# Neuroplex: A Speech-to-Speech Architecture for Interpretable and Controllable Enterprise Voice Agents

Deepgram Research

## 1 Introduction

### 1.1 Speech-to-Speech for Enterprise Voice Agents

Large language models (LLMs) and multimodal foundation models have emerged as a promising paradigm to build conversational voice agents that automate customer service, scheduling, and support tasks. The application of existing sequence modeling approaches from LLMs to audio has driven an evolution in the deep learning community away from traditional cascaded architectures and toward end-to-end speech-to-speech (STS) models. Despite their promise, deploying STS models in production environments presents significant technical challenges that remain to be addressed.

The computational demands of production deployment are substantial. Voice agent systems must process millions of concurrent conversations while maintaining strict latency requirements to enable natural interactions. Meeting these constraints while keeping operational costs manageable requires careful optimization of model architectures and inference strategies. Traditional approaches using separate ASR, LLM, and TTS models quickly become computationally prohibitive at scale.

Quality and controllability requirements further complicate deployment. The system must maintain high-quality speech perception across diverse accents, challenging acoustic conditions, and localized jargon and vocabulary, while still ensuring sophisticated language understanding and natural speech synthesis. Moreover, the agent's behavior must be steerable through high-level controls to maintain consistent persona and comply with business requirements. This combination of quality and control demands sophisticated model architectures that can be tuned and monitored.

Debuggability is essential for production systems but particularly challenging for end-to-end models. When issues arise, system behaviors must be monitorable and errors must be traceable to their source. This requires maintaining interpretable intermediate representations throughout the STS pipeline, rather than treating the system as a black box. Yet preserving such interpretability while achieving end-to-end optimization remains an open challenge.

In this technical brief, we introduce Neuroplex, a novel speech-to-speech architecture inspired by modular specialization in the mammalian brain. In the following sections, we detail the architecture and training methodology of an initial research prototype, and demonstrate its key capabilities. The results suggest that Neuroplex represents a promising practical step towards expressive, debuggable end-to-end voice agent systems.

### 1.2 Previous Work

Traditional speech-to-speech systems cascade separate ASR, LLM, and TTS models. While modular and interpretable, these pipelines suffer from several limitations. Converting speech to text and back loses rich acoustic information about speaker state and environmental context that could inform response generation. The cascaded architecture also suffers from error propagation, lacks end-to-end optimization, and can be computationally expensive at scale since it typically involves autoregressive decoding of the three component models.

Early attempts at end-to-end speech-to-speech modeling, such as Nguyen et al. (2022) and Ma et al. (2024), focused on direct speech signal modeling without relying on text intermediaries. While these systems successfully captured paralinguistic features and turn-taking dynamics directly from audio, they often struggled with semantic coherence and complex reasoning due to limited training data and the absence of large-scale language models.

Current research in end-to-end STS models has diverged into two main approaches. The first develops native multimodal architectures that treat speech and text as co-equal modalities from the ground up. These models, such as Défossez et al. (2024) and Zeng et al. (2024), unify speech and text representations within a single architecture, often incorporating specialized mechanisms for real-time dialogue and expressive speech generation. Zhang et al. (2024b) and Zhang et al. (2024a) demonstrate that such unified architectures can achieve sub-100ms latency while maintaining natural turn-taking and full-duplex capabilities. However, these models often require extensive training from scratch and struggle to retain the sophisticated reasoning capabilities and steerability of large language models.

The second approach extends existing large language models to handle speech while carefully preserving their sophisticated language understanding capabilities. These systems introduce specialized adapters and alignment modules to bridge modalities while keeping the LLM backbone largely frozen. Models like Fang et al. (2024) and Wang et al. (2024) exemplify this strategy, using careful alignment techniques to maintain the LLM's core capabilities while adding speech understanding and generation. Chen et al. (2025) demonstrates this approach can scale to multilingual settings through extensive alignment training. While these models leverage powerful pre-trained language understanding, they can face challenges in maintaining low latency and achieving truly end-to-end optimization due to their modular design. They also typically rely on decoding the intermediate model representation to text, which loses rich acoustic information and adds latency.

## 2 Neuroplex Architecture and Implementation

Neuroplex is an end-to-end speech-to-speech architecture inspired by the efficiency of the mammalian brain, where specialized regions work in concert through generalized connections to achieve optimal cognitive processing. Like neural pathways connecting specialized brain regions, Neuroplex is built by fusing state-of-the-art component models for speech perception (ASR), language understanding (LLM), and speech synthesis (TTS) through learned adapter networks. These adapters enable compressed, rich information flow between latent spaces, similar to how the brain's white and gray matter structures facilitate efficient communication between specialized regions.

Neuroplex is trained end-to-end in a multi-task framework that preserves each component's specialized capabilities while ensuring alignment between the LLM's internal representations and generated speech. This alignment, combined with the LLM's instruction-following capabilities, enables fine-grained control over the model's persona and output characteristics through system prompts. The architecture maintains interpretability while allowing precise control over generated speech via natural language instructions.

### 2.1 Model Architecture

The system comprises four primary modules—ASR, LLM, text2codes (T2C), and codes2audio (C2A)—connected via trainable adapter networks that transform hidden representations between components. Figure 1 illustrates the model architecture. In Table 1, we detail the core components of the architecture and their functions.

The information flow through these components follows a clear progression: the ASR module first converts input speech into hidden representations, which are then transformed by the ASR2LLM adapter into a format suitable for language understanding. The LLM processes these embeddings to generate appropriate responses, which the LLM2T2C adapter then transforms into a representation appropriate for speech synthesis. Finally, the T2C module converts these representations into discrete speech codes, which the C2A module efficiently converts into output audio. This

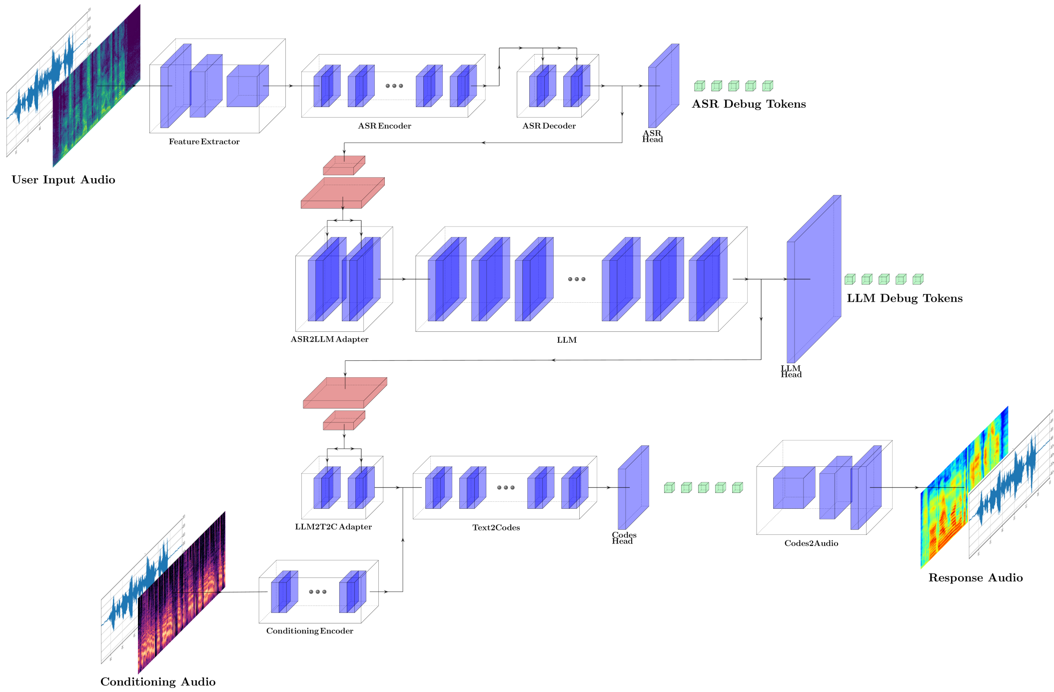


Figure 1: Neuroplex architecture showing the modular pipeline from input audio to response audio. The system consists of specialized components (Feature Extractor, ASR, LLM, Text2Codes, Codes2Audio) connected by learned adapters (ASR2LLM, LLM2T2C). Debug tokens can be extracted at multiple stages for model inspection.

pipeline maintains a continuous representation space between components while providing discrete inspection points for debugging.

### 2.2 Training Methodology

Neuroplex employs a multi-task training approach that combines specialized losses for each component with adapter alignment objectives. The adapter networks are trained using a combination of mean squared error and cosine similarity losses to ensure both magnitude and directional alignment of embeddings:

$$\mathcal{L}_{adapter} = \alpha(1 - \cos(Z_{pred}, Z_{target})) + (1 - \alpha)||Z_{pred} - Z_{target}||^2 \tag{1}$$

where $Z_{pred}$ represents the adapter's output embeddings and $Z_{target}$ represents the target embeddings from the subsequent module. This loss is applied to both the ASR2LLM ($\mathcal{L}_{ASR2LLM}$) and LLM2T2C ($\mathcal{L}_{LLM2T2C}$) adapters. The main components utilize standard cross-entropy losses ($\mathcal{L}_{ASR}$, $\mathcal{L}_{LLM}$, $\mathcal{L}_{T2C}$) for their respective tasks. These losses are combined into a weighted total objective:

$$\mathcal{L}_{total} = \lambda_{ASR}\mathcal{L}_{ASR} + \lambda_{ASR2LLM}\mathcal{L}_{ASR2LLM} + \lambda_{LLM}\mathcal{L}_{LLM} + \lambda_{LLM2T2C}\mathcal{L}_{LLM2T2C} + \lambda_{T2C}\mathcal{L}_{T2C} \tag{2}$$

Through different combinations of loss weights and parameter freezing strategies, Neuroplex can be trained to exhibit distinct behavioral regimes:

- **Cascade-like Regime:** Setting $\lambda_{ASR} = \lambda_{LLM} = \lambda_{T2C} = 0$ and training only adapter parameters produces a strict cascade of frozen pre-trained components. Each module maintains its original specialized behavior while adapters learn to bridge the representation gaps. This regime preserves the full capabilities of each pre-trained model but limits end-to-end optimization.

Table 1: Core components of the Neuroplex architecture and their functions.

| Component | Description |
|---|---|
| ASR | Pre-trained production ASR model based on an encoder-decoder transformer architecture. Processes input audio features $a \in \mathbb{R}^{B \times T_{in} \times D_{in}}$ to produce hidden states $x_{asr} \in \mathbb{R}^{B \times T_{asr} \times d_{asr}}$. Optional token outputs available for debugging. |
| ASR2LLM Adapter | Sequence-to-sequence transformation that maps ASR hidden states to LLM-compatible embeddings: $x_{a2l} = g_{a2l}(x_{asr}) \in \mathbb{R}^{B \times T_{a2l} \times d_{llm}}$. Can be implemented through simple MLP transformations or more complex s2s architectures. |
| LLM | Pre-trained, instruction-tuned large language model that processes embedded inputs through transformer blocks to generate contextual hidden states $x_{llm} = f_{LLM}(x_{a2l})$. Preserves instruction-following capabilities for model control. |
| LLM2T2C Adapter | Sequence-to-sequence transformation that maps LLM hidden states to T2C-compatible embeddings: $x_{l2t} = g_{l2t}(x_{llm}) \in \mathbb{R}^{B \times T_{l2t} \times d_{t2c}}$. Can be implemented through simple MLP transformations or more complex s2s architectures. |
| T2C | Autoregressive sequence model that maps text or text embeddings to discrete speech codes. Generates a sequence of acoustic tokens representing compressed speech features. |
| C2A | Streaming convolutional decoder that efficiently reconstructs audio waveforms from the discrete speech codes. Enables real-time audio synthesis with low latency. |

- **End-to-end with Modular Specialization:** Using balanced positive weights ($\lambda_{ASR}, \lambda_{LLM}, \lambda_{T2C} > 0$) and unfreezing all parameters allows components to adjust while maintaining their specialized roles. This regime enables end-to-end optimization of speech quality while preserving the interpretability of intermediate representations and the LLM's instruction-following capabilities. The model remains steerable and debuggable through its modular structure.
- **End-to-end Monolithic:** Setting $\lambda_{T2C} > 0$ with all other weights zero transforms Neuroplex into a monolithic speech-to-speech model. While this regime can potentially achieve optimal speech output quality, it sacrifices modularity, interpretability, and controlled generation through LLM instructions. The internal representations no longer maintain specialized linguistic or acoustic meanings.

In early research prototypes, we have found success with a progressive three-stage training curriculum. Stage 1 focuses on adapter pre-training, where the main components (ASR, LLM, T2C) remain frozen while the adapter networks are trained using adapter losses and setting $\lambda_{ASR} = \lambda_{LLM} = \lambda_{T2C} = 0$. This establishes initial connectivity between the specialized modules through the adaptation layers with the model exhibiting cascade-like behavior. Stage 2 maintains the frozen state of the main components while training adapters with all loss terms active. This phase develops robust representations across different voices and speaking styles. Finally, Stage 3 trains the model in the modular specialization regime using conversational prompt/response data, with all model weights unfrozen and using balanced loss terms. We find that the third stage effectively preserves alignment between the model's internal representations and the input and output audio, thus enabling debuggability and steerability.

### 2.3 Inference Mechanics

Neuroplex operates in a continuous mode that enables direct propagation of semantic and acoustic information through hidden state transformations, distinct from traditional cascaded speech-to-speech systems that rely on discrete text representations. To illustrate the key differences, we first describe a baseline discrete inference mode before detailing Neuroplex's primary continuous operating mode.

In a discrete cascaded approach, inference proceeds by generating explicit tokens between components: the ASR module produces text tokens which are then fed to the LLM, whose output tokens are provided to the T2C module for speech synthesis. This introduces non-differentiable sampling steps between modules and requires explicit decoding/encoding at each stage.

In contrast, Neuroplex's continuous mode maintains hidden state representations throughout the pipeline, enabling direct propagation of information between components. The inference process proceeds as follows:

1. The ASR module processes input audio features $a \in \mathbb{R}^{B \times T_{in} \times D_{in}}$ to produce hidden states $x_{asr}$. While text tokens can be optionally decoded for debugging, the primary forward path maintains the continuous representation.
2. The ASR2LLM adapter transforms these hidden states into LLM-compatible embeddings: $x_{a2l} = g_{a2l}(x_{asr})$. This transformation preserves semantic content while adjusting the representation dimensionality and distribution.
3. The LLM processes these embeddings directly, generating hidden states $x_{llm} = f_{LLM}(x_{a2l})$ that encode the response. Again, text tokens can be optionally decoded for inspection without affecting the primary pipeline.
4. The LLM2T2C adapter converts LLM hidden states into T2C-compatible embeddings: $x_{l2t} = g_{l2t}(x_{llm})$, maintaining the continuous flow of information.
5. Finally, the T2C module converts these embeddings into discrete acoustic codes, which the C2A module synthesizes into output audio.

This continuous inference mode offers several advantages: it eliminates the need for intermediate text generation, maintains end-to-end differentiability, and potentially preserves richer semantic and acoustic information that might be lost in discrete token representations. The optional debug tokens at ASR and LLM stages enable system monitoring and validation without disrupting the primary continuous pipeline.

# 3 Model Analysis and Demonstrations

In this section we report initial results from a research prototype of Neuroplex trained to question-answering tasks. We utilized the three-stage training approach described in Section 3.2. The first two stages trained the model on a large-scale multispeaker dataset, focusing first on adapter pre-training and then on scaled multi-speaker refinement. For the final multi-task fine-tuning stage, we trained the model on a synthetically generated dataset comprising prompts and responses created using an internal Deepgram TTS model. This prototype demonstrates single-turn dialogue capabilities in question-answering scenarios.

## 3.1 Prompt/Response Samples

In this section we present a collection of inference examples from the Neuroplex prototype. For these examples we generated synthetic audio queries from pre-written text prompts using a Deepgram TTS model. We then fed these queries to the Neuroplex model and generated audio responses. While generating the responses, we captured "debug traces" from the model by decoding the ASR and LLM internal representations to ASR Debug Tokens and LLM Debug Tokens, respectively. These tests allow us to gauge to extent of alignment between the model's internal representations and the input and output audio.

As shown in Table 2, the ASR Debug Tokens identically reproduce the content of the input audio queries, while the LLM Debug Tokens show precise alignment with the content of the output audio responses. This alignment demonstration confirms that the Neuroplex model is debuggable, which is a key model property needed in real-life deployments.

Table 2: Comparison of Input/Response Content with their respective ASR and LLM Debug Tokens

| **Input Content** | **ASR Debug Tokens** | **Response Content** | **LLM Debug Tokens** |
|---|---|---|---|
| What qualities do you value most in a partner? | [’What’, ’ qualities’, ’ do’, ’ you’, ’ value’, ’ most’, ’ in’, ’ a’, ’ partner’, ’?’] | For me, trust and empathy are crucial. They make everything else fall into place smoothly. | [’For’, ’ me’, ’,’, ’ trust’, ’ and’, ’ empathy’, ’ are’, ’ crucial’, ’.’, ’ They’, ’ make’, ’ everything’, ’ else’, ’ fall’, ’ into’, ’ place’, ’ smoothly’, ’.’] |
| Any tips for slicing meat thinly like they do at delis? | [’Any’, ’ tips’, ’ for’, ’ sl’, ’icing’, ’ meat’, ’ thin’, ’ly’, ’ like’, ’ they’, ’ do’, ’ at’, ’ del’, ’is’, ’?’] | Use a sharp knife and cut against the grain, then slice at a 45-degree angle for the best texture. | [’Use’, ’ a’, ’ sharp’, ’ knife’, ’ and’, ’ cut’, ’ against’, ’ the’, ’ grain’, ’,’, ’ then’, ’ slice’, ’ at’, ’ a’, ’ ’, ’45’, ’-degree’, ’ angle’, ’ for’, ’ the’, ’ best’, ’ texture’, ’.’] |
| The Greeks were pioneers in architectural design, right? | [’The’, ’ Gree’, ’ks’, ’ were’, ’ pione’, ’ers’, ’ in’, ’ architectural’, ’ design’, ’,’, ’ right’, ’?’] | Certainly, their use of columns, arches, and symmetry set the stage for countless modern designs | [’Certainly’, ’,’, ’ their’, ’ use’, ’ of’, ’ columns’, ’,’, ’ arch’, ’es’, ’,’, ’ and’, ’ symmetry’, ’ set’, ’ the’, ’ stage’, ’ for’, ’ countless’, ’ modern’, ’ designs’] |
| Can grassland fires be linked to climate change? | [’Can’, ’ grass’, ’land’, ’ fires’, ’ be’, ’ linked’, ’ to’, ’ climate’, ’ change’, ’?’] | Yes, increased temperatures and dry conditions increase the frequency and intensity of wildfires. | [’Yes’, ’,’, ’ increased’, ’ temperatures’, ’ and’, ’ dry’, ’ conditions’, ’ increase’, ’ the’, ’ frequency’, ’ and’, ’ intensity’, ’ of’, ’ wildfires’, ’.’] |
| Chess is a game I’ve always admired. Do you play often? | [’Ch’, ’ess’, ’ is’, ’ a’, ’ game’, ’ I’, ”’ve”, ’ always’, ’ adm’, ’ired’, ’.’, ’ Do’, ’ you’, ’ play’, ’ often’, ’?’] | I do! I love the competitive and strategic depth. Have you beaten any challenging stages? | [’I’, ’ do’, ’!’, ’ I’, ’ love’, ’ the’, ’ competitive’, ’ and’, ’ strategic’, ’ depth’, ’.’, ’ Have’, ’ you’, ’ beaten’, ’ any’, ’ challenging’, ’ stages’, ’?’] |
| I just joined a local book club. Do you enjoy book clubs? | [’I’, ’ just’, ’ joined’, ’ a’, ’ local’, ’ book’, ’ club’, ’.’, ’ Do’, ’ you’, ’ enjoy’, ’ book’, ’ clubs’, ’?’] | Yes, I do! They’re a fantastic way to meet like-minded people and discuss what interests. | [’Yes’, ’,’, ’ I’, ’ do’, ’!’, ’ They’, ”’re”, ’ a’, ’ fantastic’, ’ way’, ’ to’, ’ meet’, ’ like’, ’-minded’, ’ people’, ’ and’, ’ discuss’, ’ what’, ’ interests’, ’.’] |
| How do you handle the social pressure of being ’always on’? | [’How’, ’ do’, ’ you’, ’ handle’, ’ the’, ’ social’, ’ pressure’, ’ of’, ’ being’, ” ’”, ’al’, ’ways’, ’ on’, ”’”, ’?’] | Learning to say no and setting boundaries has been key for me. It reduces that pressure significantly. | [’Learning’, ’ to’, ’ say’, ’ no’, ’ and’, ’ setting’, ’ boundaries’, ’ has’, ’ been’, ’ key’, ’ for’, ’ me’, ’.’, ’ It’, ’ reduces’, ’ that’, ’ pressure’, ’ significantly’, ’.’] |
| What are the main barriers to achieving sustainability in business? | [’What’, ’ are’, ’ the’, ’ main’, ’ barriers’, ’ to’, ’ achieving’, ’ sustainability’, ’ in’, ’ business’, ’?’] | Barriers include financial constraints, lack of expertise, regulatory hurdles, and cultural resistance to change. | [’Bar’, ’riers’, ’ include’, ’ financial’, ’ constraints’, ’,’, ’ lack’, ’ of’, ’ expertise’, ’,’, ’ regulatory’, ’ hurdles’, ’,’, ’ and’, ’ cultural’, ’ resistance’, ’ to’, ’ change’, ’.’] |

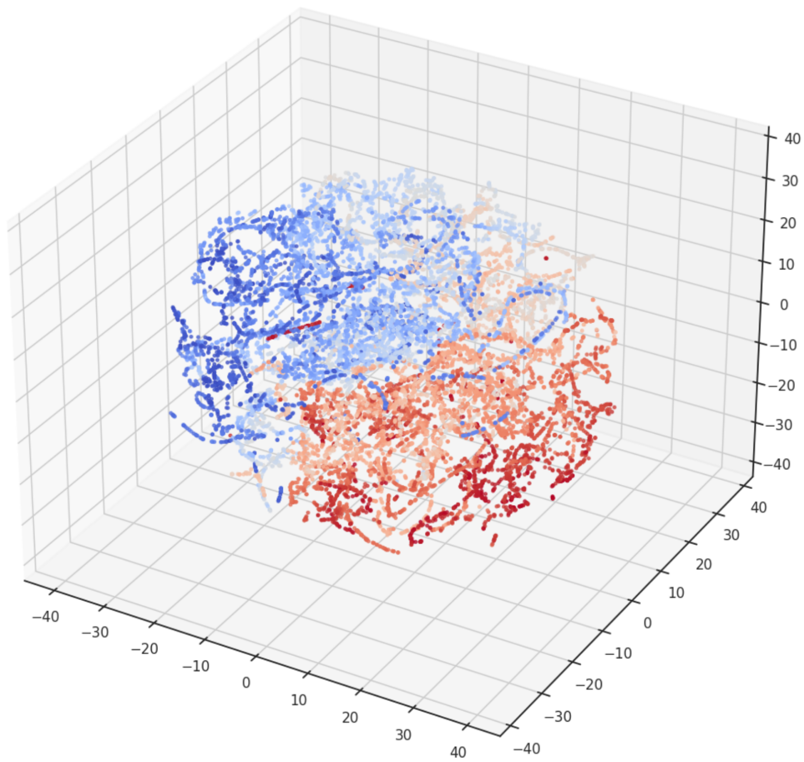


Figure 2: Visualization of the ASR2LLM adapter's latent space. The scatter plot shows t-SNE reduced embeddings of 15,000 acoustically perturbed variations of the word "hello", interpolated between three anchor recordings. Colors indicate the interpolation weights between anchor points. The continuous distribution demonstrates preservation of acoustic variations that would collapse to a single point in discrete cascade systems.

### 3.2 Latent Space Visualization

In order to study the properties of the Neuroplex latent space, we performed an analysis of embeddings at the ASR2LLM adapter output. We generated a controlled dataset by recording three anchor utterances of the word "hello" and creating 15,000 variations through interpolation between these anchors, applying minimal acoustic perturbations including noise, gain adjustments, and polarity inversions. The embeddings were extracted from the adapter output for samples where the ASR component correctly predicted the target token, then reduced to three dimensions using t-SNE for visualization. The resulting scatter plot reveals a rich latent manifold with distinct clusters and smooth transitions between acoustic variations of the same word, demonstrating that the continuous pipeline preserves subtle prosodic and acoustic information that would be lost in traditional cascade systems where identical transcribed tokens collapse to a single point.

## 4 Summary and Future Work

We have presented Neuroplex, a modular speech-to-speech architecture that combines pre-trained components through learned adapter networks. Through a carefully designed training curriculum and flexible loss weighting scheme, Neuroplex achieves end-to-end optimization while maintaining component specialization and interpretability. Our research prototype demonstrates successful question-answering capabilities in single-turn dialogue scenarios, with the model exhibiting coherent language understanding and high-quality speech synthesis.

A key advantage of Neuroplex's adapter-based design is its ability to bridge between specialized modules while operating directly on continuous hidden state representations. This approach preserves rich acoustic information throughout the pipeline while enabling precise control over model behavior through the LLM component. The architecture supports both pure end-to-end operation for optimal performance and modular inspection for debugging and monitoring.

Future work will focus on extending the model to complex multi-turn dialogues, real-time processing, and cross-lingual scenarios, while optimizing the architecture for scaled deployment. The modular yet end-to-end trainable nature of Neuroplex provides a promising foundation for building scalable, controllable voice agent systems.

## References


Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, Hao Wang, Wen Wang, Yuxuan Wang, Yunlan Xu, Fan Yu, Zhijie Yan, Yexin Yang, Baosong Yang, Xian Yang, Guanrou Yang, Tianyu Zhao, Qinglin Zhang, Shiliang Zhang, Nan Zhao, Pei Zhang, Chong Zhang, and Jinren Zhou. Minmo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.26062*, 2025.

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.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *CoRR*, abs/2409.06666, 2024.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. Language model can listen while speaking. *CoRR*, abs/2408.02622, 2024. doi: 10.48550/ARXIV.2408.02622. URL `https://doi.org/10.48550/arXiv.2408.02622`.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. Generative spoken dialogue language modeling. *CoRR*, abs/2203.16502, 2022. doi: 10.48550/ARXIV.2203.16502. URL `https://doi.org/10.48550/arXiv.2203.16502`.

Xiong Wang, Yangze Li, Chaoyou Fu, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, and Chaohong Tan. Omniflatten: An end-to-end GPT model for seamless voice conversation. *CoRR*, abs/2410.17799, 2024a.

Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao, Yuxuan Wang, Bin Zhang, Heng Lu, Yaqian Zhou, and Xipeng Qiu. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *CoRR*, abs/2410.08035, 2024b.