BRIDGE-DUPLEX: BRIDGING SEMANTIC AND GENERATIVE STREAMS FOR FULL-DUPLEX SPEECH-TO-SPEECH DIALOGUE

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex speech-to-speech (S2S) dialogue requires dialogue-level semantic processing to remain coordinated with continuous frame-wise generation as conversational evidence evolves. Existing systems increasingly make semantic computation explicit, but how semantic state should persist, inform ongoing generation, and be updated throughout dialogue remains underexplored. We propose Bridge-Duplex, a persistent semantic–generative dual-stream framework for full-duplex S2S dialogue. Bridge attention connects the two streams on a shared causal timeline, exposing persistent semantic memory to ongoing generation. Dialogue next-latent prediction further regularizes semantic state evolution by predicting each subsequent semantic state from the current user–system state pair and next-token evidence during training. Bridge-Duplex improves multi-turn dialogue quality and context consistency over the underlying full-duplex models. Selective bridge attention preserves most of the multi-turn semantic performance of dense bridge attention while improving inference efficiency through selective semantic memory access. Audio samples are available at `https://anonymous77077.github.io/Bridge-Duplex_Demo/`.

1 INTRODUCTION

Full-duplex speech-to-speech (S2S) dialogue requires understanding, interaction decisions, and speech generation to unfold concurrently as conversational evidence changes over time. Earlier spoken dialogue systems typically relied on cascaded pipelines connecting automatic speech recognition (ASR), text-based language modeling, and text-to-speech (TTS) synthesis (Ji et al., 2024; Chen et al., 2025; Lu et al., 2026). This modular design enables direct access to the knowledge and reasoning capabilities of text-based large language models (LLMs). However, sequential execution introduces component-wise latency and may discard paralinguistic information during speech-to-text conversion. Recent end-to-end full-duplex S2S models address these limitations by modeling speech streams directly within a unified architecture, enabling more responsive interaction while retaining acoustic information that is difficult to represent in text alone (Défossez et al., 2024). However, these models must sustain dialogue-level semantic processing while continuously generating speech, and speech models still lag behind text LLMs on knowledge-intensive and complex reasoning tasks (Liu et al., 2026; Wang et al., 2026). Strengthening semantic capability without compromising real-time full-duplex interaction therefore remains an important challenge.

Recent research has consequently explored multiple approaches to strengthen semantic processing within full-duplex S2S systems. Some approaches incorporate external LLM guidance or retrieved knowledge (Kuroki et al., 2026; Chien et al., 2026), others perform semantic or reasoning-related computation concurrently with ongoing interaction (Wu et al., 2026; Shih et al., 2026), and modular Thinker–Talker architectures coordinate semantic processing with speech generation (Bai et al., 2026). These developments make semantic processing increasingly explicit in real-time spoken dialogue. Yet these approaches leave open how dialogue-level semantic state should persist across turns, inform ongoing generation, and evolve as interaction unfolds.

We organize this design problem around three requirements: semantic persistence, cross-stream communication, and semantic state evolution. Semantic persistence concerns maintaining dialogue-

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.

level semantic information throughout the interaction as new conversational evidence becomes available. Cross-stream communication concerns how this evolving semantic information should inform ongoing full-duplex generation. Semantic state evolution concerns how semantic states should change as user and system contributions accumulate while preserving information needed for coherent continuation. This framing shifts the focus from merely introducing semantic capability to maintaining, exposing, and updating semantic state on a shared streaming timeline.

To address this challenge, we propose Bridge-Duplex, which maintains role-separated dialogue text as a persistent semantic stream alongside frame-wise audio–text generation on a shared causal timeline. To connect the two streams, we introduce bridge attention, which causally exposes this semantic history to the generative stream. For efficient communication over growing dialogue histories, we introduce selective bridge attention, which retrieves query-relevant semantic blocks instead of accessing the full semantic history. Separately, we introduce a dialogue-specific extension of next-latent prediction (NextLat), termed Dialogue NextLat, which models role-specific transitions from paired user and system states during training. Across two full-duplex backbones, Bridge-Duplex improves multi-turn semantic performance, while selective bridge attention retains nearly all of the gains of dense coupling at lower inference cost. The framework makes the following contributions:

- **Semantic Persistence:** We introduce a persistent semantic stream alongside frame-wise generation, maintaining dialogue-level semantic history throughout full-duplex interaction to strengthen semantic processing.
- **Cross-Stream Communication:** Bridge attention enables efficient semantic-to-generative coupling, preserving the semantic quality of dense access while reducing its decoding cost through selective memory retrieval.
- **Semantic State Evolution:** Dialogue NextLat regularizes role-specific semantic transitions conditioned on paired user–system states, improving multi-turn context consistency over Standard NextLat without an additional autoregressive inference stage.

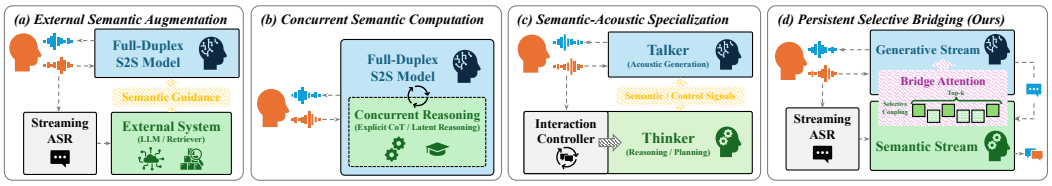


Figure 1: Conceptual comparison of semantic processing approaches for full-duplex S2S dialogue.

2 Related Work and Background

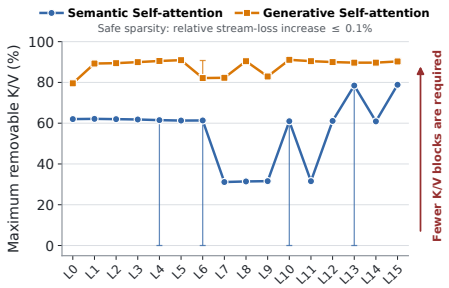
Semantic Processing Strategies in Full-Duplex S2S. End-to-end full-duplex S2S models (Défossez et al., 2024; Züfle et al., 2026; Roy et al., 2026) enable streaming spoken interaction but generally lag behind text-conditioned counterparts on complex reasoning tasks, motivating recent efforts to strengthen semantic processing under full-duplex streaming constraints (Liu et al., 2026; Wang et al., 2026). One direction augments streaming S2S models with external semantic resources (Figure 1a). KAME (Kuroki et al., 2026) uses a concurrently operating back-end text LLM for response guidance, and MoshiRAG (Chien et al., 2026) asynchronously incorporates retrieved knowledge during full-duplex interaction. A second direction performs semantic or reasoning-related computation concurrently with streaming interaction (Figure 1b). The think-while-listening approach (Shih et al., 2026) begins text-based chain-of-thought reasoning before the complete user query has arrived, whereas FLAIR (Wu et al., 2026) recurrently updates latent representations during speech perception. A further direction separates high-level semantic processing from speech generation (Figure 1c). Qwen2.5-Omni (Xu et al., 2025) conditions streaming speech generation on representations from a text-oriented Thinker, and JoyAI-Talker (Bai et al., 2026) extends this Thinker–Talker organization to full-duplex dialogue. More closely related to our motivation, Lychee-FD (Liu et al., 2026) addresses semantic–acoustic interference from shared-parameter optimization through hierarchical parameter separation and semantic alignment. Building on this specialization paradigm, Bridge-Duplex focuses on maintaining and causally exposing evolving semantic memory throughout full-duplex interaction (Figure 1d).

Specialized Computation and Selective Context Access. Recent architectures address two complementary concerns: specializing computation across heterogeneous representation types and selectively allocating computation over growing historical context. Mixture-of-Transformers architectures (Liang et al., 2025; Agarwal et al., 2026) address the former by assigning different representations to specialized Transformer paths while preserving information exchange through shared attention context. For the latter, sparse-attention and key/value (K/V) compression methods reduce long-context computation by identifying informative tokens or blocks and restricting subsequent access to selected context (Tang et al., 2024; Li et al., 2024; Lai et al., 2026). Together, these approaches support non-uniform allocation of computation across both representation types and historical context. In full-duplex S2S, these concerns matter more because semantic processing and continuous generation must proceed concurrently as dialogue history grows.

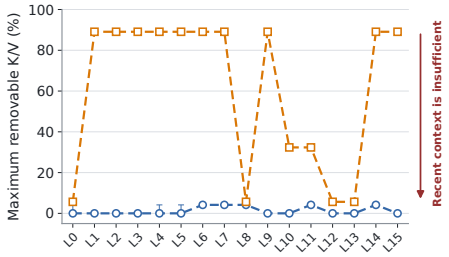
Latent-State Prediction and Representation Dynamics. Beyond selective context access, maintaining persistent representations raises the question of how their states evolve as new evidence arrives. Self-predictive representation learning has explored this problem by learning latent states together with transition dynamics that predict future representations (Ni et al., 2024). Standard autoregressive training supervises next-token prediction but does not explicitly constrain transitions between successive latent representations. NextLat addresses this by predicting future representations from the current state and new evidence and matching them to detached target representations (Teoh et al., 2025; Shi et al., 2026). This perspective is relevant to full-duplex S2S, where semantic states must evolve as user and system contributions accumulate over an ongoing dialogue.

3 Motivating Analysis: Stream-Wise Context Dependence

To characterize stream-wise context dependence under semantic–generative specialization, we analyze a dual-stream configuration with separately parameterized semantic and generative streams, each using dense causal self-attention over its full K/V history. We examine whether the two streams differ in their dependence on historical context and whether recent context alone captures the most relevant states. To this end, we perform a post-hoc K/V pruning analysis under dense causal attention, following the query-aware diagnostic of Tang et al. (2024). We report safe sparsity, defined as the largest measured fraction of K/V accesses removed for which the corresponding stream loss increases by no more than 0.1%; the complete pruning and aggregation protocol is provided in Appendix A.



(a) Attention-guided K/V removal



(b) Recency-based K/V removal

Figure 2: Layer-wise loss-constrained K/V removability. Error bars denote conversation-level 95% bootstrap confidence intervals.

Figure 2a evaluates attention-guided pruning, where K/V blocks are ranked by their dense attention importance and low-importance blocks are removed first. The generative stream tolerates substantially greater removal than the semantic stream across Transformer layers, suggesting greater redundancy in its contextual K/V states. Although the semantic stream also admits some removal, its lower safe sparsity indicates greater dependence on a broader set of contextual K/V states. Figure 2b instead applies a fixed-recency policy that progressively removes older context. The generative stream remains relatively robust when it retains only recent context, consistent with greater temporal locality in frame-wise generation. In contrast, the semantic stream degrades markedly under the same policy, suggesting that useful semantic context is not confined to the most recent dialogue history. These results reveal asymmetric context dependence between the two streams and support differentiated memory-access policies rather than uniform treatment of their histories. This asymmetry motivates the subsequent architecture to preserve broad semantic context while adapting how it is exposed to frame-wise generation.

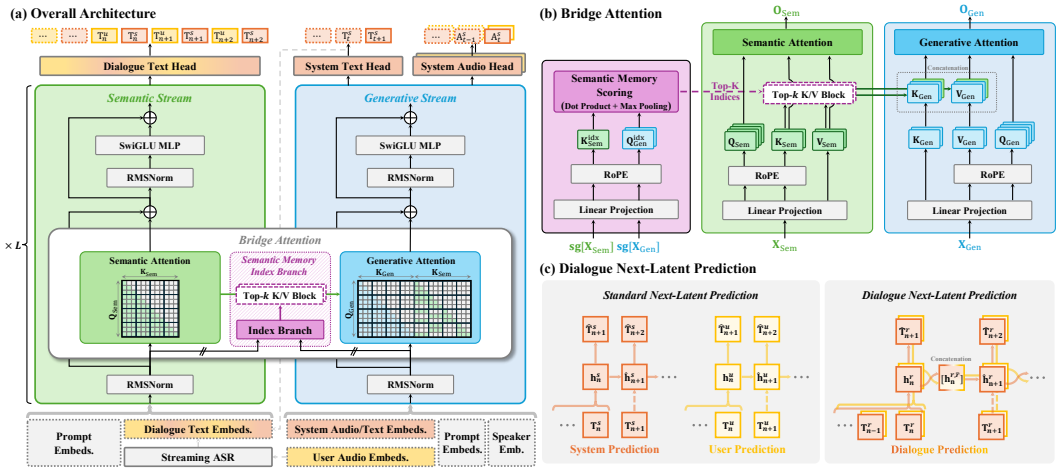


Figure 3: Overview of the separately parameterized dual-stream instantiation of Bridge-Duplex.

4 Bridge-Duplex

Bridge-Duplex augments an autoregressive full-duplex S2S backbone with persistent semantic and generative streams aligned on a shared causal timeline, as illustrated in Figure 3. Bridge attention couples the streams, while Dialogue NextLat regularizes semantic state transitions during training.

4.1 Semantic–Generative Dual-Stream Architecture

Bridge-Duplex organizes computation into two layer-aligned paths: a semantic stream (Sem) and a generative stream (Gen). The semantic stream processes role-separated user and system text and is trained with role-separated next-token prediction; its K/V history serves as semantic memory. The generative stream uses the backbone's frame-wise audio–text inputs to predict text and acoustic tokens. The streams operate at different temporal resolutions: the generative stream advances at every codec frame, whereas the semantic stream updates only when a text token from either the user or the system becomes causally available, preserving that token's codec-frame position. During training, the semantic stream uses ground-truth system text and ground-truth user text with streaming-text augmentation to simulate delayed and incomplete streaming ASR observations (Appendix B.2.2). During inference, it receives predicted system text from the generative stream and user text from streaming ASR. Until the next text token becomes available, the generative stream can access the semantic memory through bridge attention.

4.2 Bridge Attention

We first define dense bridge attention, where each generative query jointly attends to semantic and generative history. We then introduce selective bridge attention to reduce computation by restricting semantic memory access to query-relevant blocks.

4.2.1 Dual-Stream Joint Causal Attention

Inspired by shared-context specialized Transformer computation (Liang et al., 2025; Agarwal et al., 2026), dual-stream joint causal attention couples the semantic and generative streams through a joint causal attention context rather than a separate cross-attention module. Let $\mathbf{Q}_{\text{Sem}}$, $\mathbf{K}_{\text{Sem}}$, and $\mathbf{V}_{\text{Sem}}$ denote the query, key, and value projections of the semantic stream, and let $\mathbf{Q}_{\text{Gen}}$, $\mathbf{K}_{\text{Gen}}$, and $\mathbf{V}_{\text{Gen}}$ denote their generative counterparts. The two attention outputs are defined as

$$\begin{aligned} \mathbf{O}_{\text{Sem}} &= \text{Attn}_{\text{causal}}\left(\mathbf{Q}_{\text{Sem}}, \mathbf{K}_{\text{Sem}}, \mathbf{V}_{\text{Sem}}\right), \\ \mathbf{O}_{\text{Gen}} &= \text{Attn}_{\text{causal}}\left(\mathbf{Q}_{\text{Gen}}, [\mathbf{K}_{\text{Gen}}; \mathbf{K}_{\text{Sem}}], [\mathbf{V}_{\text{Gen}}; \mathbf{V}_{\text{Sem}}]\right), \end{aligned} \tag{1}$$

where $[\cdot;\cdot]$ denotes concatenation along the sequence dimension. The original codec-frame positions on the shared dialogue timeline determine causal visibility. Accordingly, the semantic stream attends only to its own causal history, whereas each generative query jointly attends to causally visible semantic and generative keys under a single softmax.

4.2.2 SELECTIVE SEMANTIC MEMORY COUPLING

Dense bridge attention allows each generative query to attend to the entire causally available semantic history, but its cost grows with dialogue length. Section 3 shows that semantic context is not well characterized by a fixed-recency policy, motivating the preservation of broad semantic history. To control the cost of exposing this growing history to generation, we introduce selective semantic memory coupling, which preserves semantic-stream self-attention while restricting each generative query to query-relevant semantic memory blocks, including earlier context.

Semantic Memory Index Branch. Inspired by MiniMax Sparse Attention (Lai et al., 2026), we use a lightweight index branch to select a small set of semantic memory blocks for each generative query. We perform selection separately for predefined query-head groups, indexed by g, with the selected blocks shared by the query heads within each group. To score the semantic memory, the index branch forms separate learned projections of the generative and semantic hidden states, denoted by $\mathbf{Q}^{\mathrm{idx}}_{\mathrm{Gen}}$ and $\mathbf{K}^{\mathrm{idx}}_{\mathrm{Sem}}$. For generative position t and semantic state j, let τ_j denote the codec-frame position at which state j becomes causally available on the shared timeline. The relevance score for query-head group g is

$$S^{\mathrm{idx},(g)}_{t,j} = \frac{\left(\mathbf{Q}^{\mathrm{idx}}_{\mathrm{Gen}}\right)^{(g)}_{t} \left(\mathbf{K}^{\mathrm{idx}}_{\mathrm{Sem}}\right)^{\top}_{j}}{\sqrt{d_{\mathrm{idx}}}}, \tag{2}$$

where d_{idx} denotes the shared dimensionality of the index representations.

The semantic memory is partitioned in memory order into N_B contiguous blocks, $\{\mathcal{B}_b\}_{b=1}^{N_B}$, each containing at most B_{Sem} states. The block relevance and selected block indices are

$$M^{\mathrm{idx},(g)}_{t,b} = \max_{\substack{j\in\mathcal{B}_b \\ \tau_j \le t}} S^{\mathrm{idx},(g)}_{t,j}, \qquad \mathcal{I}^{(g)}_{t} = \mathrm{TopK}_{b\in\{1,\dots,N_B\}}\left(M^{\mathrm{idx},(g)}_{t,\cdot}, k\right). \tag{3}$$

Here, $\mathcal{I}^{(g)}_t$ denotes the selected semantic block indices for query-head group g. The Top-k selection is constrained to include the most recent causally available semantic block within the k-block budget.

Main Attention Branch. The selected block indices $\mathcal{I}^{(g)}_t$ define the full-dimensional semantic K/V support used by the main attention branch. Denoting the corresponding selected states by $\widetilde{\mathbf{K}}_{\mathrm{Sem}}$ and $\widetilde{\mathbf{V}}_{\mathrm{Sem}}$, the generative attention in Eq. 1 becomes

$$\mathbf{O}_{\mathrm{Gen}} = \mathrm{Attn}_{\mathrm{causal}}\left(\mathbf{Q}_{\mathrm{Gen}}, \left[\mathbf{K}_{\mathrm{Gen}}; \widetilde{\mathbf{K}}_{\mathrm{Sem}}\right], \left[\mathbf{V}_{\mathrm{Gen}}; \widetilde{\mathbf{V}}_{\mathrm{Sem}}\right]\right). \tag{4}$$

Therefore, selective semantic memory coupling realizes selective bridge attention by restricting semantic memory access to the retrieved blocks while retaining the causal generative context and joint attention structure.

4.3 DIALOGUE NEXT-LATENT PREDICTION

Bridge attention controls how semantic memory is exposed to generation, while semantic state evolution is addressed separately. Although next-token supervision updates semantic representations as new text arrives, it does not explicitly regularize transitions between successive semantic states. We therefore introduce Dialogue NextLat, a training-time extension of NextLat (Teoh et al., 2025) that conditions each role-specific transition on paired user and system states.

For target role $r \in \{s, u\}$, let n index its semantic updates, with $\mathbf{h}^r_n$ denoting its final-layer semantic hidden state and $\mathbf{h}^{\bar{r}}_n$ the latest causally available state of the opposite role. We define $\mathbf{h}^{r,\bar{r}}_n = [\mathbf{h}^r_n; \mathbf{h}^{\bar{r}}_n]$ as the paired semantic state. Let $\mathbf{T}^r_{n+1}$ denote the embedding of the ground-truth token at the next semantic update for role r during training. Using a shared transition predictor f_ψ, the next semantic state is predicted as

$$\widehat{\mathbf{h}}^r_{n+1} = \mathbf{h}^r_n + f_\psi\left(\mathrm{Norm}\left(\left[\mathbf{h}^{r,\bar{r}}_n; \mathbf{T}^r_{n+1}\right]\right)\right). \tag{5}$$

The paired state $\mathbf{h}^{r,\bar{r}}_n$ allows each transition to depend on both dialogue roles, and the shared f_ψ models both system-target and user-target transitions. The predicted state is supervised in both hidden and token-distribution spaces, as defined in Section 4.4. Dialogue NextLat is used only during training, and the transition predictor is removed during autoregressive inference.

4.4 TRAINING OBJECTIVES

Bridge-Duplex jointly optimizes stream-specific objectives for the generative and semantic streams, together with an objective for semantic memory selection.

Generative Stream Objective. The generative stream retains the underlying full-duplex backbone (Défossez et al., 2024; Züfle et al., 2026) objectives for predicting text and acoustic tokens, which we denote collectively by $\mathcal{L}_{\mathrm{Gen}}$.

Semantic Stream Objective. The semantic stream objective jointly combines role-separated next-token prediction with hidden-state and token-distribution consistency under Dialogue NextLat. The next-token term $\mathcal{L}_{\text{next-token}}$ is the role-separated cross-entropy objective for predicting the next semantic token for each dialogue role. Let $\mathcal{V}$ denote the set of valid consecutive semantic transitions within each target role. For each transition $(n, r) \in \mathcal{V}$, hidden-state consistency is optimized as

$$\mathcal{L}_{\text{next-h}} = \frac{1}{|\mathcal{V}|} \sum_{(n,r)\in\mathcal{V}} \mathrm{SmoothL1}\left(\mathrm{sg}\left[\mathbf{h}^r_{n+1}\right], \widehat{\mathbf{h}}^r_{n+1}\right), \tag{6}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. Let $p_{\mathrm{Sem}}(\cdot \mid \mathbf{h})$ denote the token distribution produced by the semantic stream text head. Token-distribution consistency is optimized as

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{|\mathcal{V}|} \sum_{(n,r)\in\mathcal{V}} D_{\mathrm{KL}}\left(p^{\mathrm{sg}}_{\mathrm{Sem}}\left(\cdot \mid \mathrm{sg}\left[\mathbf{h}^r_{n+1}\right]\right) \,\|\, p^{\mathrm{sg}}_{\mathrm{Sem}}\left(\cdot \mid \widehat{\mathbf{h}}^r_{n+1}\right)\right), \tag{7}$$

where $p^{\mathrm{sg}}_{\mathrm{Sem}}$ denotes the semantic stream text head with its parameters held fixed under $\mathcal{L}_{\mathrm{KL}}$. Thus, the target semantic state is detached while gradients through the predicted semantic state are preserved. The semantic stream objective is defined as

$$\mathcal{L}_{\mathrm{Sem}} = \mathcal{L}_{\text{next-token}} + \lambda_{\text{next-h}}\mathcal{L}_{\text{next-h}} + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}. \tag{8}$$

Semantic Memory Selection Objective. Since the Top-k selection in Eq. 3 is non-differentiable, we train the index branch to approximate the token-level semantic attention distribution of the main branch, following MiniMax Sparse Attention (Lai et al., 2026). Let $p^{(g)}_{\mathrm{idx},t}$ and $p^{(g)}_{\mathrm{main},t}$ denote the corresponding token-level distributions over semantic states for the index and main branches, respectively, normalized over the same stage-specific semantic memory support. We minimize

$$\mathcal{L}_{\mathrm{idx}} = \frac{1}{|\mathcal{Q}|} \sum_{(t,g)\in\mathcal{Q}} D_{\mathrm{KL}}\left(\mathrm{sg}\left[p^{(g)}_{\mathrm{main},t}\right] \,\|\, p^{(g)}_{\mathrm{idx},t}\right), \tag{9}$$

where $\mathcal{Q}$ denotes the set of valid (t, g) pairs.

Staged Optimization. Training follows a dense-to-selective schedule. We first optimize the dual-stream model with dense bridge attention, then train the index branch using the dense main attention distribution as supervision, and finally activate selective semantic memory coupling. Stage-specific freezing, support normalization, and optimization settings are detailed in Appendix B.2.

5 EXPERIMENTS AND RESULTS

Datasets. We use Behavior-SD (Lee et al., 2025) as the task-specific full-duplex adaptation corpus for all Bridge-Duplex variants. Its 2,164 hours of English multi-turn, two-speaker conversations include full-duplex behaviors such as backchannels and interruptions, closely matching the interaction setting targeted in this work. The speech is synthetically generated using CosyVoice TTS (Du et al., 2024), with further preprocessing and evaluation details provided in Appendix B.1.

Implementation Details. Bridge-Duplex uses the same semantic–generative dual-stream formulation across backbones with different parameterization strategies. Our primary F-Actor-based instantiation uses separately parameterized Transformer streams, while a Moshi-based variant uses stream-specific low-rank adaptation (LoRA) modules over a frozen shared backbone (Hu et al., 2022). During inference, a frozen causal streaming ASR frontend provides user-side semantic observations. Detailed backbone parameterization, training, and inference configurations are provided in Appendix B.2.

Evaluation Metrics. We evaluate multi-turn semantics using Dialogue Quality (DQ) and Context Consistency (CC). For MTR-DuplexBench (MTR-DB) (He et al., 2026) and Behavior-SD (Lee et al., 2025), we follow the turn-by-turn evaluation protocol of MTR-DB, using its DQ criterion and a CC criterion adapted from M3-DuplexBench (Fukuda et al., 2026). We also evaluate two-instance dialogue generation, where generated responses propagate across turns as the dialogue evolves. DQ and CC are scored using Qwen3-14B (Yang et al., 2025) as the LLM judge. For full-duplex interaction, we follow the subset-specific protocols of Full-Duplex-Bench v1.0 and v1.5 (Lin et al., 2025; 2026). Detailed evaluation procedures are provided in Appendix B.4.

Table 1: Comparison with representative full-duplex speech-to-speech systems.

Model	LLM Params.	Multi-Turn Semantic				Full-Duplex Interaction			
		MTR-DB		*Behavior-SD*		Turn-Taking	Interruption	Interruption	Backchannel
		DQ ↑	CC ↑	DQ ↑	CC ↑	TOR ↑	Respond ↑	Stop Lat. ↓	JSD ↓
Conventional Speech-to-Speech									
Moshi (Défossez et al., 2024)	7B	1.99	2.64	2.91	3.49	**1.000**	0.560	1.401	1.000
F-Actor (Züfle et al., 2026)	1B	2.73	3.90	3.51	4.13	0.950	0.425	1.010	0.887
PersonaPlex (Roy et al., 2026)	7B	2.09	2.71	3.36	4.06	0.950	0.650	**0.581**	**0.779**
Externally Augmented Speech-to-Speech									
KAME (Kuroki et al., 2026)	7B	2.65	3.10	2.67	2.34	0.874	0.855	1.595	1.000
MoshiRAG (Chien et al., 2026)	7B	2.52	3.11	2.90	3.48	0.857	0.865	1.774	0.975
Proposed: Selective Bridge-Duplex									
Moshi Backbone‡	7B	3.01	**4.34**	**3.82**	**4.59**	0.622	**0.870**	0.700	0.948
F-Actor Backbone†	1B/1B	**3.03**	4.09	3.68	4.31	0.866	0.800	1.395	0.849

† Separately parameterized dual-stream, with parameter sizes reported as semantic/generative; ‡ LoRA-based dual-stream.

Table 2: Controlled comparison of semantic-processing strategies under the F-Actor backbone.

Method	LLM Params.	Multi-Turn Semantic						Inference Efficiency
		MTR-DB		*Behavior-SD*		*Two-Instance Dialogue*		E2E
		DQ ↑	CC ↑	DQ ↑	CC ↑	DQ ↑	CC ↑	RTF ↓
Baseline and Capacity Controls								
F-Actor (Züfle et al., 2026)	1B	2.73	3.90	3.51	4.13	3.42	3.47	0.5318
	3B	2.92	3.97	3.63	4.25	3.50	3.63	0.7480
Comparison Methods								
+ Oracle Conditioning (Kuroki et al., 2026)	1B	2.79	3.94	3.41	4.01	3.45	3.48	0.5348
+ Latent Reasoning (Wu et al., 2026)	1B	2.90	4.03	3.55	4.18	3.48	3.49	1.0260
Proposed Variants								
Dense Bridge-Duplex†	1B/1B	**3.04**	**4.09**	**3.72**	**4.35**	**3.51**	**3.68**	1.0055
Selective Bridge-Duplex†	1B/1B	3.03	**4.09**	3.68	4.31	3.49	3.67	0.8287

† Separately parameterized dual-stream, with parameter sizes reported as semantic/generative.

5.1 Multi-Turn Semantic Performance and Controlled Analysis

Table 1 compares Bridge-Duplex with representative conventional full-duplex S2S systems (Défossez et al., 2024; Züfle et al., 2026; Roy et al., 2026) and systems that augment full-duplex generation with external semantic or knowledge processing (Kuroki et al., 2026; Chien et al., 2026). Across both F-Actor-based and Moshi-based instantiations, Bridge-Duplex consistently improves DQ and CC over the corresponding backbone on both MTR-DB and Behavior-SD. The gains across both separately parameterized and LoRA-based dual-stream instantiations suggest that maintaining dialogue-level semantic information alongside continuous generation benefits multi-turn contextual modeling without requiring fully separate Transformer parameterizations. At the same time, Bridge-Duplex maintains overall full-duplex interaction capability while primarily improving multi-turn semantic performance.

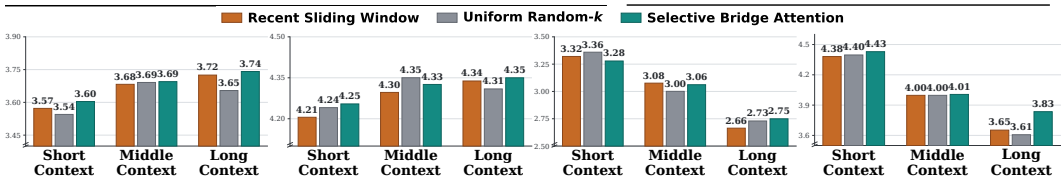
Table 2 further examines model capacity and alternative semantic processing strategies under matched F-Actor and Behavior-SD training conditions. In addition to MTR-DB and Behavior-SD, we report DQ and CC under the two-instance dialogue generation setting (Züfle et al., 2026). Increasing the F-Actor backbone to 3B improves DQ and CC across all three evaluation settings, while the smaller Bridge-Duplex variants remain broadly stronger across the evaluated semantic metrics, suggesting that the gains are not explained by parameter scaling alone. As additional controls, Oracle Conditioning provides explicit text-based semantic guidance (Kuroki et al., 2026), and Latent Reasoning introduces concurrent latent-state reasoning during speech perception (Wu et al., 2026). These comparison methods improve semantic performance in several settings, while Bridge-Duplex remains stronger across all six DQ and CC measures. The two-instance results further indicate that the semantic gains extend to settings where subsequent dialogue context is constructed through ongoing model-generated interaction rather than reference history. Within Bridge-Duplex, selective bridge attention preserves nearly all of the multi-turn semantic performance of dense coupling while reducing end-to-end real-time factor (RTF) by approximately 18%.

Table 3: Comparison of semantic memory access strategies for turns with over eight semantic memory blocks.

Configuration	Blocks Budget	*MTR-DB* DQ ↑	*MTR-DB* CC ↑	*Behavior-SD* DQ ↑	*Behavior-SD* CC ↑	E2E RTF ↓
F-Actor (Single-Stream)	0	2.28	3.53	3.48	4.04	0.5318
Dense Bridge Attention	All	2.62	3.70	3.73	4.32	1.0055
Recent Sliding Window	4	2.68	3.78	3.72	4.35	0.7668
Uniform Random-k	4	2.64	3.77	3.68	4.34	0.7750
Selective Bridge Attention	4	2.70	3.82	3.74	4.35	0.8287

Table 4: Analysis of NextLat role conditioning with F-Actor as a single-stream reference.

Configuration	*MTR-DB* DQ ↑	*MTR-DB* CC ↑	*Behavior-SD* DQ ↑	*Behavior-SD* CC ↑
F-Actor (Single-Stream)	2.73	3.90	3.51	4.13
Standard NextLat	2.82	3.90	3.66	4.30
Dialogue NextLat	3.04	4.09	3.72	4.35



(a) *Behavior-SD* DQ ↑ (b) *Behavior-SD* CC ↑ (c) *MTR-DB* DQ ↑ (d) *MTR-DB* CC ↑

Figure 4: Context-length analysis of semantic-memory access across three equal-sized groups.

5.2 Component Analysis

To examine semantic memory access under longer contexts, Table 3 focuses on target turns with over eight semantic memory blocks, with RTF measured over the full evaluation set. We compare variants separately fine-tuned with a recent sliding window, uniform random-k selection, or selective bridge attention against dense bridge attention. Selective bridge attention achieves the strongest or tied-best sparse scores under the same block budget, while retaining query-dependent access to earlier semantic context. Figure 4 extends this analysis to the full evaluation set by dividing target turns into three equally sized groups according to preceding context length. Across these groups, the advantage of selective retrieval becomes more apparent as context grows, suggesting that query-dependent semantic memory access may become increasingly useful for longer dialogue histories.

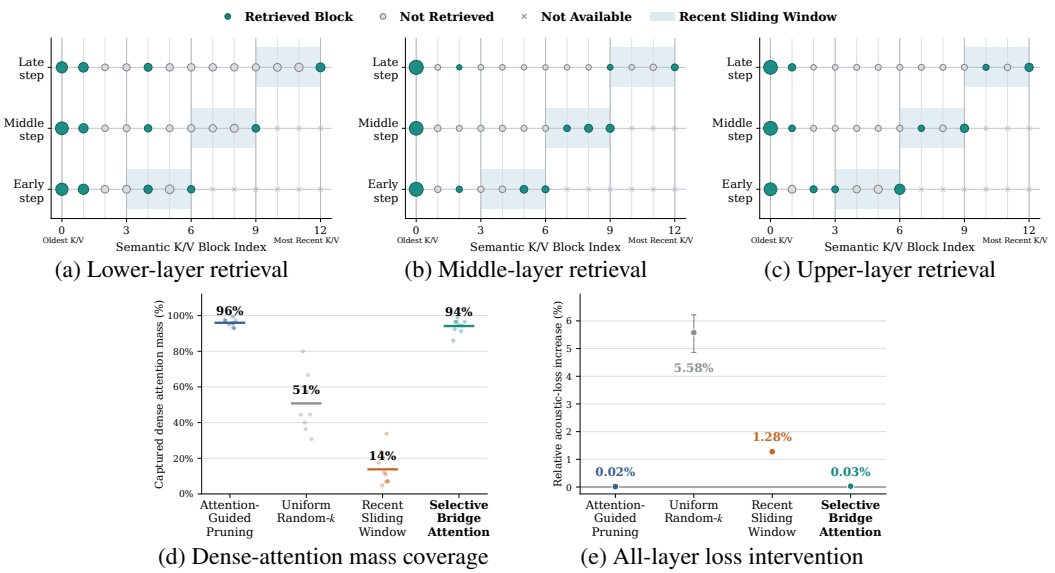
We next examine role conditioning in semantic transition prediction in Table 4. The F-Actor baseline serves as a single-stream reference for comparison with the two dual-stream NextLat variants. Both NextLat variants use the same bridge attention configuration and semantic stream objective structure in Eq. 8. Standard NextLat predicts user and system semantic transitions independently, whereas Dialogue NextLat conditions each target-role transition on the paired user–system semantic state. Dialogue NextLat improves all four DQ and CC measures over Standard NextLat, suggesting that paired-role conditioning provides additional benefit for semantic state transitions.

5.3 Analysis of Query-Dependent Semantic Memory Retrieval

Section 3 showed that the semantic stream depends on a broader set of contextual states than the generative stream and is less well served by a fixed-recency policy. This post-hoc analysis motivates preserving broad semantic history, but does not reveal whether a learned selector can identify the relevant support once selective bridge attention is introduced. We therefore examine three complementary aspects of the learned cross-stream behavior: which semantic blocks are selected, how much attention mass under dense bridge attention they capture, and how restricting semantic memory access to the selected blocks affects acoustic loss. We benchmark selective bridge attention against attention-guided pruning, which ranks blocks using attention mass under dense bridge attention and serves as a diagnostic reference. We additionally compare a recent sliding window and uniform random-k selection under the same budget of four 64-state semantic memory blocks.

Figures 5a–5c qualitatively visualize semantic memory retrieval for a representative conversation at different Transformer depths. Teal circles indicate blocks selected by selective bridge attention, and the shaded region shows the recent sliding-window baseline for reference. Marker size encodes the attention mass assigned to each semantic memory block under dense bridge attention. In this example, the selected sets include both recent and earlier blocks, and their composition varies across generative query positions and Transformer depths. This visualization illustrates that the learned selector can adapt its semantic memory access to the current query and layer-wise representation rather than following a fixed recency-based policy.

Figures 5d and 5e provide aggregate analyses of retrieval fidelity and its effect on acoustic loss. Figure 5d measures the fraction of this attention mass captured by the blocks selected by each



(a) Lower-layer retrieval (b) Middle-layer retrieval (c) Upper-layer retrieval

(d) Dense-attention mass coverage (e) All-layer loss intervention

Figure 5: Analysis of query-dependent semantic-memory retrieval and dense-attention coverage.

policy. Selective bridge attention uses the lightweight index branch to select semantic blocks before the main attention branch is evaluated, restricting full-dimensional semantic K/V access to the retrieved support. Despite this constraint, its selected support achieves nearly the same dense attention mass coverage as attention-guided pruning, indicating close alignment with the semantic memory emphasized by dense bridge attention. Figure 5e then applies each selection policy across all Transformer layers and measures the resulting relative increase in acoustic loss. Selective bridge attention introduces only a small loss increase, comparable to attention-guided pruning and lower than the recent-window and random alternatives. Together, these analyses show that selective bridge attention identifies sparse semantic memory support that closely aligns with dense attention while limiting acoustic degradation.

6 Limitations

Bridge-Duplex incurs inference cost from maintaining a separate semantic path and semantic-to-generative bridge attention. Selective bridge attention reduces this overhead but does not eliminate the additional computation inherent to the dual-stream architecture. Appendix C.4 further analyzes how K/V organization and attention backends affect its realized latency. Additionally, the semantic stream relies on user text from streaming ASR at inference, making persistent semantic memory sensitive to transcription errors and delays in causal text observations. Training-time streaming-text augmentation improves robustness to delayed and incomplete streaming ASR observations but does not eliminate this dependence on explicit text input. Future work will explore semantic pathways that preserve multi-turn semantic performance without explicit ASR transcription, further reduce dual-stream overhead, and preserve causal, low-latency coordination with generation.

7 Conclusion

We presented Bridge-Duplex, a semantic–generative dual-stream framework that coordinates dialogue-level semantic processing with continuous frame-wise generation in full-duplex S2S dialogue. Bridge attention connects the streams through persistent semantic memory on a shared causal timeline; selective retrieval improves the efficiency of this communication, while Dialogue NextLat regularizes semantic state transitions during training. Across distinct backbone instantiations, Bridge-Duplex improves multi-turn semantic performance relative to the underlying full-duplex models. Selective bridge attention preserves nearly all of the multi-turn semantic performance of dense bridge attention while reducing the end-to-end real-time factor. Together, these results support persistent semantic memory, causal semantic access, and regularized state transitions as useful design choices for multi-turn semantic modeling in full-duplex S2S dialogue.

Reproducibility Statement

We provide detailed descriptions of the training and evaluation data, model configurations, training procedures, streaming inference setup, comparison-model implementations, and evaluation protocols in Appendix B.1–B.4. The complete prompts used for multi-turn semantic evaluation are provided in Figures 10 and 11, with additional analyses reported in the subsequent appendix sections.

AI Use Statement

We used OpenAI GPT-5.6 to assist with language editing and drafting, literature discovery, and reviewing code and experimental design. The authors reviewed and verified all AI-assisted outputs. We take full responsibility for the final content of this work, including any text or artifacts produced with the assistance of generative AI.

References

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Yinhao Bai, Jinming Chen, Yafeng Chen, Wei Deng, Boya Dong, Nan Duan, Yu Gu, Weisheng Han, Yankun Huang, Ming Ke, et al. Joyai-talker: Full-duplex speech interactive large model built for empathetic voice agents. *arXiv preprint arXiv:2608.01119*, 2026.

Edresson Casanova, Paarth Neekhara, Ryan Langman, Shehzeen Hussain, Subhankar Ghosh, Xuesong Yang, Ante Jukic, Jason Li, and Boris Ginsburg. NanoCodec: Towards High-Quality Ultra Fast Speech LLM Inference. In *Interspeech 2025*, pp. 5028–5032, 2025. doi: 10.21437/Interspeech.2025-827.

Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, Kun Liu, Wenpeng Li, Xu Li, Ziyuan Li, Feiyu Shen, Xu Tang, et al. Fireredchat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *arXiv preprint arXiv:2509.06502*, 2025.

Chung-Ming Chien, Manu Orsini, Eugene Kharitonov, Neil Zeghidour, Karen Livescu, and Alexandre Défossez. MoshiRAG: Asynchronous knowledge retrieval for full-duplex speech language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=4aI2vOyyHH.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Nauman Dawalatabad, Mirco Ravanelli, François Grondin, Jenthe Thienpondt, Brecht Desplanques, and Hwidong Na. ECAPA-TDNN Embeddings for Speaker Diarization. In *Interspeech 2021*, pp. 3560–3564, 2021. doi: 10.21437/Interspeech.2021-941.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024.

Ryo Fukuda, Atsushi Ando, Hiroki Kanagawa, Takatomo Kano, Marc Delcroix, Naohiro Tawara, and Yuya Chiba. M3-duplexbench: A multi-turn, multilingual, multidomain benchmark for full-duplex spoken dialogue models. *arXiv preprint arXiv:2607.29125*, 2026.

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

Zhang He, Wenqian Cui, Haoning Xu, Xiao-Hui Li, Lei Zhu, Haoli Bai, Ma Shaohua, and Irwin King. MTR-DuplexBench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5334–5351, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.263. URL https://aclanthology.org/2026.findings-acl.263/.

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 *International Conference on Learning Representations*, 2022. URL https://mlanthology.org/iclr/2022/hu2022iclr-lora/.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*, 2024.

So Kuroki, Yotaro Kubo, Takuya Akiba, and Yujin Tang. Kame: Tandem architecture for enhancing knowledge in real-time speech-to-speech conversational ai. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19362–19366, 2026. doi: 10.1109/ICASSP55912.2026.11464526.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, et al. Minimax sparse attention. *arXiv preprint arXiv:2606.13392*, 2026.

Sehun Lee, Kang-wook Kim, and Gunhee Kim. Behavior-SD: Behaviorally aware spoken dialogue generation with large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9574–9593, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.484. URL https://aclanthology.org/2025.naacl-long.484/.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970. Curran Associates, Inc., 2024. doi: 10.52202/079017-0722. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/28ab418242603e0f7323e54185d19bde-Paper-Conference.pdf.

Weixin Liang, LILI YU, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Nu6N69i8SB.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2025.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-Yi Lee. Full-duplex-bench v1.5: Evaluating overlap handling for full-duplex speech models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19447–19451, 2026. doi: 10.1109/ICASSP55912.2026.11463576.

Zhenyu Liu, Xuanyu Zhang, Yunxin Li, Qixun Teng, Shenyuan Jiang, Haolan Chen, Mingjun Zhao, Fanbo Meng, Yu Xu, Yancheng He, Baotian Hu, Haizhou Li, and Min Zhang. Hierarchical acoustic-semantic modeling: Modality separation and semantic coherence for full-duplex SLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the*

64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 9264–9280, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.419. URL https://aclanthology.org/2026.acl-long.419/.

Jingyu Lu, Yuhan Wang, Jianming Luo, Yifu Chen, Tianle Liang, Shengpeng Ji, Ziyue Jiang, Xiaoda Yang, Yu Zhang, Xize Cheng, et al. A survey of full-duplex spoken dialogue systems: Architectural hierarchy, interaction ontology, and decision state machine. *arXiv preprint arXiv:2606.19453*, 2026.

MiniCPM Team, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, et al. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*, 2025.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23555–23569, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/666c1861d709bd84e20b6e0e02a2c223-Paper-Conference.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, et al. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

Rajarshi Roy, Jonathan Raiman, Sang-Gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. Personaplex: Voice and role control for full duplex conversational speech models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16137–16141, 2026. doi: 10.1109/ICASSP55912.2026.11463413.

Chang Shi, Tim Pearce, Manan Tomar, Siddhartha Sen, and John Langford. Hierarchical latent prediction for language models. *arXiv preprint arXiv:2608.05806*, 2026.

Yi-Jen Shih, Desh Raj, Chunyang Wu, Wei Zhou, SK Bong, Yashesh Gaur, Jay Mahadeokar, Ozlem Kalinli, and Mike Seltzer. Can speech llms think while listening? In *International Conference on Learning Representations*, volume 2026, pp. 72392–72414, 2026.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47901–47911. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/tang24l.html.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S Hu, Tim Pearce, Pratyusha Sharma, Akshay Krishnamurthy, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models. *arXiv preprint arXiv:2511.05963*, 2025.

Chaoren Wang, Heng Lu, Xueyao Zhang, Shujie Liu, Yan Lu, Jinyu Li, and Zhizheng Wu. Closing the modality reasoning gap for speech large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pp. 18821–18835, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.857. URL https://aclanthology.org/2026.acl-long.857/.

Donghang Wu, Tianyu Zhang, Yuxin Li, Hexin Liu, Chen Chen, Eng Siong Chng, and Yoshua Bengio. The silent thought: Modeling internal cognition in full-duplex spoken dialogue models via latent reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=jzUCodmd4C.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025. URL https://arxiv.org/abs/2503.20215.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Maike Züfle, Ondrej Klejch, Nicholas Sanders, Jan Niehues, Alexandra Birch, and Tsz Kin Lam. F-actor: Controllable conversational behavior in full-duplex models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4904–4921, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.242. URL https://aclanthology.org/2026.findings-acl.242/.

A Details of Stream-Wise K/V Sparsity Analysis

We provide additional details for the stream-wise context dependence analysis in Figure 2. We analyze a separately parameterized F-Actor-based dual-stream checkpoint trained without bridge attention, evaluating the self-attention of the semantic and generative streams independently. Following the post-hoc pruning analysis of Tang et al. (2024), we evaluate one Transformer layer at a time while keeping all remaining layers dense. We partition historical key/value (K/V) states into non-overlapping blocks of 64 states, and evaluate each pruning policy over a fixed set of removal budgets. For attention-guided pruning, block importance is computed independently for each query–head pair by summing the dense attention probabilities within each K/V block, and blocks are removed in ascending order of this mass. Recency-based pruning instead removes blocks from oldest to newest.

For each stream, we measure pruning sensitivity using its corresponding training objective: $\mathcal{L}_{\mathrm{Sem}}$ for the semantic stream and $\mathcal{L}_{\mathrm{Gen}}$ for the generative stream. For each pruning budget b, we compare the corresponding loss $\mathcal{L}(b)$ with the dense attention loss $\mathcal{L}_{\mathrm{dense}}$ using

$$\Delta\mathcal{L}(b) = \frac{\mathcal{L}(b) - \mathcal{L}_{\mathrm{dense}}}{\mathcal{L}_{\mathrm{dense}}}. \tag{10}$$

We select the largest evaluated budget whose relative loss increase does not exceed 0.1%. If no non-zero budget satisfies this criterion, safe sparsity is set to zero.

For each budget, we measure sparsity as the fraction of causally visible K/V accesses removed across all queries and attention heads, rather than the fraction of unique K/V cache entries removed. We use an access-based measure because the same K/V state can be visible to many queries and heads, while pruning decisions are made separately for each query–head pair. We compute these access removal ratios once over the full evaluation set and keep them fixed for each budget. The safe sparsity reported in Figure 2 is therefore the access removal ratio associated with the largest budget satisfying the loss tolerance.

We estimate uncertainty using pointwise percentile bootstrap intervals. For each stream, layer, and pruning policy, we perform 1,000 bootstrap replicates with random seed 42, sampling 500 conversation samples with replacement. Within each replicate, the same sampled indices are used for the paired dense and pruned losses across all evaluated budgets. We recompute the relative loss increases and reselect the largest budget satisfying the 0.1% criterion, while keeping the access removal ratio associated with each budget fixed to its full-sample value. The 2.5th and 97.5th percentiles of the resulting safe sparsity values define the reported 95% confidence interval. This analysis characterizes within-stream dependence on historical K/V context independently of semantic-to-generative bridge attention.

B Experimental Details

B.1 Training and Evaluation Data

Training Data. Behavior-SD (Lee et al., 2025) serves as the task-specific full-duplex adaptation corpus for all Bridge-Duplex models in this study. We select Behavior-SD because its multi-turn, two-speaker conversations and explicit full-duplex behaviors closely match the interaction setting our framework targets. We keep the task-specific adaptation corpus fixed across Bridge-Duplex variants to focus the comparison on architectural differences rather than data-scaling effects. We follow the training and validation splits released with F-Actor (Züfle et al., 2026), containing 106,250 and 932 conversations, respectively. For the F-Actor-based instantiation, we use the public NanoCodec-preprocessed release[1]. For the Moshi-based instantiation, we use the corresponding Mimi-preprocessed release[2] as the basis for backbone-specific preprocessing. Both releases are derived from the same underlying Behavior-SD conversations, with backbone-specific codec representations and aligned metadata.

[1] https://huggingface.co/datasets/maikezu/f-actor-behavior-sd-nanocodec

[2] https://huggingface.co/datasets/maikezu/f-actor-behavior-sd-mimi

For the F-Actor-based instantiation, each speaker channel is represented by four NanoCodec (Casanova et al., 2025) code streams on a 12.5-Hz codec timeline together with frame-aligned, role-separated text. The narrative and annotated behavior counts are provided as separate conditioning inputs. The corresponding 192-dimensional ECAPA-TDNN system-speaker embedding (Dawalatabad et al., 2021) is provided as an additional conditioning input. For the Moshi-based instantiation, we derive frame-aligned role text from the same conversations and retain eight Mimi codebooks (Défossez et al., 2024) per speaker from the backbone-native acoustic representation. The Moshi-based instantiation uses these aligned text and acoustic streams but omits narrative and speaker-embedding conditioning.

Evaluation Data. For MTR-DuplexBench (MTR-DB) (He et al., 2026)[3], we use the dialogue-quality subset, containing 1,175 target turns from 200 dialogues. For Behavior-SD, we construct 539 target-response instances from 320 unique source recordings selected from the 925 recordings in the test split. We identify intermediate system turns bounded by user turns and retain targets with at least 5 seconds of speech, 12 words, and non-empty linguistic content. Instances containing interruptions, temporal overlap, or nested user backchannels are excluded. Both datasets are used for multi-turn evaluation with Dialogue Quality (DQ) and Context Consistency (CC).

Full-duplex interaction is evaluated using subsets from Full-Duplex-Bench v1.0 and v1.5 (Lin et al., 2025; 2026)[4]. Version 1.0 provides scenarios for pause handling, turn taking, backchanneling, and synthetic user interruption, whereas version 1.5 extends the evaluation to user interruption, user backchanneling, background speech, and talking-to-other conditions. These subsets evaluate whether the model appropriately initiates, continues, or stops speech under different full-duplex interaction conditions.

B.2 Implementation Details

B.2.1 Backbone-Specific Dual-Stream Parameterization

We instantiate Bridge-Duplex with two backbone-specific dual-stream parameterizations: independently parameterized Transformer paths and stream-specific LoRA adapters over a shared frozen backbone.

Separately Parameterized Dual-Stream Instantiation. For the F-Actor-based instantiation (Züfle et al., 2026), the semantic and generative streams are implemented as layer-aligned Transformer paths with independent parameters, each initialized from `meta-llama/Llama-3.2-1B-Instruct`[5] (Grattafiori et al., 2024). Each path contains 16 Transformer layers with hidden size 2,048, 32 query heads, and 8 K/V heads.

LoRA-Based Dual-Stream Instantiation. For the Moshi-based instantiation (Défossez et al., 2024), the pretrained backbone is shared and kept frozen, while the semantic and generative streams are specialized through stream-specific LoRA adapters. Our LoRA-based adaptation follows the open-source Moshi-Finetune framework[6], with separate adapters assigned to the semantic and generative streams. We apply rank-8 LoRA with scaling parameter $\alpha = 16$ and dropout 0.05 to the query, key, value, and output projection matrices in both the temporal Transformer and the DepFormer. The LoRA parameters are not shared between the semantic and generative streams.

Shared Dual-Stream Coupling. The dual-stream K/V organization is adapted from the open-source Mixture-of-Transformers implementation in Cosmos 3 (Agarwal et al., 2026)[7], with causal visibility and temporal alignment modified for our semantic–generative setting. The semantic memory index implementation is adapted from MiniMax Sparse Attention (Lai et al., 2026)[8].

[3] `https://huggingface.co/datasets/Jeff0918/MTR-DuplexBench`
[4] `https://github.com/DanielLin94144/Full-Duplex-Bench`
[5] `https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`
[6] `https://github.com/kyutai-labs/moshi-finetune`
[7] `https://github.com/nvidia/cosmos-framework`
[8] `https://github.com/MiniMax-AI/MSA`

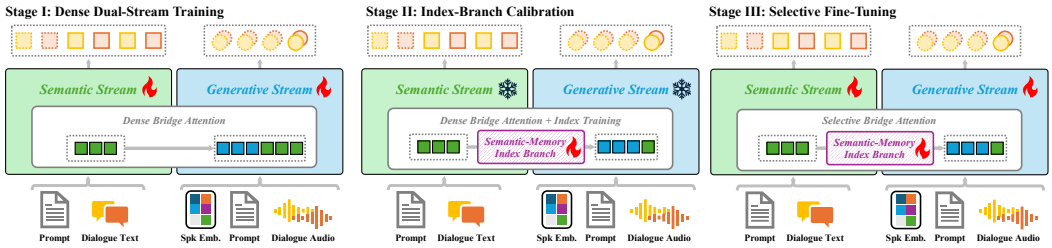


Figure 6: Three-stage optimization for the separately parameterized Bridge-Duplex instantiation.

Table 5: Stage-wise optimization settings for Bridge-Duplex across F-Actor and Moshi backbones.

Optimization Phase	Updates	Peak LR	Batch	Warm-up
F-Actor Backbone				
Stage I: Dense Dual-Stream Training	8,000	5×10^{-5}	32	1,908
Stage II: Index-Branch Calibration	2,000	5×10^{-5}	32	200
Stage III: Selective Fine-tuning	8,000	1×10^{-5}	32	500
Moshi Backbone				
Stage I: Dense Dual-Stream Training	30,000	3×10^{-5}	64	1,000
Stage II: Index-Branch Calibration	6,000	5×10^{-5}	64	600
Stage III: Selective Fine-tuning	9,000	1×10^{-5}	64	900

B.2.2 Training Configuration

Three-Stage Optimization. Figure 6 summarizes the dense-to-selective optimization procedure. Stage I trains the dense dual-stream model with $\mathcal{L}_{\text{Gen}}$ and $\mathcal{L}_{\text{Sem}}$. Stage II freezes the dual-stream model and trains the index branch with $\mathcal{L}_{\text{idx}}$. Stage III activates selective bridge attention and jointly fine-tunes the dual-stream model and index branch. For Moshi, only the stream-specific LoRA parameters are updated while the pretrained backbone remains frozen. Stage-specific settings are reported in Table 5. F-Actor and Moshi are trained on four NVIDIA A6000 GPUs with ZeRO-2 and eight NVIDIA A100 GPUs with ZeRO-3, respectively (Rasley et al., 2020).

Selective-Memory Configuration. Both backbone instantiations use eight selector groups with index-head dimension $d_{\text{idx}} = 64$. We set the semantic memory block size to $B_{\text{Sem}} = 64$ and retrieve $k = 4$ blocks per group, giving up to 256 selected semantic states per selector group. For the F-Actor-based instantiation, each generative hidden state produces eight 64-dimensional index queries, while a single 64-dimensional semantic index-key projection is shared across groups. Each of the eight K/V heads defines one selector group containing its four associated query heads. For the Moshi-based instantiation, we replace the token-wise max-pooled selector above with a backbone-specific block-level variant. The 32 attention heads form eight groups of four, with separate 64-dimensional index-query and index-key projections for each group. Semantic index keys are mean-pooled within each block before computing the group-specific query–block score. For $\mathcal{L}_{\text{idx}}$, the shared support is the full causal history in Stage II and the selected memory in Stage III, with main-attention probabilities renormalized over semantic memory.

Dialogue NextLat Configuration. We use a one-step prediction horizon, such that Dialogue next-latent prediction (NextLat) targets the next valid semantic update for each role. The transition predictor follows the residual MLP design of NextLat (Teoh et al., 2025)[9], with normalized inputs, bias-free projections, and an MLP expansion factor of 1.6. We set $\lambda_{\text{next-h}} = \lambda_{\text{KL}} = 1$.

Streaming-Text Augmentation. Following the user-stream delay augmentation used in KAME (Kuroki et al., 2026)[10], we augment only the ground-truth user text provided to the semantic stream. For each training sample, a delay is applied with probability 0.7, and a single offset is drawn uniformly from 1 to 6 codec frames and applied to the entire user-side semantic token stream. The same offset shifts the corresponding user-side semantic prediction targets to preserve temporal alignment. After the temporal shift, user-side input tokens are independently dropped with probability 0.1, while the prediction targets remain uncorrupted. System-side semantic inputs are not perturbed.

[9] https://github.com/JaydenTeoh/NextLat

[10] https://github.com/SakanaAI/kame

B.2.3 STREAMING INFERENCE CONFIGURATION

Streaming ASR Input. During inference, user speech is transcribed using the pretrained `nvidia/nemotron-3.5-asr-streaming-0.6b`[11] as a frozen frontend without task-specific fine-tuning. The model uses a 24-layer cache-aware FastConformer encoder (Rekesh et al., 2023) with a language-ID-conditioned RNN-T decoder. We fix the language prompt to `en-US` and use the zero-look-ahead configuration corresponding to the 80-ms streaming profile. Audio is processed sequentially in 80-ms chunks while maintaining a persistent streaming session that reuses the encoder cache throughout each utterance. Newly recognized words from each partial hypothesis are committed immediately without revision, lowercased, retokenized with the dialogue-model tokenizer, and passed to the semantic stream. Each ASR token is assigned the codec-frame position at which it becomes available to the model, rather than its earlier acoustic alignment. At frame t, bridge attention can use semantic states available by that frame, including user text delivered by streaming ASR at t; system text generated at t becomes available only to subsequent generative queries.

Cached Decode Execution. Selective Bridge-Duplex maintains separate compact semantic and generative K/V caches while tracking their original codec-frame positions for positional encoding and causal visibility. For each K/V head, the selected semantic states are gathered and concatenated with the visible generative K/V states before PyTorch scaled dot-product attention (SDPA). A grouped-query packed layout allows query heads associated with the same K/V head to share this K/V sequence without explicit replication (Lai et al., 2026). When the visible semantic history contains at most 256 states, selection is bypassed and all semantic states are used directly. A fast execution path additionally skips inactive stream modules at each decoding step, avoiding padded Boolean routing and output scatter.

Table 6: Implementation sources and pretrained checkpoints used for the comparison models.

Model	Implementation Source	Checkpoint
Conventional Speech-to-Speech		
Moshi (Défossez et al., 2024)	`https://github.com/kyutai-labs/moshi`	`kyutai/moshika-pytorch-bf16`
F-Actor (Züfle et al., 2026)	`https://github.com/MaikeZuefle/f-actor`	`maikezu/f-actor`
PersonaPlex (Roy et al., 2026)	`https://github.com/NVIDIA/personaplex`	`nvidia/personaplex-7b-v1`
Externally Augmented Speech-to-Speech		
KAME (Kuroki et al., 2026)	`https://github.com/SakanaAI/kame`	`SakanaAI/kame`
MoshiRAG (Chien et al., 2026)	`https://github.com/kyutai-labs/moshi-rag`	`kyutai/moshika-rag-pytorch-bf16`

B.3 COMPARISON MODEL IMPLEMENTATIONS

For published comparison systems, we use the released implementations and pretrained checkpoints summarized in Table 6. This includes Moshi (Défossez et al., 2024), F-Actor (Züfle et al., 2026), PersonaPlex (Roy et al., 2026), KAME (Kuroki et al., 2026), and MoshiRAG (Chien et al., 2026). Unless otherwise noted, each system is evaluated using its released inference configuration.

In addition, we construct two F-Actor-based controls to compare Bridge-Duplex with alternative forms of semantic augmentation under matched training conditions.

Oracle Conditioning. Following the simulated-oracle setting of KAME (Kuroki et al., 2026), we augment F-Actor with semantic guidance generated by `Qwen/Qwen3.6-35B-A3B-FP8`[12]. The guidance is generated from the partial dialogue history and reference next response, and updated every 0.5 seconds as additional user context becomes available. Only guidance targeting the current system role is retained and provided as conditioning to F-Actor. During training, guidance events are randomly omitted and temporally shifted to improve robustness to missing or misaligned guidance.

Latent Reasoning. Following FLAIR (Wu et al., 2026), we train the causal F-Actor backbone with guidance from a non-causal T5-Large expert (Raffel et al., 2020). At listening frames, the expert produces a vocabulary distribution from the full dialogue context, which is converted into a soft text embedding. F-Actor is trained to match the expert distribution using KL divergence while retaining its standard generation objectives. At inference, the expert is removed and the model recursively feeds its predicted soft text embedding into the next listening step.

[11] `https://huggingface.co/nvidia/nemotron-3.5-asr-streaming-0.6b`
[12] `https://huggingface.co/Qwen/Qwen3.6-35B-A3B-FP8`

B.4 Evaluation Protocols

Multi-Turn Semantic Evaluation. We follow the turn-by-turn evaluation protocol of MTR-DB (He et al., 2026) for both MTR-DB and Behavior-SD. For each target system turn, the preceding user and system turns are provided from the reference dialogue, and the model generates only the current response. Generated responses are not propagated to subsequent targets, preventing accumulated generation errors across evaluated turns.

Generated speech is transcribed using `nvidia/parakeet-tdt-0.6b-v2`[13] before text-based evaluation. If no speech is produced or the ASR transcript is empty, the target is assigned DQ $= 0$ and CC $= 0$ without invoking the LLM judge. DQ follows MTR-DB (He et al., 2026) and evaluates the meaningfulness of the target response. For CC, we adapt the context-consistency criterion from M3-DuplexBench (Fukuda et al., 2026) and evaluate whether relevant information from the preceding dialogue is preserved and correctly used. Both metrics receive the same timestamped two-speaker transcript containing the preceding dialogue and the generated target response, together with the interval corresponding to the target response. The complete DQ and CC prompts are shown in Figure 10.

We use `Qwen/Qwen3-14B`[14] (Yang et al., 2025) as the LLM judge, with thinking disabled and sampling temperature 0.7. For each target turn, we run the judge three times and average the resulting turn-level scores. For DQ, each run assigns a score from 0 to 5 in increments of 0.5, and the final DQ is averaged over all target turns. For CC, the same score range is used, and the final CC is averaged over all target turns.

Two-Instance Dialogue Evaluation. Following F-Actor (Züfle et al., 2026), we evaluate multi-turn dialogue generation using two interacting instances of the same model. The two instances receive distinct personalized instructions, and each generated response is passed to the other instance as the subsequent user-side interaction. Unlike the turn-by-turn protocol, generated responses propagate throughout the interaction, and the resulting dialogue is evaluated as a whole.

DQ evaluates the overall meaningfulness and clarity of the generated interaction. CC evaluates whether information and dialogue state established in earlier turns are preserved and correctly used in later turns. We use the same Qwen3-14B judge, decoding configuration, score ranges, and three-run evaluation as above, with setting-specific full-dialogue prompts. The complete two-instance DQ and CC prompts are shown in Figure 11.

Full-Duplex Interaction. We follow the official evaluation protocols of Full-Duplex-Bench v1.0 and v1.5 (Lin et al., 2025; 2026) and report takeover rate (TOR), Respond, Stop Latency, and Jensen–Shannon Divergence (JSD). For turn-taking, TOR is measured on the Candor turn-taking subset of v1.0, where a takeover denotes model speech that is neither silence nor a backchannel. For user interruption, Respond and Stop Latency are measured on the v1.5 user-interruption subset. Respond measures whether the model addresses the interrupting query rather than continuing its previous response. Generated speech is transcribed using `nvidia/parakeet-tdt-0.6b-v2` and evaluated with `meta-llama/Llama-3.1-8B-Instruct`[15], following the official evaluation procedure for Respond. Stop Latency measures the interval from interruption onset to termination of the ongoing system speech. For backchanneling, JSD is measured on the ICC backchannel subset of v1.0 and quantifies the divergence between model and human backchannel timing distributions, considering only trials in which the model does not take over the turn.

End-to-End Generation RTF. We measure end-to-end generation real-time factor (RTF) per sample as elapsed wall-clock time divided by the generated speech duration. The generated duration is derived from valid acoustic outputs using each model's native temporal resolution. Timing begins after device synchronization immediately before input preparation and includes preprocessing, online frontend processing when required, prefill, autoregressive generation, sampling, output materialization, and final device synchronization. The first sample is excluded as warm-up, and the reported RTF is the mean over the remaining samples.

[13] `https://huggingface.co/nvidia/parakeet-tdt-0.6b-v2`
[14] `https://huggingface.co/Qwen/Qwen3-14B`
[15] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

Table 7: LLM judges and pretrained checkpoints used for multi-turn semantic evaluation.

LLM Judge	Checkpoint
Qwen3-14B	`Qwen/Qwen3-14B`
Qwen3-32B	`Qwen/Qwen3-32B`
Qwen2.5-14B-Instruct	`Qwen/Qwen2.5-14B-Instruct`
Llama-3.1-8B-Instruct	`meta-llama/Llama-3.1-8B-Instruct`
Mistral-Nemo-12B-Instruct	`mistralai/Mistral-Nemo-Instruct-2407`

Table 8: Consistency of multi-turn semantic results across LLM judges. DQ and CC are computed by five different LLM judges on the same generated outputs.

LLM Judge	Model	*MTR-DB*		*Behavior-SD*	
		DQ ↑	CC ↑	DQ ↑	CC ↑
Qwen3-14B	Moshi	1.99	2.64	2.91	3.49
	F-Actor	2.73	3.90	3.51	4.13
	PersonaPlex	2.09	2.71	3.36	4.06
	KAME	2.65	3.10	2.67	2.34
	MoshiRAG	2.52	3.11	2.90	3.48
	Bridge-Duplex (F-Actor)†	**3.03**	4.09	3.68	4.31
	Bridge-Duplex (Moshi)‡	3.01	**4.34**	**3.82**	**4.59**
Qwen3-32B	Moshi	1.95	2.78	2.89	3.35
	F-Actor	2.48	3.24	3.29	3.91
	PersonaPlex	2.11	2.89	3.33	3.90
	KAME	2.50	3.35	2.33	2.71
	MoshiRAG	2.43	3.24	2.89	3.48
	Bridge-Duplex (F-Actor)†	2.76	3.48	3.47	4.22
	Bridge-Duplex (Moshi)‡	**2.85**	**3.76**	**3.65**	**4.32**
Qwen2.5-14B-Instruct	Moshi	2.16	2.86	3.44	3.38
	F-Actor	3.22	3.48	4.15	4.10
	PersonaPlex	2.38	2.96	3.94	3.93
	KAME	2.85	3.27	3.04	2.92
	MoshiRAG	2.73	3.21	3.26	3.42
	Bridge-Duplex (F-Actor)†	3.60	3.64	**4.31**	4.28
	Bridge-Duplex (Moshi)‡	**3.69**	**4.02**	4.30	**4.31**
Llama-3.1-8B-Instruct	Moshi	2.16	2.57	2.50	2.90
	F-Actor	2.65	2.73	3.31	3.44
	PersonaPlex	2.17	2.65	2.92	3.39
	KAME	2.52	2.97	2.68	2.83
	MoshiRAG	2.41	2.89	2.64	2.97
	Bridge-Duplex (F-Actor)†	2.53	2.85	**3.39**	**3.66**
	Bridge-Duplex (Moshi)‡	**2.76**	**3.15**	3.04	3.41
Mistral-Nemo-12B-Instruct	Moshi	2.31	3.10	2.98	3.21
	F-Actor	2.79	3.37	3.69	3.85
	PersonaPlex	2.40	3.10	3.48	3.75
	KAME	2.87	3.49	2.97	3.53
	MoshiRAG	2.72	3.38	2.98	3.33
	Bridge-Duplex (F-Actor)†	2.92	3.41	**3.75**	3.95
	Bridge-Duplex (Moshi)‡	**3.17**	**3.85**	3.58	**4.01**

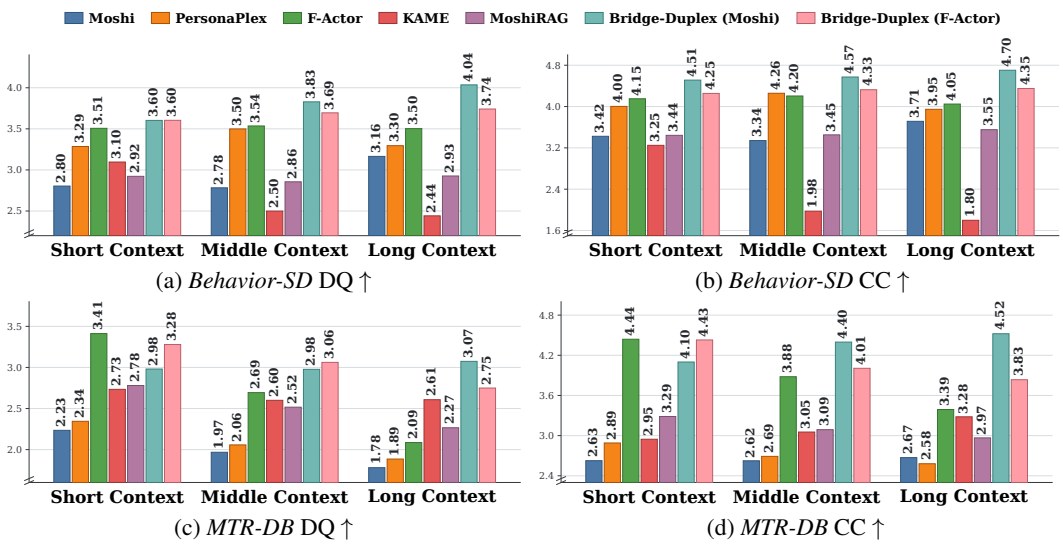
† Separately parameterized dual-stream; ‡ LoRA-based dual-stream.

C Additional Analyses

C.1 Robustness Across LLM Judges

We assess whether the multi-turn semantic results in Table 1 are consistent across different LLM judges. The same generated outputs are re-evaluated with five instruction-tuned LLMs while keeping the evaluation inputs, DQ and CC criteria, prompts, and score aggregation procedure fixed. The evaluated judges and checkpoints are listed in Table 7.

Table 8 shows that absolute DQ and CC scores vary across judges. Despite these differences, the overall improvement trend remains consistent across judges, with Bridge-Duplex improving over its corresponding backbone in nearly all evaluated dataset–metric combinations. The only exception is MTR-DB DQ under Llama-3.1-8B-Instruct for the F-Actor-based instantiation.



(a) *Behavior-SD* DQ ↑ (b) *Behavior-SD* CC ↑

(c) *MTR-DB* DQ ↑ (d) *MTR-DB* CC ↑

Figure 7: Context-dependent dialogue quality and consistency across full-duplex S2S systems, with target turns grouped by the number of preceding semantic-memory blocks.

C.2 Context-Length Analysis Across Systems

Figure 7 further breaks down the multi-turn semantic results in Table 1 by preceding context length. For each dataset, target turns are divided into three equally sized Short, Middle, and Long Context groups according to the number of semantic memory blocks induced from the preceding reference context. The same partition is applied to all systems within each dataset, and DQ and CC are averaged separately within each group.

Across Behavior-SD, both Bridge-Duplex instantiations improve over their corresponding backbones across all context-length groups. On MTR-DB, the gains are smaller in the Short Context group, with the F-Actor-based instantiation showing a slight decrease in DQ, but become more pronounced in the Middle and Long Context groups. These results complement the aggregate improvements in Table 1, showing that the relative benefit of Bridge-Duplex becomes more apparent as preceding dialogue context grows. This pattern is consistent with the intended role of persistent semantic memory in maintaining information across longer dialogue histories.

C.3 Effect of Streaming-Text Augmentation

To assess the effect of streaming-text augmentation, we compare Dense Bridge-Duplex trained with and without augmentation, keeping all other training and inference settings fixed. We evaluate both variants under the two-instance dialogue setting, where generated responses propagate across turns and are observed through the streaming ASR frontend as subsequent user-side semantic input. This setting allows ASR errors to accumulate across turns, as each generated response is retranscribed and used as subsequent input.

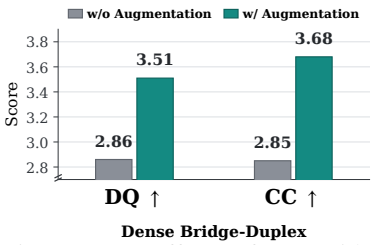
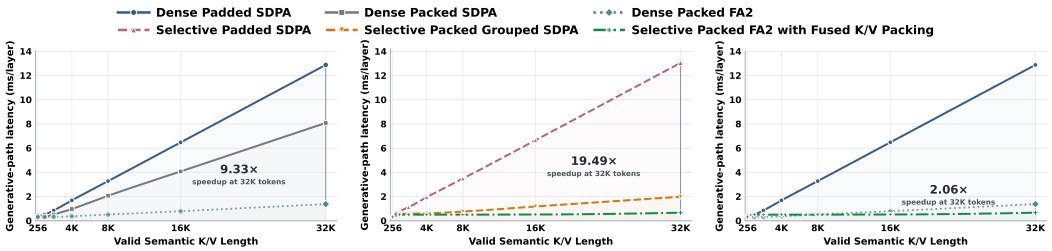


Figure 8: Effect of user-side streaming-text augmentation on Two-Instance Dialogue evaluation with causal streaming ASR.

As shown in Figure 8, removing the augmentation decreases DQ from 3.51 to 2.86 and CC from 3.68 to 2.85. The larger reduction in CC suggests that exposure to delayed and incomplete user text observations during training is particularly important for preserving contextual information under causal streaming ASR. These results indicate that streaming-text augmentation helps reduce the mismatch between ground-truth user text available during training and the evolving causal text observations available to the persistent semantic stream at inference.



(a) Dense execution paths (b) Selective execution paths (c) Optimized dense vs. selective

Figure 9: Generative attention-path efficiency comparison under a shared cached-decoding configuration. FA2 denotes FlashAttention-2.

C.4 Generative Attention-Path Scaling under Cached Decoding

Experimental Setup. While the main experiments report end-to-end RTF during dialogue generation, we further isolate the generative attention path to examine how its latency scales with increasing semantic history under cached decoding. Figure 9 measures a single cached generative query at one decoder layer as the visible semantic history increases from 256 to 32K states. The measurement includes K/V preparation and attention for Dense Bridge-Duplex, and additionally includes selector query projection, index search, block Top-k, and selected K/V packing for Selective Bridge-Duplex. The remaining Transformer computation is excluded, so the reported latency does not represent complete-decoder or end-to-end inference.

For Dense Bridge-Duplex, we compare padded and packed execution using PyTorch scaled dot-product attention (SDPA), together with FlashAttention-2 (FA2) (Dao, 2024) as an optimized dense-attention reference. Dense–Padded SDPA retains the codec-timeline layout and masks invalid semantic positions, whereas Dense–Packed SDPA removes these inactive positions while retaining all causally valid semantic and visible generative K/V states. Dense–Packed FA2 uses the same packed K/V support and changes only the attention backend. For Selective Bridge-Duplex, all conditions use the same learned retrieval policy and therefore the same selected semantic support. Selective–Padded SDPA masks unselected semantic states without reducing the executed K/V length, whereas Selective–Packed Grouped SDPA physically gathers only the retrieved semantic blocks. Selective–Packed FA2 with Fused K/V Packing preserves the same selected support but combines semantic K/V gathering and visible generative K/V copying into a single packing operation before standard FA2 attention. For histories longer than 256 semantic states, four 64-state blocks are retrieved per K/V head; selection is bypassed at or below 256 states.

Scaling Behavior. Figure 9a shows that Dense Bridge-Duplex benefits from both physical K/V packing and an optimized attention backend. At 32K semantic states, latency decreases from 12.88 ms/layer with Dense–Padded SDPA to 8.08 ms/layer with Dense–Packed SDPA and 1.38 ms/layer with Dense–Packed FA2. For Selective Bridge-Duplex, Figure 9b shows that masking alone provides little benefit when unselected states remain in the executed tensors. Physical packing reduces latency from 13.05 ms/layer with Selective–Padded SDPA to 1.99 ms/layer with Selective–Packed Grouped SDPA, and FA2 with Fused K/V Packing further reduces it to 0.67 ms/layer. These results show that selective semantic access must be accompanied by physical K/V reduction and an efficient attention backend to translate reduced support into wall-clock savings.

Figure 9c directly compares the optimized dense and selective execution paths. At 32K semantic states, Selective–Packed FA2 with Fused K/V Packing achieves 2.06× lower latency than Dense–Packed FA2. This difference should not be attributed to Fused K/V Packing alone, since the selective path also restricts semantic K/V access to the retrieved support. At the same time, the current Selective–Packed Grouped SDPA implementation remains slower than Dense–Packed FA2 at 32K states, requiring 1.99 ms/layer compared with 1.38 ms/layer. This distinction shows that selective retrieval and execution-backend efficiency are separate axes of optimization. Recent systems such as MiniCPM 4 (MiniCPM Team et al., 2025) combine sparse attention with dedicated CUDA kernels and inference runtimes, suggesting further opportunities to translate selective access into lower wall-clock latency. Developing kernels specialized for the group-dependent selective K/V access pattern of Bridge-Duplex remains an important direction for future work.

Target-Turn Dialogue Quality Prompt

{STEREO_TRANSCRIPT}

Please evaluate the following two-speaker dialogue transcript for how meaningful the speech is (based on its content), only focusing on the model channel's output from {ASSISTANT_TURN_START} to {ASSISTANT_TURN_END} seconds.

Use the following scale:
0: Completely meaningless; no coherent sentences, random words, or unintelligible.
0.5: Almost no meaning; isolated words or phrases, but no understandable ideas.
1: Extremely low meaning; rare, vague fragments of ideas, but mostly incoherent or off-topic.
1.5: Very little meaning; a few short, unclear ideas, but mostly disjointed or confusing.
2: Low meaning; some recognizable ideas or topics, but mostly unclear, incomplete, or off-topic.
2.5: Somewhat low meaning; a few coherent points, but overall lacks clarity or logical flow.
3: Moderate meaning; general topic is understandable, but there are gaps, unclear parts, or weak connections.
3.5: Fairly meaningful; mostly coherent and relevant, but with some confusion, repetition, or lack of detail.
4: Meaningful; clear and logical, with relevant and connected ideas, though may lack depth or detail.
4.5: Very meaningful; almost fully coherent, with well-developed, relevant, and connected ideas.
5: Extremely meaningful; highly coherent, clear, and detailed, with all ideas well connected and relevant.

Only output the final score (0, 0.5, 1, 1.5, ..., 5) ONLY according to the above rubric. Do not output anything else.

(a) Prompt for target-turn DQ evaluation.

Target-Turn Context Consistency Prompt

{STEREO_TRANSCRIPT}

Please evaluate CONTEXT CONSISTENCY only, focusing on the model channel's output from {ASSISTANT_TURN_START} to {ASSISTANT_TURN_END} seconds and comparing it with the dialogue history before that interval.

Core question: Does this generated model utterance remember and use earlier dialogue information without contradiction?

Judge preservation of persistent semantic state such as facts, entities, user preferences, constraints, decisions, commitments, and dialogue-state updates. Do not reward a response merely for being meaningful; score only whether it preserves and correctly uses relevant prior context.

Use the following scale:
0: Completely inconsistent; directly contradicts or entirely forgets indispensable prior context.
0.5: Almost completely inconsistent; preserves almost none of the required context.
1: Extremely inconsistent; major facts, preferences, constraints, decisions, or commitments are lost or contradicted.
1.5: Very inconsistent; multiple important context errors substantially damage the response.
2: Inconsistent; some relevant context is recognized, but major omissions or contradictions remain.
2.5: Somewhat inconsistent; partially preserves context but has a significant error or underspecification.
3: Moderately consistent; generally follows prior context but has noticeable omissions or weak use of it.
3.5: Fairly consistent; preserves most relevant context with only a minor error or omission.
4: Consistent; correctly preserves and uses relevant prior context with at most small imperfections.
4.5: Very consistent; nearly complete and accurate use of all relevant prior context.
5: Fully consistent; preserves and correctly uses all relevant facts, preferences, constraints, decisions, commitments, and state updates without contradiction.

Only output one final scalar from 0, 0.5, 1, 1.5, ..., 5. Do not output JSON, an explanation, or any other text.

(b) Prompt for target-turn CC evaluation.

Figure 10: Prompt templates used for target-turn semantic evaluation.

Two-Instance Full-Dialogue Dialogue Quality Prompt

{STEREO_TRANSCRIPT}

Please evaluate the following two-speaker dialogue transcript for the overall meaningfulness and quality of the generated interaction. Consider the full interaction from beginning to end, while judging each exchange primarily for whether it is understandable, meaningful, locally coherent, and appropriately responsive.

Use the following scale:
0: Completely meaningless; no coherent exchange, understandable sentences, or recognizable conversational structure.
0.5: Almost no meaning; isolated words or phrases appear, but there is no understandable dialogue or exchange of ideas.
1: Extremely low meaning; rare and vague fragments of ideas are present, but the dialogue is mostly incoherent, off-topic, or locally unresponsive.
1.5: Very little meaning; a few short and partially understandable ideas appear, but most exchanges are disjointed, confusing, or unresponsive.
2: Low meaning; some recognizable ideas, topics, or responses are present, but many exchanges are unclear, incomplete, repetitive, or poorly responsive.
2.5: Somewhat low meaning; several turns are understandable and locally coherent, but many exchanges remain unclear or weakly responsive.
3: Moderate meaning; the overall interaction is understandable, but there are noticeable unclear turns, weak local responses, repetition, or abrupt topic shifts.
3.5: Fairly meaningful; most exchanges are understandable, relevant, and locally coherent, but some confusion, repetition, weak responses, or awkward transitions remain.
4: Meaningful; the interaction is clear and logical, with most exchanges relevant and locally responsive, though it may lack depth, specificity, or detail.
4.5: Very meaningful; nearly all exchanges are clear, well developed, relevant, responsive, and locally coherent, with only minor imperfections.
5: Extremely meaningful; the interaction is highly clear, meaningful, and locally coherent throughout, with each exchange understandable and appropriately responsive to its immediate conversational context.

Only output the final score (0, 0.5, 1, 1.5, ..., 5) ONLY according to the above rubric. Do not output anything else.

(a) Prompt for two-instance DQ evaluation.

Two-Instance Full-Dialogue Context Consistency Prompt

{STEREO_TRANSCRIPT}

Please evaluate CONTEXT CONSISTENCY across the entire two-speaker dialogue transcript.
Consider how well information and dialogue state established in earlier turns are preserved, updated, and correctly used in later turns throughout the interaction.

Core question: Does the dialogue preserve and correctly use relevant earlier information and dialogue state in later turns without contradiction, stale-context reuse, or loss of relevant context?

Judge preservation of persistent semantic state such as facts, entities, user preferences, constraints, decisions, commitments, and dialogue-state updates. Do not reward a response merely for being meaningful; score only whether it preserves and correctly uses relevant prior context.

Use the following scale:
0: Completely inconsistent; later turns directly contradict or entirely forget indispensable information or dialogue state established earlier in the conversation.
0.5: Almost completely inconsistent; the dialogue preserves almost none of the relevant earlier context and contains severe contradictions or context-tracking failures.
1: Extremely inconsistent; major facts, entities, preferences, constraints, decisions, commitments, referents, or dialogue-state updates are repeatedly lost, contradicted, or incorrectly reused.
1.5: Very inconsistent; multiple important context-tracking errors substantially disrupt the correct use of earlier information.
2: Inconsistent; some relevant earlier context is recognized, but major omissions, contradictions, stale-context reuse, or failures to update dialogue state remain.
2.5: Somewhat inconsistent; the dialogue partially preserves prior context but contains at least one significant contradiction, omission, or incorrect use of earlier information.
3: Moderately consistent; the dialogue generally preserves relevant prior context but contains noticeable omissions, weak contextual grounding, or inconsistent use of earlier information.
3.5: Fairly consistent; most relevant context is preserved and correctly used, with only minor omissions, stale references, or context-tracking errors.
4: Consistent; relevant facts, entities, preferences, constraints, commitments, and dialogue-state updates are correctly preserved and used across the dialogue, with at most small imperfections.
4.5: Very consistent; the dialogue demonstrates nearly complete and accurate preservation, updating, and use of relevant prior context across turns.
5: Fully consistent; all relevant facts, entities, preferences, constraints, decisions, commitments, referents, and dialogue-state updates are preserved, appropriately updated, and correctly used throughout the dialogue without contradiction.

Only output one final scalar from 0, 0.5, 1, 1.5, ..., 5. Do not output JSON, an explanation, or any other text.

(b) Prompt for two-instance CC evaluation.

Figure 11: Prompt templates used for two-instance dialogue evaluation.