosthesis, 2026. URL https://arxiv.org/abs/2605.00025.

Avihu Dekel, Samuel Thomas, Takashi Fukada, and George Saon. NLE: Non-autoregressive LLM-based ASR by Transcript Editing, 2026. URL https://arxiv.org/abs/2603.08397.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Ebrahim Feghhi, Shreyas Kaasyap, Nima Hadidi, and Jonathan C. Kao. Time-masked transformers with lightweight test-time adaptation for neural speech decoding, 2025. URL https://arxiv.org/abs/2507.02800.

Ebrahim Feghhi, Junlin Hu, Nima Hadidi, and Jonathan C. Kao. Lightbeam: An accurate and memory-efficient ctc decoder for speech neuroprostheses, 2026. URL https://arxiv.org/abs/2603.14002.

Sheng Feng, Heyang Liu, Yu Wang, and Yanfeng Wang. Towards an End-to-End Framework for Invasive Brain Signal Decoding with Large Language Models. In *Interspeech 2024*, pp. 1495–1499, 2024. doi: 10.21437/Interspeech.2024-382.

Tobias Fiedler, Leon Hermann, Florian Müller, Sarel Cohen, Peter Chin, Tobias Friedrich, and Eilon Vaadia. Teaching Wav2Vec2 the language of the brain, 2025. URL https://arxiv.org/abs/2501.09459v1.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29.

Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf.

Zachery M. Fogg, Nicholas S. Card, Maitreyee Wairagkar, Aparna Srinivasan, Tyler Singer-Clark, Xianda Hou, Elizaveta Okorokova, Hamza Peracha, Carrina Iacobacci, Tiffany Brailow, Justin Jude, Hadar Levi-Aharoni, Trung Le, Domenick Mifsud, Pranav I. Deevi, Samuel R. Nason-Tomaszewski, Anna L. Pritchard, Yunnuo Zhang, Brice Richards, Payton Bechefsky, Leigh R. Hochberg, Ziv Williams, Kiarash Shahlaie, Nicholas AuYong, Daniel B. Rubin, Chethan Pandarinath, David M. Brandman, and Sergey D. Stavisky. A generalizable speech neuroprosthesis. *bioRxiv*, 2026. doi: 10.64898/2026.07.23.739430. URL https://www.biorxiv.org/content/early/2026/07/27/2026.07.23.739430.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, ICML '06, pp. 369–376. ACM Press, 2006. doi: 10.1145/1143844.1143891. URL http://dx.doi.org/10.1145/1143844.1143891.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR 2022*, April 2022. URL https://www.microsoft.com/en-us/research/publication/lora-low-rank-adaptation-of-large-language-models/.

Ke Hu, Tara N. Sainath, Ruoming Pang, and Rohit Prabhavalkar. Deliberation Model Based Two-Pass End-To-End Speech Recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7799–7803. IEEE, 2020. doi: 10.1109/icassp40776.2020.9053606. URL https://doi.org/10.1109/ICASSP40776.2020.9053606.

Owais M. Khanday, Jose A. Gonzalez-Lopez, Marc Ouellet, Alberto Galdón, and Gonzalo Olivares Granado. End-to-end intracortical speech decoding from neural activity. In *Odyssey 2026*, pp. 165–172, 2026. doi: 10.21437/Odyssey.2026-24. URL https://www.isca-archive.org/odyssey_2026/khanday26_odyssey.html.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, Volume 2 - 2008, 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008. URL https://www.frontiersin.org/journals/systems-neuroscience/articles/10.3389/neuro.06.004.2008.

Jaesong Lee and Shinji Watanabe. Intermediate loss regularization for CTC-based speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228. IEEE, June 2021. doi: 10.1109/ICASSP39728.2021.9414594. URL https://doi.org/10.1109/ICASSP39728.2021.9414594.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://mlanthology.org/iclr/2019/loshchilov2019iclr-decoupled/.

Rao Ma, Tongzhou Chen, Kartik Audhkhasi, and Bhuvana Ramabhadran. LegoSLM: Connecting LLM with speech encoder using CTC posteriors. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18171–18186, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.985. URL https://aclanthology.org/2025.findings-emnlp.985/.

Samuel R Nason, Alex K Vaskov, Matthew S Willsey, Elissa J Welle, Hyochan An, Philip P Vu, Autumn J Bullard, Chrono S Nu, Jonathan C Kao, Krishna V Shenoy, et al. A low-power band of neuronal spiking activity dominated by local single units improves the performance of brain–machine interfaces. *Nature biomedical engineering*, 4(10):973–983, 2020.

Michal Olak, Tommaso Boccato, and Matteo Ferrante. Decoding the decoder: Contextual sequence-to-sequence modeling for intracortical speech decoding, 2026. arXiv:2603.20246v1.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, pp. 2613–2617, 2019. doi: 10.21437/Interspeech.2019-2680.

Kyubyong Park and Jongseok Kim. g2pE. https://github.com/Kyubyong/g2p, 2019.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 51899–51928, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e444859b2a22df6b56af9381ad1e9480-Paper-Conference.pdf.

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51094–51116. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ramesh25a.html.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-ASR technical report, 2026. URL https://arxiv.org/abs/2601.21337.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5434be94e82c54327bb9dcaf7fca52b6-Paper-Conference.pdf.

Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620 (7976):1031–1036, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06377-x. URL https://doi.org/10.1038/s41586-023-06377-x.

Francis R. Willett, Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, Eli Shlizerman, Yue Chen, Xin Zheng, Tatsuo S. Okubo, Tyler Benster, Hyun Dong Lee, Maxwell Kounga, E. Kelly Buchanan, David Zoltowski, Scott W. Linderman, and Jaimie M. Henderson. Brain-to-text benchmark '24: Lessons learned, 2024. URL https://arxiv.org/abs/2412.17227.

Seonghyun Yoon, Donald T. Avansino, Sasidhar Madugula, Alisa D. Levin, Chaofei Fan, Benyamin Abramovich Krasa, Akansha Singh, Christina Vo, Nick V. Hahn, Nicholas S. Card, Zachery Fogg, Maitreyee Wairagkar, Samuel R. Nason-Tomaszewski, Brandon G. Jacques, Payton H. Bechefsky, Carrina Iacobacci, Darrel R. Deo, Leigh R. Hochberg, David M. Brandman, Sergey D. Stavisky, Nicholas Au Yong, Chethan Pandarinath, Jaimie M. Henderson, and Francis R. Willett. Neural decoding of speech using deep neural ensembles. *bioRxiv*, 2026. doi: 10.64898/2026.06.02.729705. URL https://www.biorxiv.org/content/10.64898/2026.06.02.729705v1. Preprint.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

Yizi Zhang, Linyang He, Chaofei Fan, Tingkai Liu, Han Yu, Trung Le, Jingyuan Li, Scott Linderman, Lea Duncker, Francis Willett, Nima Mesgarani, and Liam Paninski. A cross-species neural foundation model for end-to-end speech decoding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 30773–30797, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/34307cd20546fa1c6de90f303973ccf7-Paper-Conference.pdf.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. In *Forty-third International Conference on Machine Learning*, 2026.

A LIMITATIONS AND FUTURE DIRECTIONS

The encoder uses bidirectional attention and processes complete utterances, as in BIT (Zhang et al., 2026). A streaming version could combine causal neural encoding with expert messages updated as new activity arrives. Inference also includes expert generation, replay, and arbitration; reducing repeated language-model computation is an important direction for lowering latency. Another consideration is how much information each message needs to retain. Our validation analyses find a larger benefit from alternative tokens on T12, particularly when experts disagree, and a small aggregate effect on T15. This suggests investigating communication that adapts to the experts' agreement and uncertainty. The current models are trained separately for two participants with one language-model backbone. Training across more participants and evaluating additional backbones would clarify how the expert roles transfer across neural recordings and decoder families. Repeated training runs would quantify the variability of the smaller ablation gains. Finally, adapting the shared encoder and agents to recording drift and reduced supervision remains important for long-term speech BCI use.

B IMPLEMENTATION DETAILS

B.1 DATA AND INITIALIZATION

Section 4.1 describes dataset splits and checkpoint selection.

Input features are concatenated as threshold crossings followed by spike-band power. Each input bin spans 20 ms, giving $C = 256$ channels for T12 and $C = 512$ for T15. The encoder uses Gaussian smoothing, non-overlapping five-frame patches, and seven bidirectional Transformer blocks. T12 uses 128 electrodes, block-wise z-scoring, and subsequent day-level normalization estimated from that day's training trials. T15 uses all 256 electrodes and the normalized features supplied in the released HDF5 files. The maximum input lengths are 1,250 and 2,500 frames, respectively; all training utterances fit these limits. Transcriptions are lowercased, and punctuation other than apostrophes is removed. Phoneme targets are generated from the transcriptions with `g2p_en` (Park & Kim, 2019), with stress markers removed, spaces mapped to silence, and a final silence appended. The 41-class inventory contains 39 ARPAbet phones, silence, and blank.

The released BIT model supplies the neural encoder and its matching phoneme readout. MAS-BIT introduces a new linear neural-to-text projection, a phoneme embedding table, two message-role vectors, and three LoRA adapters on top of the pretrained Qwen text decoder. These are trained within one run beginning with the bridge stage. The neural encoder uses a width of 384, seven layers, six attention heads, a feedforward width of 512, GELU activations, and rotary positional embeddings (Su et al., 2024). Its participant-specific patch input layer applies LayerNorm, a linear projection from $5C$ to 384, and another LayerNorm. Gaussian smoothing has a standard deviation of two input bins (40 ms); embedding and Transformer dropout are 0.2 and 0.4. The text decoder has width 2048 and 28 layers. Each role's LoRA is applied to the query, key, value, output, gate, up, and down projections, with rank 8, scaling parameter 32 (a factor of $32/8$), and dropout 0.2.

Role-specific prompts. Text delimiters identify each role's continuous input, followed by an instruction to produce an English sentence. The neural expert's prompt names neural activity, the phonetic expert's prompt names phoneme probability features, and the arbiter's prompt names neural activity and both experts' transcript estimates. Figure 2A shows the complete prompts, with # marking the continuous-input insertion. Table 2 gives their exact text and the single-agent prompt used during training and inference. Hard-text communication and joint optimization retain the same three role prompts; the single-agent prompt names both input views. The delimiters and instructions use the pretrained tokenizer's existing vocabulary. For the single-agent control, v_N and v_P are learned view embeddings for the neural and phonetic inputs, respectively.

B.2 MESSAGE CONSTRUCTION

The continuous inputs to the neural expert, phonetic expert, and arbiter are Z_N, Z_P, and $[Z_N; M_N; M_P]$, respectively. The hard-text control constructs M_N and M_P from the discrete draft-token embeddings. The single-agent control concatenates Z_N and Z_P after adding learned

Table 2: *Prompts for MAS-BIT and the single-agent control.* The placeholder # is replaced by the continuous input in the middle column. Paired text delimiters and instruction text are shown verbatim.

Role	Input	Prompt
Neural expert	Z_N	`<|neural_activity|>#` `<|/neural_activity|>` *decode the above neural activity into an English sentence:*
Phonetic expert	Z_P	`<|phoneme_probability_features|>#` `<|/phoneme_probability_features|>` *decode the above phoneme probability features into an English sentence:*
Arbiter	$[Z_N; M_N; M_P]$	`<|neural_and_expert_messages|>#` `<|/neural_and_expert_messages|>` *decode the above neural activity and both agents' transcript estimates into an English sentence:*
Single agent, two views	$[Z_N + v_N;\ Z_P + v_P]$	`<|neural_and_phoneme_features|>#` `<|/neural_and_phoneme_features|>` *decode the above neural activity and phoneme probability features into an English sentence:*

view identifiers. Cross-entropy supervises answer tokens, including EOS; conditioning tokens and padding receive no labels. EOS token IDs are 151643 and 151645.

For latent-message replay, the retained draft includes the first EOS token; generated padding is excluded by its validity mask. Each expert's whole message is zeroed independently with probability 0.15, without rescaling the retained messages; subsequent elementwise embedding dropout has probability 0.1 and uses standard inverse-retention scaling. Role vectors are added after both operations.

B.3 Neural input augmentation

Every training example enters the augmentation pipeline. Time warping and masking follow principles used in SpecAugment (Park et al., 2019). First, we draw two interior cut points and remove duplicates to define temporal segments; we then resample each segment linearly by a factor drawn from $\mathcal{N}(1, 0.05^2)$ and clip it to $[0.9, 1.1]$. Trials shorter than nine frames retain their original time grid. A proposed warp is rejected if it exceeds the input limit or leaves fewer encoder patches than required by the target-phone count plus adjacent repeated phones. The remaining transformations still apply when warping is rejected. Second, we mask each electrode independently with probability 0.05, zeroing its threshold-crossing and spike-power features together across the trial. Third, raw temporal spans of 10–20 frames are sampled with a cumulative span-length budget of $\lfloor 0.1 T_{\text{aug}} \rfloor$, where T_{aug} is the post-warp sequence length; the last span is shortened to the remaining budget, and overlaps are allowed. Fourth, independent Gaussian noise with standard deviation 0.2 is added in normalized feature units. These operations precede the encoder's Gaussian smoothing.

Let $S_{\text{aug}} = \lceil T_{\text{aug}}/5 \rceil$ be the number of valid augmented patches. After the patch input projection, a contiguous span length is sampled uniformly from 1 to $\lfloor 0.2 S_{\text{aug}} \rfloor$ and placed uniformly among valid starting positions. Replace its patches with the frozen pretrained mask embedding, retaining sequence length and attention validity. No span is applied when $\lfloor 0.2 S_{\text{aug}} \rfloor = 0$. One augmentation realization is shared across the three roles for each example, including during the bridge stage. Evaluation retains deterministic smoothing and disables these stochastic corruptions.

B.4 Optimization and reproducibility

The schedule in Equation 6 corresponds to bridge epochs 1–5, followed by N epochs 6–10, P epochs 11–15, and A epochs 16–20 in the first cycle. The same five-epoch N/P/A blocks repeat in epochs 21–35, 36–50, and 51–65. The intermediate-CTC ramp advances during encoder-active training, beginning at epoch 6. All three cross-entropy terms have unit weight and use no label smoothing. Intermediate CTC uses the same output normalization and phoneme readout as the final state, with

Table 3: *Parameter groups updated in each stage.* The pretrained language-model backbone remains frozen throughout.

Stage	Updated parameters
Bridge	Neural projector, phoneme embedding table, message-role embeddings
N	Neural encoder, neural projector, N adapter
P	Phoneme readout, phoneme embedding table, P adapter
A	Message-role embeddings, A adapter

total weight 0.05 across layers 3 and 5 in addition to final-layer weight 0.3. The CTC implementation uses blank index zero and suppresses infinite losses for infeasible alignments.

We use AdamW with an effective batch size of 32, weight decay 10^{-5}, and gradient-norm clipping of the active parameters at 1. Peak learning rates are 2×10^{-5} for the neural encoder and phoneme readout and 10^{-4} for the interfaces and adapters. Learning rates follow a separate warmup–cosine curve within each five-epoch block. For a block containing K optimizer updates, let $w = \max(1, \mathrm{round}(0.05K))$ and let t count completed updates in that block. An active group's learning rate is

$$\eta_g(t) = \eta_g^{\mathrm{peak}}\, \mu_c\, s(t), \qquad s(t) = \begin{cases} 0.1 + 0.9t/w, & t < w, \\ 0.1 + 0.45\left[1 + \cos\left(\pi \frac{t-w}{K-w}\right)\right], & t \geq w, \end{cases} \tag{7}$$

where the cycle peak multipliers μ_c are 1, 0.5, 0.25, and 0.125; the bridge uses multiplier 1. The block-local warmup restarts at each stage transition, while optimizer moments are retained throughout the run. The learning-rate schedule is fixed and is not reduced in response to validation plateaus.

We train on four data-parallel ranks with an effective batch size of 32. Each rank accumulates two microbatches of four utterances for T12 and four microbatches of two utterances for T15. Within each rank, cross-entropy sums are divided by that window's actual target-token count, including EOS, and CTC sums by its target-phone count. The update averages the four resulting rank-local objectives. The last partial window uses its actual counts, and padding or inactive duplicate slots contribute zero loss. Biases and one-dimensional parameters are excluded from weight decay. Training uses BF16 autocast and random seed 42. Validation generation and loss evaluation both use batch size one.

B.5 Checkpoint selection and evaluation

Each epoch is evaluated by greedy sentence generation on all provided labeled validation utterances. Checkpoint selection begins at epoch 20, after the first complete specialist–arbiter cycle in the staged model. Earlier epochs are evaluated but are ineligible for selection. We retain the eligible checkpoint with the smallest total word-edit count and stop after 15 epochs without a strict improvement, or at the maximum of 65 epochs. The reference-word denominator is fixed, so minimizing edit count is equivalent to minimizing corpus validation-set WER. Ties retain the earlier checkpoint; stage transitions do not reset patience, and training may stop within a block.

The T15 training set includes the three released "do nothing" control trials. All declared rows are retained in training and evaluation. The labeled validation sets support checkpoint selection, analysis of expert predictions, and the input-masking study. Benchmark comparisons and the three trained controls report holdout WER; T15 public and private scores are kept separate. For final holdout decoding, the selected checkpoint uses FP32 and a batch size of one. Expert and arbiter generation are greedy, with fixed caps of 24 new tokens for T12 and 29 for T15, and terminate earlier at an end-of-sequence token. Reference transcriptions are excluded from generation, stopping, and candidate selection.

B.6 Implementation of the ablation controls

All controls share the data splits, augmentation, effective batch size 32, and selection rule. Corresponding components use common initial weights; subsequent independent training can change their representations.

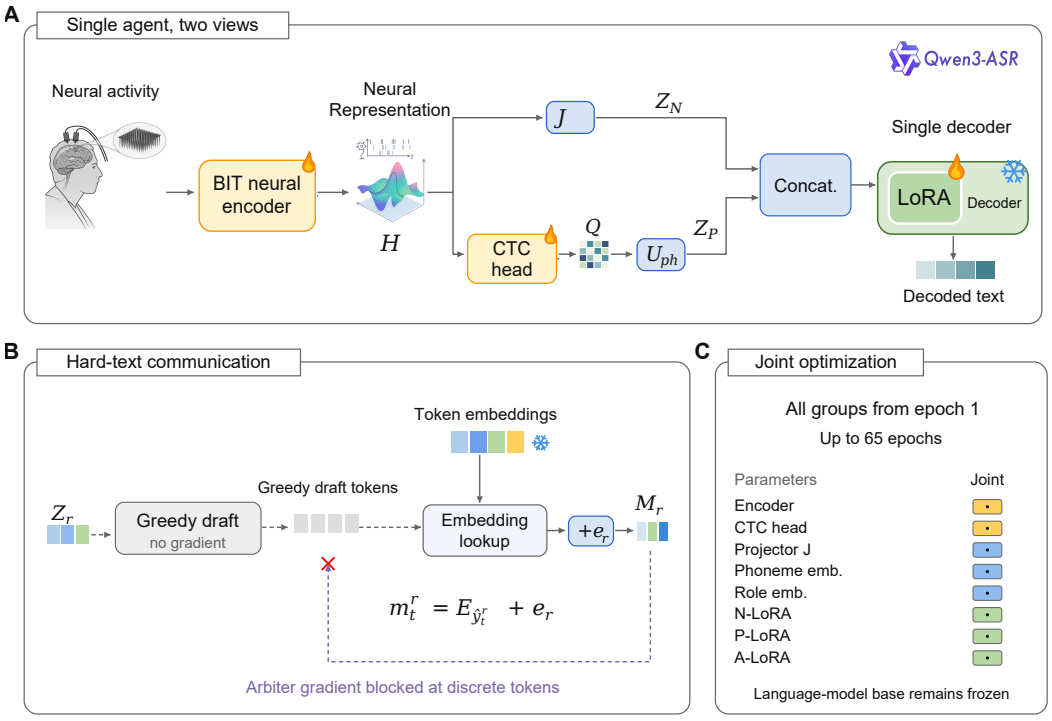


Figure 6: *Three controls alter how neural evidence is decoded, communicated, and optimized.* The diagrams modify the corresponding components of Figure 1. **(A)** A single decoder receives concatenated neural and phonetic views, without expert messages. Learned view embeddings are added before concatenation. **(B)** Each expert $r \in \{N, P\}$ converts its input Z_r into greedy draft tokens $\hat{y}_t^r$. Adding the learned message-role embedding e_r to the frozen token embedding $E_{\hat{y}_t^r}$ gives m_t^r; the message sequence M_r is sent to the arbiter. Gray dashed arrows mark the detached draft path; the purple path stops at the red cross, blocking arbitration gradients to the expert through this message. Expert sentence supervision and direct neural input remain. **(C)** Joint optimization updates all eight adaptation groups from epoch 1, within the same 65-epoch maximum. Flames denote adaptable components and snowflakes denote frozen language-model weights or token embeddings; the green decoder modules share one backbone.

Single agent, two views. The input concatenates Z_N and Z_P, with learned view identifiers. Its prompt requests an English transcription from neural activity and phoneme probability features. The single adapter starts from the untrained arbiter adapter and is active in all 60 scheduled post-bridge epochs. Its block schedule matches the number of scheduled encoder, readout, and input-interface update epochs in MAS-BIT. The decoder uses one role adapter and a single sentence loss.

Hard-text communication. Each message uses the frozen token embeddings of the expert's greedy draft in place of probability-weighted embeddings. The direct neural input to the arbiter and message-role embeddings are retained. The discrete message blocks arbitration gradients into the sending expert, while the experts retain their own sentence supervision. Consequently, this control changes both the retained token information and the gradient path.

Joint optimization. All adaptation groups, including the encoder, update from epoch 1. The frozen language-model base, thirteen five-epoch learning-rate windows, and reference intermediate-CTC weighting calendar are preserved. The maximum number of epochs is matched, but joint optimization gives individual groups more active update epochs.

C ANALYSIS OF EXPERT PREDICTIONS

We decode all 880 T12 and 1426 T15 validation utterances using the selected checkpoints and the same greedy decoding settings as for the holdout sets. The expert-state, word-level, and message-intervention analyses use the same expert drafts and complete-input predictions. All three roles use the same checkpoint and recording; references are read only after generation. A draft is correct when it matches the target under the common evaluation normalization. Table 4 partitions every utterance into four expert-draft states and records both correct and incorrect arbiter outputs. The analysis covers the full checkpoint-selection sets.

Table 4: *Arbitration gains and failures across expert-draft states.* Correct and wrong refer to arbiter exact match after normalization. The four states cover the complete checkpoint-selection sets.

	T12			T15		
Expert drafts	**Correct**	**Wrong**	**Total**	**Correct**	**Wrong**	**Total**
Both experts correct	350	1	351	861	5	866
Only neural expert correct	91	13	104	143	10	153
Only phonetic expert correct	36	4	40	51	7	58
Both experts wrong	59	326	385	52	297	349
Total	536	344	880	1107	319	1426

Gains and losses relative to complete-draft selection. For each utterance, an offline, reference-aware complete-draft oracle selects the neural or phonetic draft with fewer word edits, with the neural draft retained on ties. Let e_i^N, e_i^P, and e_i^A denote the word-edit distances of the two experts and arbiter to the reference, and let $e_i^* = \min(e_i^N, e_i^P)$. We count edits removed and introduced by arbitration as

$$G = \sum_i \max(e_i^* - e_i^A, 0),$$

$$H = \sum_i \max(e_i^A - e_i^*, 0).$$

Thus $G - H$ is the net reduction in word edits relative to the oracle. Table 5 also counts utterances with fewer, equal, or more edits, retaining every declared row. The exact-match gains are recoveries when both drafts are wrong; exact-match losses include every utterance with an available correct draft that the arbiter fails to preserve.

Table 5: *Arbitration gains and losses relative to a complete-draft oracle.* All entries are counts on the checkpoint-selection sets. The oracle selects one complete expert draft using the reference. Positive net changes indicate more exact matches or fewer word edits.

Quantity	**T12**	**T15**
Oracle exact matches	495	1077
Arbiter exact matches	536	1107
Correct utterances gained	59	52
Correct utterances lost	18	22
Net exact matches	+41	+30
Reference words	5496	9445
Oracle word edits	689	579
Arbiter word edits	624	554
Edits removed (G)	125	96
Edits introduced (H)	60	71
Net edits removed ($G - H$)	+65	+25
Utterances with fewer edits	106	76
Utterances with equal edits	722	1291
Utterances with more edits	52	59

C.1 ARBITRATION BEYOND ALIGNED DRAFT RECOMBINATION

The complete-draft oracle selects one expert sentence. We construct a stronger word-combination oracle by aligning the neural and phonetic drafts before reading the reference. We retain all minimum-cost word alignments using unit costs for substitutions, deletions, and insertions. At each alignment edge, the candidate graph offers either the expert's word or an empty choice where that alignment has a gap. The graph preserves word order and contains both original drafts. For its set of candidate sentences $\mathcal{L}_i$, the oracle edit count is

$$e_i^{\text{comb}} = \min_{z \in \mathcal{L}_i} \text{ED}(z, y_i).$$

We compute this minimum by dynamic programming over the graph and reference positions. The reference is used for offline scoring after graph construction. We also evaluate a more permissive graph that allows an empty choice at every edge, including positions where both drafts contain a word.

Table 6: *Complete-draft selection, aligned word combination, and arbitration.* Edit counts and exact matches cover all 880/1426 T12/T15 checkpoint-selection utterances, containing 5496/9445 reference words. Both oracles use the reference for scoring.

	T12		T15	
Output	**Word edits**	**Exact matches**	**Word edits**	**Exact matches**
Neural expert	821	455	701	1019
Phonetic expert	1023	391	1071	924
Complete-draft oracle	689	495	579	1077
Word-combination oracle	657	504	547	1085
+ Optional word deletion	634	505	537	1088
Arbiter	624	536	554	1107

With draft-supported gaps, the word-combination oracle incurs 657/547 edits on T12/T15. Allowing optional word deletion reduces these counts to 634/537, compared with 624/554 for the arbiter (Table 6). The arbiter has fewer word edits than either word-combination oracle on T12 and more on T15, while producing more exact transcriptions on both participants. Relative to this more permissive oracle, the arbiter improves, ties, or worsens the edit count on 85/726/69 T12 utterances and 64/1283/79 T15 utterances. Of all correct arbiter transcriptions, 477/1055 exactly match a complete expert draft. Among the 59/52 exact recoveries with both drafts wrong, 6/5 targets are available through aligned word combination, while 53/47 lie outside the graph (Figure 3). This classification is unchanged by optional word deletion. Of these outside-graph recoveries, 53/47 contain a correct word absent from both drafts. Using one deterministic minimum-cost alignment instead of all optimal alignments gives 658/549 oracle edits; the corresponding all-alignment counts are 657/547.

Word-level complementarity. We align each draft and the arbiter output to the reference with the same unit edit costs, resolving ties by diagonal, deletion, then insertion. At each reference word, we record which experts are correct and whether arbitration corrects or introduces a substitution or deletion relative to the better complete draft. At each reference boundary, we count changes in inserted words. Figure 7 decomposes these changes by expert support. Corrected substitutions/deletions/insertions total 128/2/8 on T12 and 100/2/4 on T15; introduced edits total 63/2/8 and 69/2/10, respectively. The phonetic expert is uniquely correct at 148/147 reference-word positions on T12/T15. The arbiter is correct at 126/121 of these positions.

Substitutions account for most expert disagreements (Table 7). We count target-word occurrences in the 8,800 T12 or 8,072 T15 training transcripts and group them as unseen, 1–5, 6–20, or more than 20 occurrences. We use NLTK's contextual part-of-speech tags to group words: closed-class tags and a fixed pronoun/auxiliary list define function words, while the remaining noun, verb, adjective, and adverb tags define content words. For substitutions, we use stress-stripped CMU pronunciations and take the minimum, over pronunciation pairs, of edit distance divided by the longer pronunciation length. Words absent from the dictionary retain missing phoneme-distance values.

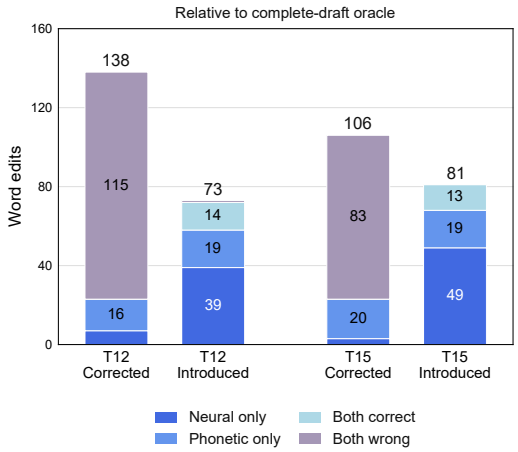


Figure 7: *Word edits corrected and introduced by arbitration.* Changes are measured relative to the better complete draft. Colors indicate which experts are correct at the aligned reference position; insertion changes use the absence of extra words at a reference boundary. A sentence can contribute both corrections and new errors. All counts use the full validation sets.

Table 7: *Word positions correct in exactly one expert draft.* Columns identify the correct expert; substitution/deletion counts describe the other expert's error. Training frequency is based only on the corresponding participant's training transcripts. Phoneme distance is averaged over dictionary-covered substitutions, with coverage shown explicitly.

	T12		T15	
Quantity	**Neural**	**Phonetic**	**Neural**	**Phonetic**
Word positions	334	148	446	147
Arbiter correct	295	126	401	121
Other expert: substitution	326	146	431	141
Other expert: deletion	8	2	15	6
Content words	252	113	350	122
Function words	81	31	86	22
Other tags	1	4	10	3
Unseen in training	45	23	104	39
Training count 1–5	99	37	121	43
Training count 6–20	61	36	76	27
Training count > 20	129	52	145	38
Mean phoneme distance	0.586	0.563	0.601	0.499
Dictionary coverage	311/326	145/146	420/431	136/141

C.2 EXPERT MESSAGES COMPLEMENT DIRECT NEURAL CONDITIONING

We freeze the selected checkpoints and decode the full validation sets under five input conditions (Table 8). For each utterance, the experts generate their drafts and latent messages once; the arbiter then decodes with the complete input, each message masked separately, both messages masked, or the direct neural input zeroed. Message masking follows training dropout: we zero the message content before adding its role embedding, preserving sequence lengths, positions, and validity masks. Zeroing the projected neural input leaves both expert messages unchanged.

Table 8: *Arbitration is most accurate with both expert messages and direct neural conditioning.* Corpus validation-set WER (%) is reported on 880 T12 and 1,426 T15 utterances. Recovered counts of exact transcriptions among the same 385 T12 and 349 T15 utterances with both expert drafts wrong. All conditions use the same checkpoints and holdout decoding settings; the complete-input predictions match those used in the preceding analyses.

	T12		T15	
Arbiter input	**Validation WER**	**Recovered**	**Validation WER**	**Recovered**
Complete input	11.354	59	5.866	52
Neural message masked	13.119	48	6.903	48
Phonetic message masked	12.136	57	6.215	55
Both messages masked	16.157	28	8.767	27
Direct neural input zeroed	61.554	15	13.287	10

The complete input gives the lowest validation-set WER for both participants. Masking both messages increases validation-set WER by 4.803 percentage points on T12 and 2.901 on T15. Masking both messages reduces these recoveries from 59 to 28 on T12 and from 52 to 27 on T15; zeroing direct neural conditioning reduces them to 15 and 10. On T15, masking the phonetic message increases validation-set WER but raises the recovery count from 52 to 55, indicating that its benefit varies across utterances. Direct-input zeroing extends beyond the message-dropout conditions used in training, so its effect measures input dependence rather than a causal contribution share.

C.3 TOKEN ALTERNATIVES IN FROZEN EXPERT MESSAGES

We vary expert messages while fixing the selected model weights, greedy expert drafts, and direct neural input to the arbiter. For each utterance, we cache the neural representations and the top-16 token logits from replay along each expert's own draft. We retain the top $k \in \{1, 2, 4, 8, 16\}$ candidates at each message position, renormalize their probabilities at $\tau = 1$, and construct the weighted embeddings in Equation 3. Both experts use the same k; message lengths, masks, and learned role embeddings remain fixed.

To test the correspondence between alternative tokens and their probabilities, we retain the leading token and its probability and randomly permute the probabilities assigned to ranks 2–16. This preserves the candidate set and probability entropy at each position. We use one fixed permutation per expert and position, with random seed 42, and reuse it for a second condition that matches each perturbed content vector's norm to the original before adding the role embedding. We also vary the message temperature over $\tau \in \{0.5, 1, 2\}$ at $k = 16$. The arbiter uses the same greedy decoding rule throughout.

Figure 8 shows the changes in corpus validation-set WER relative to the original top-16 messages. Tables 9 and 10 report the absolute WER on the complete provided validation sets and within each expert-draft group. Expert agreement, disagreement, and correctness are determined from the original fixed drafts under the common text normalization. These comparisons measure message sensitivity at fixed weights; the hard-text control in Appendix B.6 instead trains with a discrete message interface.

On T15, top-1 messages lose 18 previously correct utterances and recover 14, while the net change is one additional word edit. On T12, probability shuffling loses five exact matches and recovers seven, while adding five word edits. Thus small aggregate changes can include corrections and new errors on different utterances. Norm matching retains these small validation-set WER changes: 0.109 and 0.095 percentage points on T12 and T15, respectively.

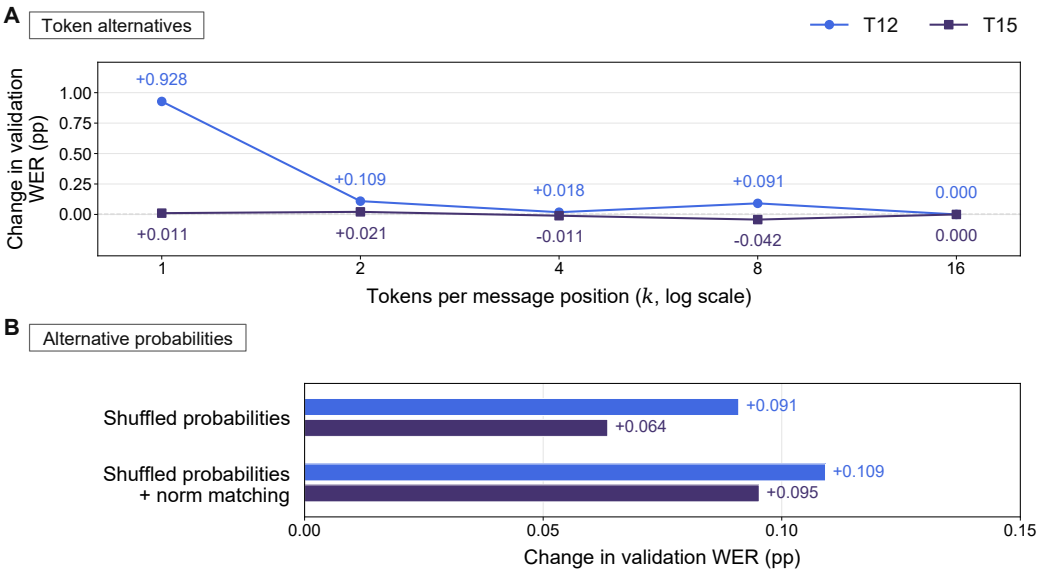


Figure 8: *Token alternatives improve T12 decoding; probability shuffling has small effects on both participants.* Changes in corpus validation WER are relative to top-16 messages at $\tau = 1$ on all 880 T12 and 1,426 T15 validation utterances. Positive values indicate higher WER. **(A)** Token support varies from 1 to 16 on a logarithmic axis. **(B)** Probabilities are shuffled among ranks 2–16 while preserving the leading token and its probability; the second condition also restores the original message content norms. The horizontal scale spans 0–0.15 percentage points. Values are labeled to three decimals. Model weights, expert drafts, direct neural input, and greedy arbiter decoding remain fixed.

Table 9: *Message interventions with fixed model weights and expert drafts.* Corpus validation-set WER (%) and word edits are reported on all 880 T12 and 1,426 T15 provided validation utterances. The reference-word denominators are 5,496 and 9,445, respectively. Top-1 changes the messages of the trained MAS-BIT model; it is distinct from the separately trained hard-text control.

	T12		T15	
Messages	**Edits**	**Validation WER**	**Edits**	**Validation WER**
Top-16 ($\tau = 1$)	624	11.354	554	5.866
Top-1	675	12.282	555	5.876
Top-2	630	11.463	556	5.887
Top-4	625	11.372	553	5.855
Top-8	629	11.445	550	5.823
Shuffled probabilities	629	11.445	560	5.929
Shuffled probabilities, norm matched	630	11.463	563	5.961
Top-16 ($\tau = 0.5$)	645	11.736	554	5.866
Top-16 ($\tau = 2$)	632	11.499	555	5.876

Table 10: *Message sensitivity across fixed expert-draft groups.* Values are corpus validation-set WER (%). Groups are defined before intervention. Agreement and disagreement partition each validation set; both-wrong utterances overlap these groups. Shuffled permutes alternative probabilities; matched additionally restores the original content norms.

Participant	**Expert drafts**	**Rows**	**Words**	**Top-16**	**Top-1**	**Shuffled**	**Matched**
T12	Agree	442	2675	3.925	4.037	4.000	4.000
T12	Disagree	438	2821	18.398	20.099	18.504	18.540
T12	Both wrong	385	2491	24.287	26.014	24.488	24.528
T15	Agree	934	6017	1.213	1.180	1.197	1.197
T15	Disagree	492	3428	14.032	14.119	14.236	14.323
T15	Both wrong	349	2439	21.771	22.099	22.140	22.263

D REPRODUCED CASCADE

Neural encoder and language models. We reproduce cascaded decoding with the released BIT encoder for each participant (Zhang et al., 2026). MAS-BIT is initialized from the same encoder.

Table 11: *Hyperparameters used for cascaded decoding.*

Parameter	Value
Maximum 5-gram hypotheses	100
Pruning beam / lattice beam	17 / 8
Maximum / minimum active states	7,000 / 200
Acoustic score scale	0.325
Blank penalty / blank-skip threshold	90 / 1
OPT rescoring weight (α)	0.55
Length penalty	0

The phoneme logits are decoded using the benchmark's OpenWebText2 5-gram weighted finite-state transducer (WFST). The lattice is rescored by replacing the pruned 5-gram contribution with the unpruned model before extracting up to 100 hypotheses. This candidate limit is fixed across all experiments; K controls only the number of ranked hypotheses passed to OPT-6.7B (Zhang et al., 2022). If fewer than K hypotheses are available, all are retained. Table 11 summarizes the decoder parameters.

For each K, the output maximizes $s(h) = 0.325\, s_{\mathrm{acoustic}}(h) + 0.45\, s_{5\mathrm{gram}}(h) + 0.55\, s_{\mathrm{OPT}}(h)$, where the OPT score is the sum of token log-probabilities without length normalization. The first candidate is retained on a tie. OPT scores every retained candidate, including the single candidate at $K = 1$. No reference labels are used to generate or rank hypotheses.

We retain the candidate sentences and their OPT scores and apply the same selection rule at each limit. The cached scores reproduce the transcriptions from separate decoding at the original seven limits. We specify all additional limits before obtaining their holdout scores. For each limit, WER is evaluated on every holdout sentence; the public and private T15 scores refer to distinct hidden subsets. Neural decoding and OPT scoring use FP32 without TF32 and process one utterance or candidate sentence at a time. Figure 4B shows the effect of the candidate limit on WER. Straight segments connect the measured settings; no smoothed curve is fitted.

Peak host RAM. We compare peak host RAM in complete inference runs, including model and data loading, decoding, and audits. The four measurements use a single NVIDIA A40 GPU and 16 cores of an AMD EPYC 7662 CPU, with FP32 computation and batch size 1. Each run loads one participant's model and completes all 1,200 T12 or 1,450 T15 holdout utterances with one decoder. The reproduced cascade uses $K = 100$, including WFST decoding, unpruned 5-gram rescoring, and OPT rescoring. We sample host resident memory once per second across the decoding process and its child processes, and report the peak.

Table 12: *Peak host RAM for T12 and T15 inference.* Each value summarizes a complete run with one participant and one decoder; units are GiB.

Method	T12	T15
Reproduced cascade ($K = 100$) (Zhang et al., 2026)	321.8	334.8
MAS-BIT (ours)	**12.7**	**31.1**

MAS-BIT uses 12.7 and 31.1 GiB on T12 and T15, compared with 321.8 and 334.8 GiB for the reproduced cascade. The lower peak host RAM reduces the memory burden of the decoding pipeline for future edge deployment.

E BENCHMARK AND CONTROL RESULTS

Tables 15 and 16 give the exact holdout results summarized in Figures 4B and 2B–C, respectively. Table 1 reports published benchmark comparisons.

Published end-to-end protocols. The Feng and BIT scores follow the benchmark tables of Zhang et al. (2026); BIT uses nucleus sampling, while MAS-BIT uses greedy decoding. MoDAl-1 uses an area 6v encoder, whereas MoDAl-Full combines three neural encoders and includes area 44 signals (Chen & Chin, 2026). Their 18.8% and 17.7% results come from the 1,200-sentence competition holdout, with 95% confidence intervals across five seeds; results on the 880-sentence model-selection set are listed separately in Table 13. Both configurations use CTC pretraining followed by direct autoregressive text decoding. BIT End-to-End + Ensemble combines independently trained decoders through an LLM merger and is retained as a separate ensemble reference. The T12 system merges ten models with a fine-tuned GPT-3.5, while the T15 system merges 29 models with a fine-tuned GPT-4; both mergers receive sentence hypotheses and phoneme outputs (Zhang et al., 2026).

Additional published end-to-end variants. The RNN, BIT-TFS, and BIT-Human end-to-end configurations in Table 1 vary the encoder within BIT (Zhang et al., 2026). We retain the one-decimal T12 and T15 competition-holdout WER values printed in their Figure 2A. The RNN configuration pairs its encoder with an end-to-end LLM decoder. BrainWhisperer has both a direct Whisper decoding path and a separate phoneme/WFST path (Boccato et al., 2026); Table 1 includes its direct-decoding competition results. BrainWhisperer-cross jointly trains on the Card, Willett, and Kunz datasets.

Decoding on other evaluation sets. Table 13 broadens the coverage of neural-to-text decoders while retaining each study's evaluation set and inference procedure. These values are reported separately from the matched holdout comparison because the partitions, training data, and selection procedures differ. Fiedler et al. (2025) reserve the first block of each training day for a custom 880-sentence test, leaving 7,920 training sentences; the original labeled test serves as validation. Both listed models are fully fine-tuned; we report the best run in each setting from the WER column without external language-model decoding.

Olak et al. (2026) exclude competition holdout data and report native greedy and beam-search decoding from a Transformer with a BART word decoder. MFCC targets provide auxiliary supervision during training rather than serving as an additional input at inference. Their paper calls the 880-sentence labeled data both test and validation, so we retain the neutral labeled-set designation. The internal phoneme selectors score word-decoder candidates using neural-conditioned phoneme likelihood or agreement with predicted phonemes after grapheme-to-phoneme conversion (G2P). These configurations use no independent language-model score and are listed separately from native word decoding; candidates are generated using beam search and nucleus sampling.

BrainWhisperer's Card Test results are distinct from those in the competition column; the Test size and split identifiers are not specified (Boccato et al., 2026). Its cross-dataset model is a jointly trained decoder, not an ensemble. For Feng et al. (2024), we distinguish the full 880-sentence test from its 720-sentence vocal subset. The additional MoDAl values are computed on the 880-sentence set used to select the models and are grouped accordingly (Chen & Chin, 2026).

Other decoding protocols. Khanday et al. (2026) report validation CER without WER, so their character-CTC model has no numerical entry here. The multi-user system of Fogg et al. (2026) and the neural ensembles of Benster et al. (2024); Yoon et al. (2026) use phoneme-to-word language-model decoding and remain cascaded systems. LightBeam uses lexicon constraints and external language-model fusion (Feghhi et al., 2026); removing a WFST does not make its deployment end-to-end under our definition.

Table 13: *Reported neural-to-text decoding on other evaluation sets.* Values are WER (%) as reported, with no cross-set ranking. Fiedler values are best runs with full fine-tuning. Olak values are mean ± SD over five seeds; MoDAl values are mean ± 95% CI over five seeds. None of these rows is an official hidden-holdout result.

Model	Evaluation set	Inference	WER (%)
Direct decoding			
Wav2Vec2Conformer (Fiedler et al., 2025)	T12 custom test (880)	Character CTC; no external LM	26.67
Wav2Vec2 (Fiedler et al., 2025)	T12 custom test (880)	Character CTC; no external LM	30.97
Transformer + MFCC + BART (Olak et al., 2026)	T12 labeled set (880)	Native beam search	24.8 ± 0.4
Transformer + MFCC + BART (Olak et al., 2026)	T12 labeled set (880)	Greedy	25.6 ± 0.2
BrainWhisperer (Boccato et al., 2026)	T15 Card Test	Native Whisper decoder	9.8
BrainWhisperer-cross (Boccato et al., 2026)	T15 Card Test	Native Whisper decoder	8.9
GRU-BPE + Llama2-7B (Feng et al., 2024)	T12 labeled test (880; all)	Autoregressive	28.5
GRU-Phone + Llama2-7B (Feng et al., 2024)	T12 labeled test (880; all)	Autoregressive	27.3
BGRU-Phone + Llama2-7B (Feng et al., 2024)	T12 labeled test (880; all)	Autoregressive	26.3
BGRU-Phone + Llama2-7B (Feng et al., 2024)	T12 vocal subset (720)	Autoregressive	26.2
Word decoding with internal phoneme selection			
Transformer + MFCC + BART (Olak et al., 2026)	T12 labeled set (880)	Phoneme-head likelihood; G2P	22.6 ± 0.2
Transformer + MFCC + BART (Olak et al., 2026)	T12 labeled set (880)	Predicted-phoneme agreement; G2P	23.2 ± 0.4
Labeled evaluation used for model selection			
MoDAl-1 (Chen & Chin, 2026)	T12 selection set (880)	Autoregressive	22.4 ± 0.2
MoDAl-Full (Chen & Chin, 2026)	T12 selection set (880)	Autoregressive	21.6 ± 0.1
MoDAl, two area-6v encoders (Chen & Chin, 2026)	T12 selection set (880)	Autoregressive	22.5 ± 0.3

Table 14: *Published cascaded references.* Values are corpus holdout WER (%) reported in the benchmark tables of Zhang et al. (2026). These systems use external language-model decoding and are not single-model end-to-end baselines. The published BIT cascade uses $K = 100$; T15 private scores were not reported.

Method	**T12 holdout**	**T15 public**	**T15 private**
RNN baseline (Willett et al., 2023; Card et al., 2024)	9.76	6.67	–
Time-masked Transformer (Feghhi et al., 2025)	7.98	–	–
Published BIT cascade (Zhang et al., 2026)	6.35	4.06	–

Table 15: *Candidate-sentence counts and holdout decoding.* Values are WER (%); T15 public and private scores refer to separate subsets. K is the candidate-sentence limit in our reproduced cascade. Blue shading marks MAS-BIT.

		T12	T15	
Method	K	**Holdout**	**Public**	**Private**
Reproduced cascade + OPT (Zhang et al., 2026)	100	7.877	5.969	6.026
	75	7.877	5.939	6.026
	50	7.931	5.939	6.057
	37	7.931	5.939	6.073
	25	7.917	5.939	6.120
	18	7.931	5.969	6.089
	12	7.958	6.030	6.136
	9	7.971	6.030	6.183
	6	8.012	6.090	6.214
	4	8.120	6.181	6.293
	3	8.256	6.090	6.387
	2	8.459	6.424	6.450
	1	9.298	7.272	7.062
MAS-BIT (ours)	–	**8.418**	**4.878**	**5.288**

Table 16: *Holdout performance of MAS-BIT and its ablations.* Values are WER (%) on the same hidden holdouts. The three controls use the shared backbone, data, and checkpoint-selection rule.

	T12	T15	
Method	**Holdout**	**Public**	**Private**
MAS-BIT	8.418	4.878	5.288
Single agent, two views	11.639	7.515	7.438
Hard-text communication	8.621	5.787	5.947
Joint optimization	9.054	5.363	5.634

F Representational Similarity Analysis

Representations and analysis sets. We analyze the same selected checkpoints as the decoding comparisons, together with their shared initialization, which contains the pretrained BIT encoder and newly initialized interfaces. For each of the 880 T12 and 1,426 T15 provided validation utterances, we average encoder states over valid time positions and apply the checkpoint's linear projector to obtain the projected sentence representation. For a linear projector, projecting the mean equals averaging the projected states. Audio reference vectors are mean-pooled Qwen3-ASR-0.6B audio-tower outputs on fixed-voice TTS of the reference transcripts; text vectors are mean-pooled final-layer Qwen3-ASR-1.7B text-decoder states on the transcripts. Thus the analysis compares sentence geometry in the selected neural and reference spaces, with reference modality and model capacity both varying.

Word-count-stratified RSA. For each representation, we z-score each feature across the full participant-specific sentence set and form a representational dissimilarity matrix using one minus Pearson correlation between sentence vectors. We retain groups of sentences with identical reference word counts and at least 20 utterances. Within each group g, we compute the Spearman correlation ρ_g between the distinct upper-triangle entries of the neural and reference matrices. The reported score is

$$\rho_{\mathrm{RSA}} = \frac{\sum_g \binom{n_g}{2} \rho_g}{\sum_g \binom{n_g}{2}}, \tag{8}$$

where n_g is the original number of sentences in group g. T12 contributes 848 sentences and 48,531 within-group pairs (2–10 words); T15 contributes 1,386 sentences and 134,018 pairs (3–11 words). The analysis unit is a sentence, with word count used for stratification.

Paired uncertainty estimates. We draw 2,000 bootstrap samples with replacement within each retained word-count group, sharing the draws across all models, output stages, and references within a participant (seed 0). Repeated sentences retain their multiplicities: a pair of distinct original sentences i, j receives weight $c_i c_j$, where c_i is the number of copies of sentence i in the draw. Pairs between copies of the same original sentence are excluded, and weighted midranks preserve the repeated-pair distribution when computing Spearman correlations. Full-data standardization and original group weights remain fixed. We take the 2.5th and 97.5th percentiles of the bootstrap scores or paired score differences. These pointwise intervals quantify sentence-sampling uncertainty conditional on the selected checkpoints, rather than variability across training runs.

Sensitivity to length adjustment. We additionally compute ungrouped RSA and partial RSA that controls for ranked absolute differences in reference word count between sentence pairs. For partial RSA, we regress each ranked dissimilarity vector on the ranked word-count-difference vector with an intercept, then correlate the residuals. MAS-BIT's projector–audio RSA exceeds initialization, the single-agent control, and joint optimization under all three estimators in both participants. The audio-over-text ordering at the projector also holds under all three estimators. At the T15 encoder, however, the ordering reverses under ungrouped and partial RSA (Table 17); the finding at both output stages in Section 5.6 is specific to the word-count-stratified comparison.

Table 17: *MAS-BIT audio and text RSA under three length treatments.* Each entry is audio / text. All estimators use the same cached sentence representations and full-data feature standardization. Grouped RSA retains the eligible word-count groups; the other estimators use all provided validation sentences.

Participant	Output	Word-count groups	Ungrouped	Partial
T12	Encoder	0.280 / 0.240	0.283 / 0.267	0.271 / 0.259
T12	Projector	0.337 / 0.228	0.318 / 0.258	0.307 / 0.249
T15	Encoder	0.263 / 0.237	0.240 / 0.243	0.233 / 0.239
T15	Projector	0.288 / 0.237	0.257 / 0.244	0.249 / 0.239