EGOMEM: LIFELONG MEMORY AGENT FOR FULL-DUPLEX OMNIMODAL MODELS

Anonymous authors
Paper under double-blind review

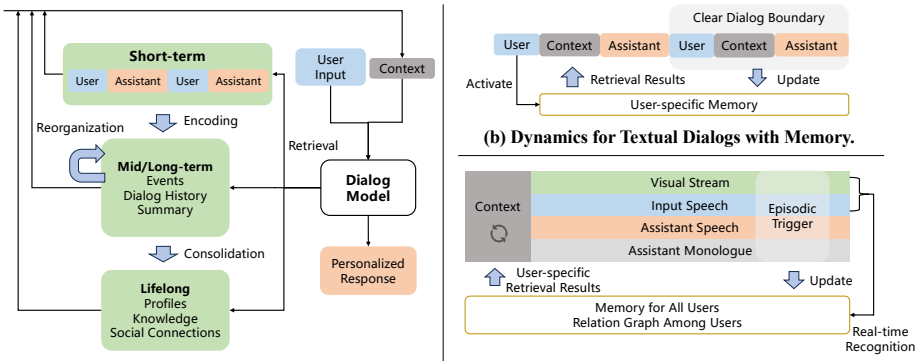
ABSTRACT

We introduce EgoMem, the first lifelong memory agent tailored for full-duplex models that process real-time omnimodal streams. EgoMem enables real-time models to recognize different users from raw audiovisual streams, to provide personalized response, and to maintain long-term knowledge of users' facts, preferences, and social relationships extracted from audiovisual history. EgoMem operates with three asynchronous processes: (i) a *retrieval* process that dynamically identifies user via face and voice, and gathers relevant context from a long-term memory; (ii) an *omnimodal dialog* process that generates personalized audio responses based on the retrieved context; and (iii) a *memory management* process that automatically detects dialog boundaries from omnimodal streams, and extracts necessary information to update the long-term memory. Unlike existing memory agents for LLMs, EgoMem relies entirely on raw audiovisual streams, making it especially suitable for lifelong, real-time, and embodied scenarios. Experimental results demonstrate that EgoMem's retrieval and memory management modules achieve over 95% accuracy on the test set. When integrated with a fine-tuned RoboEgo omnimodal chatbot, the system achieves fact-consistency scores above 87% in real-time personalized dialogs, establishing a strong baseline for future research.

1 INTRODUCTION

A wide range of AI applications involve lifelong omnimodal streams. A notable example is a robot deployed in homes and public spaces (AgiBot-World-Contributors et al., 2025; Bu et al., 2025). In similar scenarios, the models are required not only to follow instructions swiftly, but also to recognize users, remember their histories, understand social relationships, and deliver personalized services. Technically, the crucial capabilities to meet these requirements include *omnimodality*, *real-time responsiveness*, and *humanoid cognition* (Wang & Sun, 2025). For *real-time responsiveness*, there have been solutions to achieve full duplexity, either based on time-division multiplexing (Wang et al., 2024; Zhang et al., 2024b), or on native duplex (Défossez et al., 2024; Yao et al., 2025a) schemes. Yet, *humanoid cognition* remains an underexplored capability for current omnimodal, full-duplex systems. In this work, we study the lifelong memory capability as a critical step towards *humanoid cognition*, since memory is the foundation of both human and advanced artificial intelligence (Jimenez Gutierrez et al., 2024). We focus on real-time personalized dialog as a major task to validate the effectiveness of lifelong memory in omnimmodal scenarios.

We showcase the role of lifelong memory in personalized omnimodal dialogs as follows. (1) When a user Emily shows up, a polling process detects the user identity as Emily directly from the audiovisual stream (*e.g.,* camera and microphone inputs); (2) The profile of Emily is encoded and put into the dialog context of an omnimodal chatbot; (3) When Emily asks "does any of my colleagues love tennis?", a query regarding the relation "colleague" and keyword "tennis" is generated by the chatbot, activating a textual retrieval to the knowledge base containing Emily's social relation graph, which returns a dialog record of "John, colleague, 2024-05-13, user discussed a tennis game he played 2 days ago". This record is further encoded as dialog context; (4) The chatbot answers "Yes, Emily, your colleague John loves tennis" based on the available context; (5) The system extracts user facts: "Emily shows interest in tennis", and dialog record "2024-05-14, user asked if any of her colleagues loves tennis.", from the raw audiovisual stream of the recent dialog, and updates Emily's profile memory with these contents for future use; (6) When Emily shows up on another day, the model is able to greet with: "Hi Emily, did you talk to John about tennis?".



(a) Illustration of Humanoid Memory Agent Hierarchy.

(b) Dynamics for Textual Dialogs with Memory.

(c) Dynamics for Real-time, Omnimodal Dialogs with Memory.

Figure 1: Textual memory agents vs. full-duplex omnimodal memory agents (ours).

In literature, two primary approaches have been explored to equip textual large language models (LLMs) with long-term memory: extended context windows (Su et al., 2021; Wu et al., 2024) and memory agents (Zhong et al., 2024; Chhikara et al., 2025; Xu et al., 2025b; Kang et al., 2025). However, neither method transfers well to *lifelong omnimodal* scenarios. On one hand, extended context windows can retain long sequences encoding full omnimodal information (He et al., 2024). Yet, in lifelong settings, the length of audiovisual streams grows without bound, making even million-token contexts insufficient (Men et al., 2024). On the other hand, memory agent methods (Figure 1 (a)) are well-suited for lifelong operation (Lee et al., 2024; Wang & Chen, 2025; Li et al., 2025b), but typically rely on several strong assumptions: user identities are explicitly known, dialog sessions have clear boundaries, and all inputs are textual (Figure 1 (b)). Unfortunately, these assumptions do not hold in full-duplex omnimodal applications (Défossez et al., 2024; Zhang et al., 2024a; Lin et al., 2025), in which the user identities are implicitly encoded in audiovisual streams, and there is no well-defined boundaries for dialog turns or user sessions (Figure 1 (c)). Furthermore, existing memory agents generally overlook the multiuser social relation graph (Au et al., 2025), which is an important element for humanoid cognition in lifelong scenarios.

To address these issues, we propose EgoMem, the first lifelong memory system tailored for omnimodal scenarios and designed to facilitate full-duplex personalized dialog. EgoMem operates through three asynchronous processes. First, a **Retrieval Process** is responsible for real-time polling recognition of users, implemented with an audiovisual retrieval mechanism. It also contains a content-driven text retrieval module to gather related textual documents. This process facilitates efficient integration of both user-specific and RAG-style (Gao et al., 2023) information into the dialog flow. Second, an **Omnimodal Dialog Process** uses a fine-tuned dialog model to deliver full-duplex, personalized responses in real time, grounded in the retrieved context. Third, a **Memory Management Process** handles dialog boundary detection, information extraction, and memory updating, based on real-time raw audiovisual streams. This process ensures up-to-date memory over time.

We integrate EgoMem memory system to RoboEgo (Yao et al., 2025b), a *native* full-duplex model that is best aligned with our target scenarios. We fine-tune RoboEgo under our EgoMem framework to deliver real-time, lifelong, and personalized responses to arbitrary user. We conduct automated evaluations across the audio, textual, and visual retrieval modules, the memory management module, and the system's personalization abilities, considering either single-user profile (Level-1) and multi-user social graph (Level-2) as reference contexts. Evaluation results show that these modules exhibit high accuracy and robustness, and that the incorporation of EgoMem enhances personalization without compromising RoboEgo's original dialog capabilities.

Our contributions are as follows: (1) *framework*: we propose EgoMem, a lifelong memory agent for full-duplex, omnimodal interaction, which to the best of our knowledge is the first of its kind; (2) *implementation*: we provide a concrete implementation of EgoMem based on the RoboEgo backbone, including detailed module designs, data construction pipelines, and training configurations; (3) *evaluation*: we demonstrate that EgoMem achieves robust performance on personalization tasks in lifelong omnimodal scenarios, establishing a solid baseline for future research.

2 PRELIMINARIES

2.1 FULL-DUPLEX OMNIMODAL MODELS

Full-duplex omnimodal models are able to process real-time audiovisual inputs and demonstrate capabilities to listen and speak simultaneously. In each time step t, a full-duplex omnimodal model F takes as input a listening audio a_t, video frames v_t, and optionally a textual input l_t. It generates a slice of spoken audio response r_t:

$$r_t = F_\theta(a_t, v_t, l_t). \tag{1}$$

We adopt RoboEgo (Yao et al., 2025b) as our primary dialog model, as it supports a *native* full-duplex scheme at least for audio. The *native* scheme features lower response latency and better scalability (Défossez et al., 2024; Lin et al., 2025; Yao et al., 2025a), compared to time-division multiplexing (TDM) schemes. Also, in general instruction-following tasks, RoboEgo's response quality and user experiences are comparable to state-of-the-art systems such as Qwen-2.5-Omni (Xu et al., 2025a).

In RoboEgo, both the listening and speaking stream are processed with a frame rate of 12.5 fps, each frame corresponding to one autoregressive forward step t. In each step, 17 tokens are merged into one embedding: the listening and speaking audio frame are both encoded by 8 tokens, and the text channel contributes 1 token. Please refer to Appendix A for more details on the model's structure and stream organization. Note that EgoMem's framework and methodology can be applied to other full-duplex omnimodal models F or to different organizations of a_t, v_t, and l_t beyond our implementation.

2.2 MEMORY AGENT PARADIGM

EgoMem is designed to facilitate full-duplex, personalized chat for lifelong-deployed omnimodal models. As a first step, we focus on the case where there is only one active speaker at a time, leaving the more complex cocktail party problem (Haykin & Chen, 2005) for future work. In this setting, in each time step t, the main dialog model F takes two additional inputs: the user profile p_t, and reference information c_t:

$$r_t = F_\theta(a_t, v_t, l_t, p_t, c_t). \tag{2}$$

Here, p_t and c_t are encoded in the text channel, commonly referred to as the *context* or *short-term memory* in current literature, delivered to F as part of its KV-cache (Vaswani et al., 2017; Dao, 2023).

EgoMem manages a textual memory M with three core functions: *retrieval*, *writing*, and *updating*.

Retrieval. The retrieval function provides p_t and c_t to the dialog model by searching related information in M based on current dialog content:

$$p_t, c_t = \textsf{EgoMem.retr}(a_t, v_t, M). \tag{3}$$

In traditional textual memory agents (Gao et al., 2023), p_t is naively accessible from user accounts, while the retrieval process for c_t is always activated after the user's textual input. In lifelong omnimodal scenarios, however, both user identities and dialog boundaries are implicit. The memory agent should directly detect user identities and session boundaries from raw audiovisual streams.

Writing. The writing function extracts important events from lifelong multimodal streams and stores them in the memory unit M:

$$Episode = \textsf{EgoMem.extract}(a_{0\sim t}, v_{0\sim t}, l_{0\sim t}), \tag{4}$$

$$M \leftarrow \textsf{EgoMem.write}(M, Episode). \tag{5}$$

In EgoMem, memory writing is asynchronous to the main dialog, executed through independent processes. Unlike traditional memory agents, EgoMem takes as input the raw multimodal stream fragments in the dialog history ("episodic memory"), and extracts textual descriptions for the events, user persona, and other useful information.

Updating. EgoMem periodically performs online or offline memory consolidation: it integrates existing memory into new, consolidated representations, and solves potential conflicts:

$$M \leftarrow \textsf{EgoMem.update}(M). \tag{6}$$

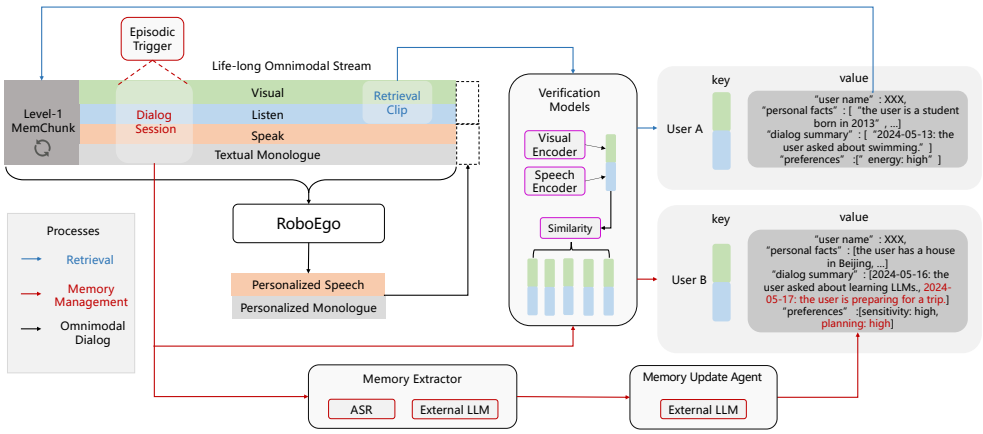


Figure 2: System illustration for EgoMem Level-1 (Profile-only).

3 EGOMEM

The design of EgoMem can be divided to two levels: Level-1 EgoMem facilitates profile-based personalization; and Level-2 EgoMem supports additional references such as users' social networks. Compared to Level-1, Level-2 is particularly suited for application scenarios where users are more interconnected, such as home robots. We introduce the implementation of each level based on our dialog flow outlined in Section 2.1.

3.1 LEVEL-1 EGOMEM: PROFILE-ONLY

In Level-1, M contains only the profile information for each of its recognized users. Formally, it sets $c_t = \text{None}$ in equations 2 and 3. The profile of each user in a key-value pair: the *key* contains one visual embedding v_f^u for face verification and one audio embedding v_s^u for speaker verification; the *value* is a dictionary storing the users' name, personal facts, summary of previous dialogs, and preferences. The operation flow of Level-1 EgoMem is illustrated in Figure 2. It is driven by the three component processes running asynchronously: the *retrieval* process (Section 3.1.1), the *omnimodal dialog* process (Section 3.1.2), and the *memory management* process (Section 3.1.3).

3.1.1 RETRIEVAL

To identify the current user, a retrieval process runs in a lifelong "polling" manner with fixed intervals of 2 seconds. This is a critical design enabling the dialog model to *actively start* talking to the user (e.g., "*Greetings John!*"). In every 2 seconds, EgoMem processes a chunk of audio and visual signals with length τ, and extracts query vectors:

$$v_f^q = \text{visual\_encoder}(v_{(t-\tau):t}), \tag{7}$$

$$v_s^q = \text{speech\_encoder}(a_{(t-\tau):t}). \tag{8}$$

v_f^q and v_s^q are used for face and speaker verification with each *user*'s key (v_f^u and v_s^u), respectively. If a valid user is found, its profile is tokenized with a textual tokenizer and pushed to the textual channel of a special *Level-1 MemChunk* field in the main dialog's token stream (Figure 2, top left). This special chunk occupies a maximum length of 512 time steps, with its 16 audio tokens always filled with <*empty*>. It is attended by every forward pass of the RoboEgo model. Note that *Level-1 MemChunk* is managed exclusively by the retrieval process: its textual content is refreshed only when the current recognized user differs from the previous one. Every time *Level-1 MemChunk* is refreshed, EgoMem triggers one additional forward pass for RoboEgo that updates the KV cache for the entire dialog history. This operation introduces ignorable overhead in inference.

Face Verification. We leverage an open-sourced pipeline from DeepFace (Serengil & Ozpinar, 2024) to extract faces from video frames. Specifically, we use Retinaface (Deng et al., 2020) as

a face detection backend, and Facenet512 (Schroff et al., 2015) as the visual encoder, resulting in 512-dimensional face features. The retrieval quits if no face is detected; otherwise, we first find the closest existing user u with the minimal cosine distance $d = 1 - cosine\_similarity(v_f^q, v_f^u)$ to the query vector v_f^q, and then verify with a pre-tuned threshold $\delta = 0.3$:

$$\text{current user} = \begin{cases} u, & \text{if } d < \delta, \\ \text{new user}, & \text{else.} \end{cases} \tag{9}$$

Speaker Verification. We leverage a wavelm_large (Chen et al., 2022) model fine-tuned specifically for speaker verification (Anastassiou et al., 2024) as our speech feature extractor, producing 256-dimensional v_s^q and v_s^u vectors. We combine cosine similarity with adaptive s-norm (Karam et al., 2011; Cumani et al., 2011) for best performance and robustness (Section 5.1).

3.1.2 Omnimodal Dialog

This is the main process running the RoboEgo chat service. *Level-1 MemChunk* is attended by RoboEgo in each step. If no user profile is returned by the retrieval process, *Level-1 MemChunk*'s text channel is filled with <*pad*> tokens. We fine-tune RoboEgo with the corresponding streaming data format (Section 4) to generate personalized spoken responses based on the user's profile information.

3.1.3 Memory Management

The *memory management* process instantiates EgoMem's extract, write, and update functions (eq. 4 - 6). With a fixed time interval, it conducts content extraction on the 17-way audio-language token stream from the main dialog process. For a 8192-step stream chunk ($\sim$11 minutes) in history, each time step is labeled by a sequence tagging model (namely **Episodic Trigger**) to mark the boundaries of dialog sessions for each user. Next, an external LLM, serving as **Memory Extractor**, is prompted to extract events, user facts, and user preferences from the fragmented streams of each session. Afterwards, the memory management process calls the retrieval functions to identify the user of this session. If a *new* user is found, EgoMem creates a new memory item in M, stores the face/speech embedding as keys, and initializes the user's profile with the extracted contents. Otherwise, the user identity and the extracted memory contents are provided to a **Memory Update Agent**, which is an external LLM prompted to figure out potential conflicts and update the user's profile in M.

Episodic Trigger. The episodic trigger is used to find the boundaries of dialog sessions in which the user's identity is consistent. It not only detects the start and end of dialog sessions, but also splits the sessions from different users. Specifically, the episodic trigger predicts tags for each time step in the audio stream (the aligned listen and speak audio tokens):

$$\text{Tag}_{0\sim t} = \text{episodic\_trigger}(a_{0\sim t}, r_{0\sim t}). \tag{10}$$

Specifically, the episodic trigger assigns a label to each time step with the following paradigm: {0: *no dialog*; 1: *start of a new user's dialog session*; 2: *in-session step*; 3: *end of current user's session*}. The detailed model structure is explained in Appendix B.1.

The **Memory Extractor** and **Memory Update Agent** are also detailed in Appendix B.1.

3.2 Level-2 EgoMem: Content-driven

In Level-2 EgoMem, M maintains not only the user profiles, but also the social relation graph among them. For each user, we add a field containing a list of triplets representing the graph edges from the current user to others. Optionally, any other useful information can be added to M for a similar RAG processes. Formally, Level-2 EgoMem provides both p_t and c_t in equations 2 and 3. While p_t comes from a *polling* user recognition, c_t comes from the the primary model (RoboEgo)'s *active* retrieval to M. We exemplify Level-2 EgoMem in Figure 3, showing its major differences to Level-1. We specify the comparison to Level-1 for each of the core processes (Section 3.1.1 - 3.1.3) as follows:

Level-2 Retrieval. As shown in Figure 3, the *Level-1 MemChunk* maintains its function in the Level-2 system; it is driven by the external polling retrieval process. An *Level-2 MemChunk* with

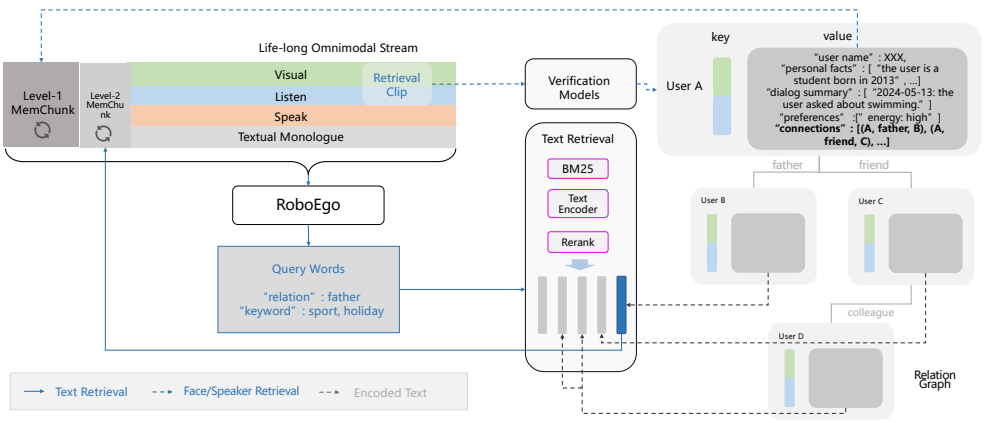


Figure 3: System illustration for EgoMem Level-2 (Content-driven). We focus on showing the differences in retrieval process and hide the details for other processes like memory management.

a maximum length of 256 is added to the reserved field in the stream; unlike *Level-1 MemChunk*, *Level-2 MemChunk* is driven by active textual queries from the RoboEgo dialog model. For example, in a dialog session between user A and RoboEgo, *Level-1 MemChunk* is kept the same (user A's profile). In contrast, after each user instruction, RoboEgo can activate an independent **Textual Retrieval** process to memory M. A textual query is generated by RoboEgo on its monologue channel based on the current dialog. The retrieval result is tokenized and cached in *Level-2 MemChunk*. The implementation of this **Textual Retrieval** sub-module is provided in Appendix B.2.

Level-2 Omnimodal Dialog. In Level-2, the primary dialog model is allowed to actively generate textual queries in arbitrary time. Specifically, we fine-tune RoboEgo to generate two groups of query words formatted as <retr>:\n<group1>\n<group2><answer>, with group1 being the "relation query" and group2 being the "keyword query". Each group is a sequence of query words separated by comma. When the final <answer> token is generated, EgoMem activates a textual retrieval process to update the *Level-2 MemChunk*. The training process is introduced in Section 4.

Level-2 Memory Management. The only differences to Level-1 include the **Memory Extractor** is prompted to also extract new *relation* facts from the raw dialog contents (e.g., User A says he is the boyfriend of User B now), and the **Memory Update Agent** is prompted to link the user to existing users accordingly, updating the edges of the social graph.

4 Training Approach

We fine-tune RoboEgo to generate personalized response with Level-1/2 EgoMem. We also train the Episodic Trigger to label the dialog boundary for memory extraction. Interestingly, the data collection for these three tasks can be unified by different *supervision masks* on the same token stream.

4.1 Data Collection

Audio Dialogs. We collect textual transcripts simulating the lifelong personalized scenarios utilizing both Level-1 and Level-2 EgoMem. We synthesize user profiles and social graphs, collect open-sourced dialog datasets, and generate ground-truth personalized answers using large (visual-)language models. The textual transcripts are converted to audiovisual dialogs using text-to-speech (TTS) models, followed by audio augmentation to improve robustness. Details are provided in Appendix C.

Token Stream Organization. Multiple dialog sessions from different users are tokenized and concatenated, forming *token streams*. With a probability of 0.3, a later user instruction *interrupts* an ongoing model response, simulating the most widely-considered full-duplex scenario. The concatenated waveform are tokenized with a Mimi tokenizer (Défossez et al., 2024), formatted

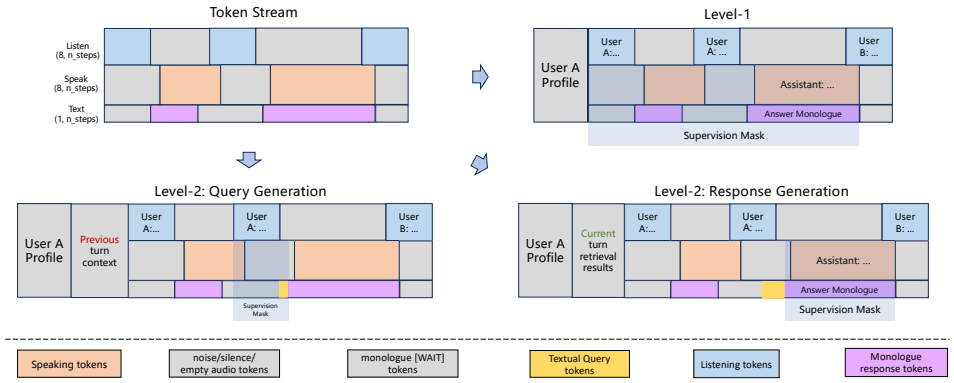


Figure 4: Token stream structure and supervision mask for EgoMem training data.

as described in Appendix A, and truncated to a maximum length of 8192 steps. For the textual monologue channel, we record the actual start point of each audio response, and position the textual response tokens to start 2 steps earlier. We follow the *natural* monologue strategy (Yao et al., 2025a) instead of applying word-level alignment between text and audio (Défossez et al., 2024). For each training sample with 8192 steps, positions 0-511 are reserved for the *Level-1 MemChunk*, and positions 512-768 for the *Level-2 MemChunk*, with dialogs beginning after these reserved slots. Figure 4 (top left) illustrates our token stream organization.

4.2 Training with Supervision Masks

We apply three different kinds of *supervision masks* on the token streams introduced above, supporting the following three tasks:

For Level-1 EgoMem, for each of the N multi-turn dialogs in a token stream, we position the corresponding user's profile in *Level-1 MemChunk*, and set the supervision mask to 1 for the textual and speak tokens only in the time span of the corresponding dialog, and 0 for other time steps, producing N distinct training samples in total (Figure 4, top right).

For Level-2 EgoMem, more fine-grained supervision mask is applied to each turn $t_i, i \in [0, T_j)$ for each dialog $d_j, j \in [0, N)$ in a token stream. Specifically, for query word generation, we maintain the content of *Level-1 MemChunk* and *Level-2 MemChunk* for the previous turn, position the ground-truth query words right before the textual response for current turn, and set supervision mask to 1 from the audio start point of the current turn until the end of query words span (Figure 4, bottom left). For personalized response generation, we conduct textual retrieval with the ground-truth query words, gather the supporting fact from the connected users in the relation graph, and fill them into the *Level-2 MemChunk*. *Level-1 MemChunk* is filled with current user profile. With these contexts, we supervise on the textual and speaking tokens from the end of query words to the audio end point of dialog, including the full response utterance (Figure 4, bottom right). To summarize, for each turn, the query words and personalized response are supervised separately with different contexts, yielding $2 * \sum_{j=0}^{N} T_j$ distinct training samples from each token stream.

For Episodic Trigger, we leverage a full supervision mask. Each time step in the token stream is labeled following the paradigm in Section 3.1.3.

The training configurations for the above three tasks are introduced in Appendix D.

5 Experiments

We focus on the following three research questions: (1) Do our retrieval sub-modules correctly identify users and recall the relevant contents? (2) Does the episodic trigger detect the correct boundaries of omnimodal dialogs? (3) Does the fine-tuned RoboEgo model effectively leverage the Level-1 and Level-2 EgoMem to deliver lifelong personalized responses? We answer these questions with quantitative results on dedicated benchmarks. As the first work in lifelong memory agents for

Table 1: Face verification, speaker recognition, and text retrieval results for EgoMem sub-modules.

Tasks	Face Verification	Speaker Recognition			Text Retrieval
Metrics	Accuracy	pass@1 w/o s-norm	pass@1 w/ s-norm	EER	pass@5
Results	0.984	0.958	0.965	0.00892	0.960
Elapsed time (s)	0.2		0.1		0.1

full-duplex omnimodal systems, we will also release part of our test set for personalized omnimodal dialog generation to benefit future research.

5.1 Retrieval Evaluation

We present the benchmark and settings to evaluate the retrieval capabilities for EgoMem. The retrieval results are summarized in Table 1.

Face Verification. We benchmark the face retrieval module on the Labeled Faces in the Wild (LFW, (Huang et al., 2008)) dataset, achieving an accuracy of 98.4%, which is consistent with public results [1]. As the open-sourced solution demonstrates satisfying face verification performance and robustness to variations in pose and angle, we directly integrate it in our system without additional fine-tuning. The face retrieval module processes one query within 0.2 seconds with a single Nvidia H100 GPU.

Speaker Verification. We construct our evaluation benchmark from the public VoxCeleb (Nagrani et al., 2017) speaker verification dataset. To enable adaptive s-norm (Karam et al., 2011; Cumani et al., 2011) which is widely agreed to benefit the task, we leverage SeedTTS-eval (Anastassiou et al., 2024) as the source of imposter cohorts, using 1,000 Chinese speech embeddings as the candidate cohort set for the queries, and 2,000 embeddings for the keys. We set up a cohort number of 200 (i.e., for both the queries and the keys, the 200 closest utterances in the candidate cohorts are used to compute the mean and variance statistics for adaptive s-norm).

To assess the impact of adaptive s-norm, we first synthesize a retrieval task with 1,000 query utterances and 120 key utterances from different speakers in VoxCeleb, and compare the pass@1 with or without adaptive s-norm. We observe a moderate improvement from 95.8% to 96.5% with adaptive s-norm, confirming its benefit for retrieval stability.

Next, we sample a more challenging speaker verification test set from VoxCeleb with a highly imbalanced ratio of positive (same-speaker) to negative (different-speaker) pairs of 1:119, yielding 5,000 samples in total. Our speaker verification module achieves an Equal Error Rate (EER) of 0.89% on this benchmark with a decision threshold of 4.63. For deployment, we adjust the threshold to 6 based on human case studies to balance precision and recall. The whole speaker verification system takes less than 0.1 seconds for a retrieval run with more than 1,000 candidate entries.

Text Retrieval. In Level-2 EgoMem, text retrieval is used to gather relevant information w.r.t the relation and keyword queries. We therefore focus on the pass@5 metric, as it measures the ability of the system to return all relevant facts within a textual window shorter than 256 tokens (the size of the *Level-2 MemChunk*). We construct a benchmark of 200 queries sampled from our personalized dialog transcripts, with a candidate entry pool of 500 (relation, personal fact) texts. Using the straightforward retrieval strategy described in Appendix B.2, the system achieves a pass@5 score of 96%. The full system latency is controlled to be under 0.1s with a single Nvidia H100 GPU.

5.2 Episodic Trigger Evaluation

We hold out a test set from the collected token streams (Section 4.1) for episodic trigger evaluation, containing 1,000 samples. We use two types of metrics: (1) *Jaccard score* measuring the overlap of dialog session spans; (2) *span_match@N* which measures precision, recall, and F1 scores for detected dialog boundaries, allowing a tolerance of $\pm N$ steps from the ground truth.

[1] https://github.com/serengil/deepface/tree/master/benchmarks

Table 2: Episodic trigger evaluation results.

Metrics	Jaccard	P/R/F1@0	P/R/F1@5	P/R/F1@10	Elapsed time (s)
Clean	0.992	0.857/0.857/0.857	0.986/0.986/0.986	0.986/0.986/0.986	0.08
Noised	0.989	0.790/0.788/0.789	0.983/0.981/0.982	0.984/0.982/0.983	

Table 3: Personalized dialog evaluation results.

Models	Level-1			Level-2		
Metrics	Fact Score	Answer Quality	Throughput (fps)	Fact Score	Answer Quality	Throughput (fps)
Clean	0.959	9.170	21.73	0.895	8.970	20.56
Noised	0.931	9.020		0.876	8.820	

The results are presented in Table 2. At $N = 5$, our episodic trigger achieves an F1 score of more than 0.98 under both clean and noised environments, indicating robust "Valid Audio Detection" (VAD) and user session splitting capabilities within a deviation of less than $(5/12.5 = 0.4)$ seconds. Notably, noisy environments can significantly affect the prediction of more fine-grained boundaries (i.e., less than 0.2s), as we observe a large gap on *span_match@0*. This is intuitive since it takes time to figure out whether a voice indicates the start of a new user's session or just another period of noise. The episodic trigger takes 0.08 seconds to annotate a 10-minute chunk with 8192 time steps.

5.3 Personalized Dialog Evaluation

We hold out a test set from the masked token streams (Section 4.1) to assess the quality of personalized responses produced by RoboEgo, integrated with Level-1 and Level-2 EgoMem. For each dialog turn, we provide an evaluator model with the following inputs: the user instruction (textual transcript), the ground-truth textual response, the contents of the *MemChunk*s, and the textual monologue response generated by RoboEgo. The evaluator is implemented with the DeepSeek-V3 API, prompted to return two scores: (1) *Fact Score*: A binary 0/1 metric for each turn indicating whether the model's response is personalized to the user and consistent with the user profile, without factual errors. (2) *Answer Quality*: A score from 0 to 10 for each turn measuring the general helpfulness and quality of the response with respect to the user instruction, regardless of personalization.

We present the results in Table 3. We observe that for both Level-1 and Level-2 EgoMem, the models successfully achieve the expected RAG capability based on the retrieval results present in *MemChunk*s. RoboEgo achieves lower *Fact Score* in the Level-2 task, largely due to more frequent *MemChunk* updates and error cascading from the textual retrieval module. For the *Answer Quality* scores which are independent of the retrieval results, the gap becomes smaller, indicating that neither Level-1 nor Level-2 EgoMem significantly degrades the base instruct-following capability of RoboEgo.

We observe a slight drop in throughput with Level-2 memory as it introduces a longer *MemChunk*. Yet, this latency is negligible in the user experiences of full-duplex real-time chatting, as the model generates audio frames in more than 20 fps, significantly exceeding the minimum requirement for real-time audio decoder (12.5 fps).

6 Conclusion and Future Challenges

In this work, we explored lifelong memory for full-duplex omnimodal models. We first defined the task and outlined the core functions, and then introduced our proposed memory system, EgoMem: Level-1 (profile-based) and Level-2 (content-driven). We integrated EgoMem to the omnimodal dialog model, RoboEgo, as an implementation example. Experimental results demonstrate that, for the first time, an omnimodal dialog agent can be equipped with robust lifelong personalization capabilities, establishing a strong baseline to support future research. Due to computational constraints, we did not explore larger model sizes or more advanced functionalities, such as complex tool use. Future directions include extending the profile/graph memory to encompass procedural memory and multimodal contents, as well as investigating whether trainable parameters can replace some of the complex agent modules and memory units.

ETHICS STATEMENT

The data used to train the three tasks supporting our EgoMem agent is derived from synthetic transcripts generated by publicly accessible large language models. No real-world users are involved in this process, and no privacy is compromised during data collection. EgoMem is a plug-in methodology that can be applied to a wide range of models. The content generated by the dialog models does not reflect the views or opinions of the authors or affiliated institutions.

REPRODUCIBILITY STATEMENT

We provide comprehensive details of the system paradigm, implementation, and training configurations in Sections 2, 3, 4, as well as in the Appendix. We will release a portion of our test set along with the associated code for stream organization, inference, and evaluation. The functions and signals used in Level-1 and Level-2 EgoMem are clearly defined, which we believe will support the community in reproducing our agent system and developing future variants.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. CoRR, abs/2503.06669, 2025. doi: 10.48550/ARXIV.2503.06669. URL https://doi.org/10.48550/arXiv.2503.06669.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, Jiaxin Li, Xiaoyang Li, Xingxing Li, Lin Liu, Shouda Liu, Sichao Liu, Xudong Liu, Yuchen Liu, Zhengxi Liu, Lu Lu, Junjie Pan, Xin Wang, Yuping Wang, Yuxuan Wang, Zhen Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuanhao Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. Seed-tts: A family of high-quality versatile speech generation models. CoRR, abs/2406.02430, 2024. doi: 10.48550/ARXIV.2406.02430. URL https://doi.org/10.48550/arXiv.2406.02430.

Steven Au, Cameron J Dimacali, Ojasmitha Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A Rossi, and Nesreen K Ahmed. Personalized graph-based retrieval for large language models. arXiv preprint arXiv:2501.02157, 2025.

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

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv preprint arXiv:2503.06669, 2025.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? arXiv preprint arXiv:2403.20330, 2024b.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing, 16(6): 1505–1518, 2022.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. arXiv preprint arXiv:2504.19413, 2025.

Sandro Cumani, Pier Domenico Batzu, Daniele Colibro, Claudio Vair, Pietro Laface, Vasileios Vasilakakis, et al. Comparison of speaker recognition approaches for real applications. In Interspeech, pp. 2365–2368, 2011.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691, 2023.

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. CoRR, abs/2410.00037, 2024. doi: 10.48550/ARXIV.2410.00037. URL https://doi.org/10.48550/arXiv.2410.00037.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5203–5212, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, Babak Naderi, Sebastian Braun, Ross Cutler, Hannes Gamper, Mehrsa Golestaneh, and Robert Aichner. Icassp 2023 deep noise suppression challenge. In ICASSP, 2023.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2(1), 2023.

Gemini. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530, 2024.

Simon Haykin and Zhe Chen. The cocktail party problem. Neural computation, 17(9):1875–1902, 2005.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13504–13514, 2024.

Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. Advances in Neural Information Processing Systems, 37:59532–59569, 2024.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. arXiv preprint arXiv:2506.06326, 2025.

Zahi N Karam, William M Campbell, and Najim Dehak. Towards reduced false-alarms using cohorts. In 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4512–4515. IEEE, 2011.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. arXiv preprint arXiv:2402.09727, 2024.

Jia-Nan Li, Jian Guan, Songhao Wu, Wei Wu, and Rui Yan. From 1,000,000 users to every user: Scaling up personalized preference for user-level alignment. arXiv preprint arXiv:2503.15463, 2025a.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, et al. Memos: An operating system for memory-augmented generation (mag) in large language models. arXiv preprint arXiv:2505.22101, 2025b.

Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. Fish-speech: Leveraging large language models for advanced multilingual text-to-speech synthesis, 2024. URL https://arxiv.org/abs/2411.01156.

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, 2025. URL https://arxiv.org/abs/2503.04721.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437, 2024.

Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. Base of rope bounds context length. arXiv preprint arXiv:2405.14591, 2024.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612, 2017.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389, 2009.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815–823, 2015.

Sefik Serengil and Alper Ozpinar. A benchmark of facial recognition pipelines and co-usability performances of modules. Journal of Information Technologies, 17(2):95–107, 2024. doi: 10.17671/gazibtd.1399077. URL https://dergipark.org.tr/en/pub/gazibtd/issue/84331/1399077.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. CoRR, abs/2104.09864, 2021. URL https://arxiv.org/abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Yuanjun Xiong, and Wei Xia. A full-duplex speech dialogue scheme based on large language models. CoRR, abs/2405.19487, 2024. doi: 10.48550/ARXIV.2405.19487. URL https://doi.org/10.48550/arXiv.2405.19487.

Yequan Wang and Aixin Sun. Toward embodied agi: A review of embodied ai and the road ahead. arXiv preprint arXiv:2505.14235, 2025. URL https://arxiv.org/abs/2505.14235.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. arXiv preprint arXiv:2507.07957, 2025.

Bingyang Wu, Shengyu Liu, Yinmin Zhong, Peng Sun, Xuanzhe Liu, and Xin Jin. Loongserve: Efficiently serving long-context large language models with elastic sequence parallelism. In Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles, pp. 640–654, 2024.

X.AI. Realworldqa, 2024. URL https://x.ai/blog/grok-1.5v.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. CoRR, abs/2304.12244, 2023. doi: 10.48550/arXiv.2304.12244. URL https://doi.org/10.48550/arXiv.2304.12244.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. arXiv preprint arXiv:2503.20215, 2025a.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. arXiv preprint arXiv:2502.12110, 2025b.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. arXiv preprint arXiv:2310.00704, 2023.

Yiqun Yao, Xiang Li, Xin Jiang, Xuezhi Fang, Naitong Yu, Wenjia Ma, Aixin Sun, and Yequan Wang. Flm-audio: Natural monologues improves native full-duplex chatbots via dual training. arXiv preprint arXiv:2509.02521, 2025a.

Yiqun Yao, Xiang Li, Xin Jiang, Xuezhi Fang, Naitong Yu, Aixin Sun, and Yequan Wang. Roboego system card: An omnimodal model with native full duplexity. arXiv preprint arXiv:2506.01934, 2025b.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, et al. Omniflatten: An end-to-end gpt model for seamless voice conversation. arXiv preprint arXiv:2410.17799, 2024a.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. Beyond the turn-based game: Enabling real-time conversations with duplex models. arXiv preprint arXiv:2406.15718, 2024b.

Hanyu Zhao, Li Du, Yiming Ju, Chengwei Wu, and Tengfei Pan. Beyond iid: Optimizing instruction learning from the perspective of instruction interaction and dependency. 2024. URL https://arxiv.org/abs/2409.07045.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pp. 19724–19731, 2024.

Yongxin Zhu, Dan Su, Liqiang He, Linli Xu, and Dong Yu. Generative pre-trained speech language model with efficient hierarchical transformer. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1764–1775, 2024.

A Introduction to the RoboEgo Model

For audio signals a_t, we use the Mimi tokenizer [2] to extract features at 12.5 frames per second. Each audio frame is represented by one semantic token and seven acoustic tokens. The audio input and output are divided into two channels, *listening* and *speaking*, while textual monologue tokens are placed in an additional textual channel. Thus, . They are additively merged into the input embedding. The model then processes all the historical input embeddings with a 7B LLM backbone to generate the hidden state for the current time step. Following the RQ-Transformer architecture (Yang et al., 2023; Zhu et al., 2024), a lightweight *depth* Transformer (with 100M parameters) first generates a textual monologue token based on the current hidden state on the top layer, and then generates eight speaking tokens autoregressively. In lifelong deployment scenarios, this process runs continuously in real time, forming the main dialog stream, while visual signals v_t are encoded through a Visual Transformer (Dosovitskiy et al., 2020) and added into the context in a time-division multiplex (TDM) manner, at fixed intervals of 2–4 seconds. We refer the readers to the related work (Yao et al., 2025b; Défossez et al., 2024; Yao et al., 2025a) for detailed structural configurations.

B Extra details for EgoMem Sub-modules

B.1 Level-1 Sub-modules

Episodic Trigger. The episodic trigger is an RQ-Transformer-based (Défossez et al., 2024) model which shares the input stream organization and model structure topology with RoboEgo (Section 2.1), despite being much smaller with 100M parameters. It consumes the 17 audio-text channels with a maximum time step of 8192 as input. Instead of generating dialog responses, it assigns a label to each time step with the following paradigm: {0: *no dialog*; 1: *start of a new user's dialog session*; 2: *in-session step*; 3: *end of current user's session*}. We modify the attention mask from a GPT-like (Brown et al., 2020) causal mask to a Bert-like (Devlin et al., 2019) full mask, as the sequence labeling process is offline and chunk-wise. The training configurations of episodic trigger is detailed in Section 4. The evaluation results are presented in Section 5.2.

Memory Extractor. The memory extractor is implemented as the following pipeline:

- The episodic trigger labels each time step. According to the labels, audio chunks starting with label "1", ending with label "3", and correctly filled with "2" are considered as the audio source for one user's dialog session. The corresponding audio waveforms are clipped.
- The clipped waveform in the listening channel goes through automatic speech recognition (ASR). Raw ASR results are fed into the memory extractor. The response texts from the monologue text channel of the dialog model are also provided as reference.
- We leverage a DeepSeek-V3 (Liu et al., 2024) API to extract meaningful contents to store in the memory. Specifically, we prompt the model to summarize the dialog content with short, precise sentences, generate sentences describing the facts about the user, and figure out a 90-dimensional personal trail (Li et al., 2025a; Kang et al., 2025) for the user. The user name (or "unknown_user") is stored in a separate field.

Memory Update Agent. This is a DeepSeek-V3 API prompted to solve profile conflicts and formalize the extracted contents from the memory extractor, making sure that they are suitable for updating the structured memory storage.

B.2 Level-2 Submodules

Textual Retrieval. The textual retrieval system gathers the top-K relevant textual information according to the query words generated by RoboEgo, and updates the content of *Level-2 MemChunk*. Specifically, for each user U connected to current user A in the social graph, U's name and relation with A are concatenated with each of U's memory items (facts, dialog history, etc.) to form one candidate document for retrieval. If the "relation query" group is not empty, we first match all relevant

[2] https://huggingface.co/kyutai/mimi

users' documents via the BM25 (Robertson et al., 2009) algorithm using only the relation queries; next, if the "keyword query" group is not empty, we concatenate all the keyword into one string, and re-rank the retrieved documents based on their vector distances to this keyword string. We leverage the BGE-small (Chen et al., 2024a) model as the textual encoder. The top-K results are returned to the textual channel of *Level-2 MemChunk*.

C Data Collection Details

Textual Transcripts *For Level-1 EgoMem,* we first use DeepSeek-V3 (Liu et al., 2024) to synthesize 500 user profiles including name, dialog history, and a 90-dimensional persona. Next, we synthesize 10k dialogs between user and AI assistant based on open-source instruct-following datasets (e.g., Infinity-Instruct (Zhao et al., 2024), WizardLM (Xu et al., 2023), and multimodal question answering datasets involving visual inputs (Chen et al., 2024b; X.AI, 2024)): the user instructions are retained, while responses are refined by DeepSeek-V3 (or Gemini-2.5-Pro (Gemini, 2024) for VQA) to be more helpful and personalized. Finally, we prompt DeepSeek-V3 to include more question styles in which users ask questions regarding their own dialog history and profiles. Each dialog typically contains 3–5 turns.

For Level-2 EgoMem, we first synthesize 500 possible relations (e.g., "father", "colleague"), construct relation graphs linking one main user with 3–5 socially connected users. We prompt DeepSeek-V3 to generate questions requiring relation-graph reasoning (e.g., "Does my mother like physical exercise?") and mix these questions with general instructions, producing 5k dialogs. For each question, the model annotates the effective query words (including both relation query and keywords query, both can be empty) sufficient to retrieve supporting facts from the profiles of connected users. After generating the query words, the model should also provide the ground-truth personalized response for training.

TTS and Augmentation. Audio dialogs are synthesized from the collected transcripts. Each user utterance is assigned a random human voice, and converted into speech with Fishaudio TTS (Liao et al., 2024), while model responses are consistently generated with a single fixed voice. For the listening channel, we add diverse noise from sources like DNS Challenge (Dubey et al., 2023) and RNNoise[3], as well as random speech clips. Following Moshi (Défossez et al., 2024), we also simulate microphone echo by mixing the speaking channel into the listening channel with probability 0.3, applying random gain (0-0.2x) and delay (0.1-0.5s).

D Training Details

For Level-1 EgoMem, we fine-tune RoboEgo with a dataset containing 158K samples with different (*Level-1 MemChunk*, token stream context, supervision mask) combinations. We duplicate the dataset by using both the original and noise-augmented listening channel. Training starts from one of RoboEgo's SFT checkpoints, running for 5 epochs with batch size 64 and a cosine learning rate decay from 1e-5 to 1.5e-6.

For Level-2 EgoMem, RoboEgo is fine-tuned on 54K samples with valid *Level-2 MemChunk* and the corresponding query words, combined with 50% of the Level-1 training dataset reformatted with empty *Level-2 MemChunk* and query words. As with Level-1, the dataset is duplicated with clean and noisy listening channels. Training resumes from the same RoboEgo checkpoint as in Level-1, running for 1 epoch with batch size 64 and a cosine learning rate decay from 1e-5 to 1.5e-6.

For Episodic Trigger, the sequence tagging model is initialized randomly. We train the model with 100K clean samples and 100K noised samples. The number of epoch is set to 45. We use a batch size of 64 and the learning rate decays from 1e-4 to 1e-6 following a cosine schedule.

[3] https://github.com/xiph/rnnoise