# Ichigo: V2T

[Demo] [Paper] [Blog] [Code] [Models] [Data] [Colab Demo]

- Dataset info is provided in detail
- Models are provided
- Training recipe is provided (Shady)

- Generates text as response to the audio, which is converted to Audio using TTS.

**Contributions:**

1. Ichigo is an tokenized early-fusion multimodal model capable of reasoning over and generating interleaved speech-text documents.
2. Training techniques without starting from scratch.
3. Recovering capability training method and techniques to stabilize cross-modality training.
4. Instruction Speech, a large-scale English speech-text cross-modal instruction-following dataset.
   a. featuring multi-turn interactions, reasoning tasks, and refusal scenarios.
5. The training and inference code.



**Tokenized early-fusion multimodal models:** Models sees each modality as tokens. Audio is seen as audio tokens.

- Quantizing speech into discrete tokens, allows us to use a decoder-only transformer architecture for both speech and text tokens, without adding a speech encoder and a speech adaptor.
- In early-fusion approach, all modalities are projected into a shared representational space from the start, allows for seamless reasoning and generation across modalities. However, it presents significant technical challenges, particularly in terms of optimization stability and scaling.
- Multimodal Models and Fusion - A Complete Guide

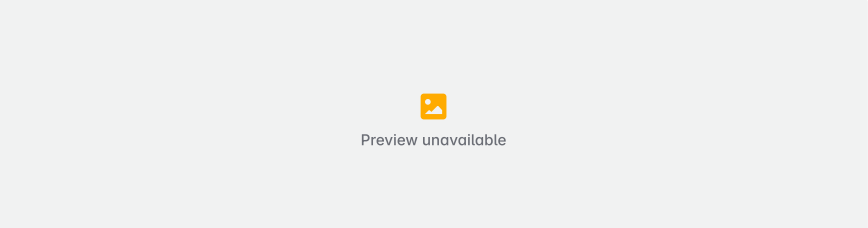
## 1. Model

### 1.1 Tokenization

Quantization of Audio and Tokens.

Speech Tokenization: WhisperVQ, a component of WhisperSpeech (Collabora, 2024).

- Codebook of 512 tokens with a codebook dimension of 64.
- Based on the Whisper Medium model, WhisperVQ processes speech input resampled to 16 kHz, achieving a frame rate of 25 Hz.
- Audio is converted to a log-mel spectrogram and processed by a Whisper encoder, producing continuous embeddings. These embeddings undergo downsampling and refinement before a vector quantization step maps them to a finite codebook, producing a sequence of discrete tokens representing the audio content.




Speech and text modalities as discrete tokens

**Expanding the Language Model**

Expand the vocabulary with new modality-specific tokens.

- This expansion necessitates extending the corresponding embeddings and prediction layer, with newly incorporated parameters initialized randomly.

### 1.2 Model Implementation

**Backbone Model:** Llama-3.1-8B-Instruct

- Pre-trained on 15 trillion text tokens
- Audio tokensare converted to the format <|sound_dddd|>, where 'dddd' represents the position of the corresponding code.
- Special tokens, <|sound_start|> and <|sound_end|>, to delimit audio file inputs.

**Initialization of new Token Embeddings**

- Default new token initialization from the HuggingFace codebase resulted in slow convergence of the loss curve.
- Initializing new token embeddings by averaging all embeddings of the current vocabulary significantly improved the speed of convergence and enhanced training stability.

## 2. Datasets

### 2.1 Pre-training Datatset

Facilitates LLM's understanding of audio signals.

To align the embeddings of text and audio, assembled a diverse collection of public Automatic Speech Recognition (ASR) datasets spanning eight languages: English, German, Dutch, Spanish, French, Italian, Portuguese, and Polish.

- English:  dataset (Pratap et al., 2020)
- Other languages:  dataset (Pratap et al., 2020).

| | Data | Duration |
|---|---|---|
| English | MLS English 10k | 10,000 hours |
| Other Languages | Multilingual LibriSpeech | 6,000 hours across languages |

Majority of the above datasets originates from public domain audiobooks like LibriVox and OpenSLR.

### 2.2 Post-training Datatset

Enables cross-modal instruction tuning.

#### 2.2.1 Text Instruction Data

Datasets:

- Magpie: High-quality alignment data by prompting aligned LLMs with their pre-query templates.
- HuggingFaceTB/everyday-conversations-llama3.1-2k: 2.2k multi-turn conversations generated by Llama-3.1-70B-Instruct
- PJMixers/Math-Multiturn-10K-ShareGPT · Datasets at Hugging Face
- euclaise/gsm8k_multiturn · Datasets at Hugging Face
- Intel/orca_dpo_pairs · Datasets at Hugging Face
- routellm/gpt4_dataset · Datasets at Hugging Face
- nomic-ai/gpt4all-j-prompt-generations · Datasets at Hugging Face
- microsoft/orca-math-word-problems-200k · Datasets at Hugging Face
- allenai/WildChat-1M · Datasets at Hugging Face
- Open-Orca/oo-gpt4-200k · Datasets at Hugging Face

- 🤗 Magpie-Align/Magpie-Pro-300K-Filtered · Datasets at Hugging Face
- 🤗 qiaojin/PubMedQA · Datasets at Hugging Face
- 🤗 Undi95/Capybara-ShareGPT · Datasets at Hugging Face
- 🤗 HannahRoseKirk/prism-alignment · Datasets at Hugging Face
- 🤗 BAAI/Infinity-Instruct · Datasets at Hugging Face

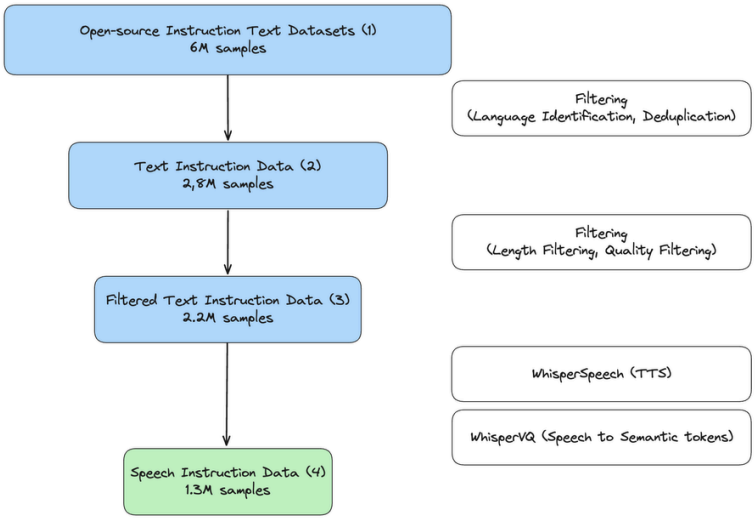
**Data Filtering:**

1. **Language Identification:** FastText model at the document level, retaining only English documents with a confidence threshold of (0.9).
   a. This decision aligns the model's distribution more closely with the original multilingual training of the base LLM.
2. **Deduplication:** Removed duplicate entries to prevent overfitting and ensure a diverse training set.
   a. Despite the tokenizer's capacity to handle eight languages, we opted to focus primarily on English for this training iteration.
   b. Scarcity of high-quality instruction data and low-resource nature of these languages.

#### 2.2.2 Speech-Text Instruction Data

Building upon the Text Instruction Dataset, we conducted further filtering to create a dataset more suitable for Instruction Speech Dataset.

- **Length Filtering:** Ffiltered out text instructions longer than 64 tokens.
  - Based on empirical observations of typical user interactions with audio assistants.
- **Quality Filtering:** Eliminated samples that would be challenging to pronounce as speech, such as URLs, mathematical symbols, and code snippets.
- **Synthetic Data Generation Pipeline:** Two-stage process to convert text-based Instruction dataset into discrete sound tokens suitable for audio input.
  - WhisperSpeech text-to-speech (TTS) model to generate audio files from the instruction dataset's questions.
  - WhisperVQ model to transform these audio files into discrete sound tokens.




- Process was applied only to the input questions, while the corresponding answers were maintained in their original text format.
- **2000 hours**: The resulting dataset comprised 2000 hours of tokenized speech audio data paired with text responses.

#### 2.2.3 Transcribe Data

- For transcription tasks, created a specialized transcribe instruction dataset derived from ASR dataset.
- Issue with special token <|transcribe|>: This approach led to catastrophic forgetting in the model.
- Pure instructions for transcription tasks: Improved the model's ability to map sound token patterns to corresponding text while minimizing the reduction in the model's text capabilities.

| Transcribe Prompts |
| --- |
| Transcribe the following audio clip: <speech> |
| Convert the spoken words to text: <speech> |
| What is being said in this audio clip: <speech> |
| Transcribe the speech in this audio sample: <speech> |
| Please write down what is being said in the audio clip: <speech> |
| Generate a transcript from this sound file: <speech> |

| |
|---|
| Recognize the speech in this audio clip: <speech> |
| Produce a text version of this audio recording: <speech> |

#### 2.2.4 Noise Audio Data

To prevent the model from being overly sensitive to inaudible inputs.

- Creating a synthetic dataset of random environmental noises proved challenging to scale.

Meaningful speech follows certain patterns and utilized this insight to generate inaudible input data.

- Using the 512 sound tokens from the WhisperVQ codebook, we randomized them into patterned sequences.
- This method allowed us to generate a vast amount of inaudible input data with a wide distribution.
- We then employed the Qwen2.5-72B model to generate diverse synthetic answers for those inaudible inputs.
  - These responses are essentially placeholder outputs generated by the model for training purposes, ensuring that the dataset covers a wide variety of possible outcomes.
  - Dataset Variability: By pairing inaudible inputs with diverse synthetic responses, the dataset becomes more comprehensive. This helps the model learn how to handle a wide range of non-speech inputs without overfitting to a specific type of inaudible noise.
  - Realism in Training: In real-world scenarios, the model will encounter unpredictable and diverse inaudible inputs. Generating synthetic responses simulates this variability, enhancing the model's robustness.
  - Model Generalization: Providing a wide array of responses ensures the model doesn't develop a rigid pattern or bias when encountering inaudible inputs. Instead, it learns to focus on identifying meaningful speech and ignoring irrelevant noise.

With an average speech input of about 50 sound tokens, there are 51350 possible arrangements, of which only a tiny fraction would constitute meaningful speech. By exposing our model to a wide range of these chaotic arrangements, we taught it to distinguish between audible and inaudible inputs effectively.

**Sequence length distribution matching between inaudible and audible data:**

- Ensure a balanced representation of both types of inputs in our training set.
- This approach involved sampling inaudible data samples to match the token count distribution of the original data, contributing to a more robust and generalizable model.

## 3. Training

Updates: 🤗🍓 Ichigo: Llama learns to talk | Menlo Research

| Parameter | Pre-training | Instruction FT | Enhancement FT |
|---|---|---|---|
| Weight Decay | 0.005 | | |
| Learning Scheduler | Cosine | | |
| Optimizer | AdamW Fused | | |
| Precision | bf16 | | |
| Hardware | 10x A6000 | 8x H100 | 8x H100 |
| Train time | 45h | 10h | 3h |
| Steps | 8064 | 7400 | 644 |
| Global batch size | 480 | 256 | 256 |
| Learning Rate | $2\times10^{-4}$ | $7\times10^{-5}$ | $1.5\times10^{-5}$ |
| Warmup Steps | 50 | 73 | 8 |
| Max length | 512 | 4096 | 4096 |

**Optimizers:** Adam-mini and Lion resulted in unstable training and frequent loss explosions.

**Infra:** A6000 with 48GB GPUs employing FSDP 2 and activation checkpointing.

### 3.1 Pre-training Methodology

Aimed to introduce speech representation into new tokens, facilitating the model's development of basic concepts regarding these additional tokens.

### 3.2 Post-training Refinements

#### 3.2.1 Instruction Fine-tuning

Focused on honing the model's question-answering capabilities

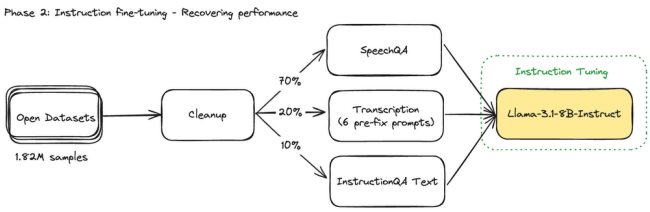
- Building upon the model from the previous stage, we concentrated on developing its question-answering abilities.

> Balancing modalities during the Supervised Fine-Tuning (SFT) stage is crucial to maintain the model's original performance.
>
> Significant imbalances between modality pairings could lead to unconditional priors, resulting in either muted or exaggerated generation of specific modalities.

**Data Recipe:**

- 70% speech instruction prompts
- 20% speech transcription prompts
- 10% text-only prompts

This distribution was determined through extensive permutation testing to achieve an optimal balance between speech understanding, transcription capabilities, and general language skills.




#### 3.2.2 Enhancement Fine-tuning

Expanded its proficiency in multi-turn conversations and appropriate responses to inaudible inputs.

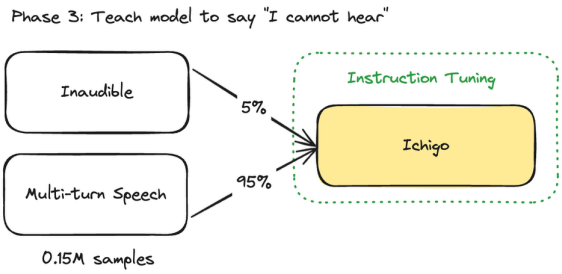
- The enhancement fine-tuning stage involved data augmentation to simulate real-world user interactions, thereby improving Ichigo's robustness in various scenarios.
- These enhancements aimed to create more fluid dialogues and improve the model's interactive capabilities.

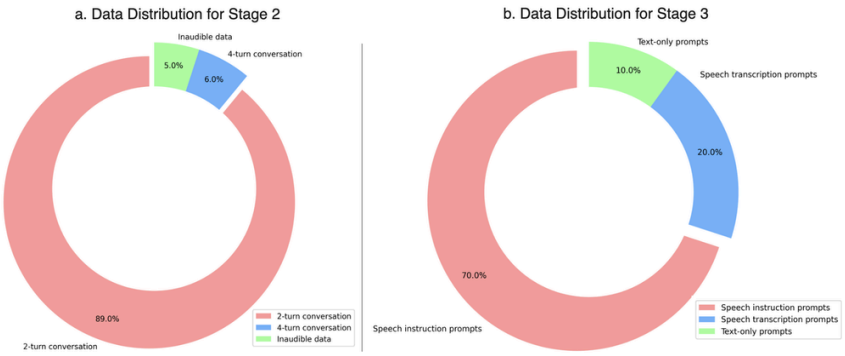
Focused on two key areas:

- multi-turn conversations with speech input
- appropriate responses to inaudible inputs

**Data Recipe:**

- 158,000 samples
- 0.5% of the total multi-turn data: Refusal capability examples
  - Higher percentage led to an increased tendency for the model to refuse inputs.



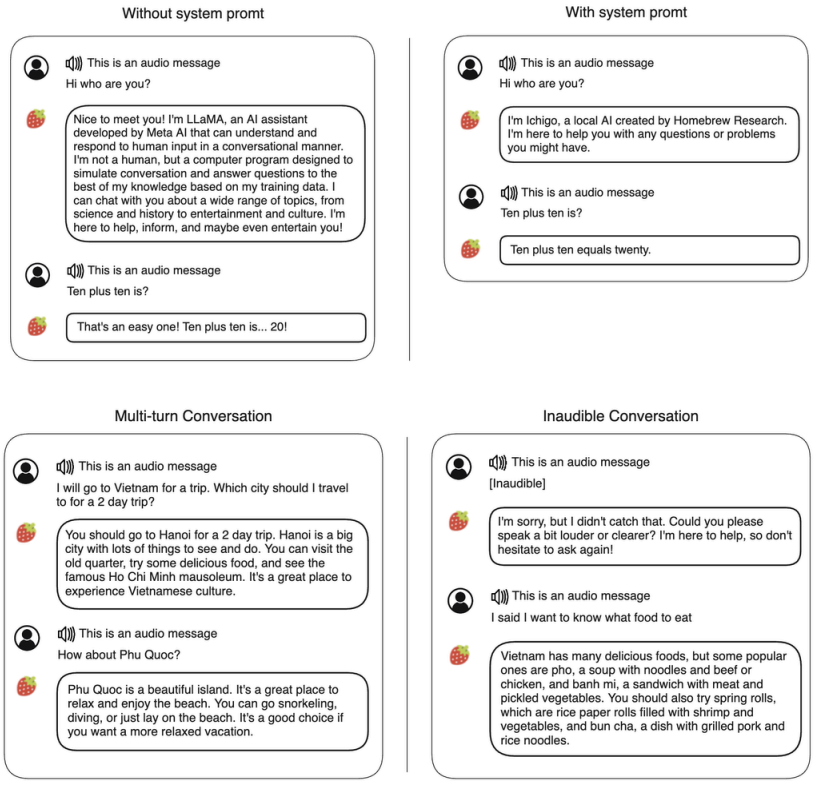



## 4. Results

**System Prompt**

> - You are Ichigo, a local realtime voice AI created by Homebrew Research. Your talk is also charming and friendly. Please say that when you are questioned against your identity. Otherwise please talk to the user like a helpful assistant.
>
> - Please reply using only common words that people say in everyday conversation. Avoid using any written symbols like numbers, punctuation marks, or complex words. Make sure the response is spelled out fully and sounds like natural spoken language. Even technical terms, code, or numbers should be spoken out fully. For example:
>
> Person A: How do you write a for loop in C plus plus?
> Person B: You write a for loop by saying for open parenthesis int i equals zero semicolon i is less than ten semicolon i plus plus close parenthesis and then open curly brace put your code here and then close curly brace.
>
> Person A: What is two plus two?
> Person B: Two plus two equals four.

**Examples:**



## 5. Related Works

### Non-Tokenized Early Fusion

The most common approach to enable cross-modal perception in LLMs is to connect pre-trained encoders of other modalities as adaptors. This method involves adding a speech encoder before the LLM and fine-tuning the entire model for speech understanding capabilities.

- Qwen2 Audio (Chu et al., 2024) introduce the new architecture to combine an audio encoder with a large language model, training to maximize next text token probability conditioned on audio representations.
- This NTEF tends to be more cost-effective, as it involves multiple training phases where most components are frozen, and it can be effective even when training with Parameter-Efficient Fine-Tuning techniques (Hu et al., 2021).

### Tokenized Early Fusion

This approach involves tokenizing multimodal inputs using either a common tokenizer or modality-specific tokenizers.

- Chameleon, which represents images and text as a series of discrete tokens within a unified transformer, trained from scratch with modified transformer architecture.
- AudioPALM (Rubenstein et al., 2023) and VoxTLM (Maiti et al., 2024), which utilize pre-trained language models and extend their vocabularies with discrete semantic audio tokens, focus on translation speech to speech tasks.
- AnyGPT (Zhan et al., 2024) leverages LLMs to enable inherent cross-modal conversation capabilities through SpeechTokenizer (Zhang et al., 2023), MusicTokenizer (Défossez et al., 2022), and ImageTokenizer (Ge et al., 2023).
- Moshi (Défossez et al., 2024) is a real-time native multimodal foundation model designed for seamless audio-text interactions. It employs a 7B parameter multimodal language model that processes speech input and output concurrently, generating text tokens and audio codecs. Moshi's innovative approach allows it to handle two audio streams simultaneously, enabling it to listen and talk in real-time while maintaining a flow of textual thoughts.

## 6 Limitations and Future work

While Ichigo represents a significant step forward in multimodal language modeling, several limitations and areas for future work remain:

- **Token Stability**: Similar to challenges faced by models like Chameleon, we encountered fluctuating loss when training with acoustic tokens, which led us to shift towards semantic tokens to achieve stable loss. This highlights the difficulty in training with rich, acoustic information. Future work should explore methods to stabilize training with acoustic tokens, potentially unlocking even more powerful models.
- **Emotional Understanding**: The current architecture does not fully account for emotional comprehension. Future iterations should focus on enhancing the model's ability to understand and respond to user emotions, allowing for more nuanced and context-appropriate responses.
- **Context Length**: Multimodal content, especially audio, often spans extensive sequences. Ichigo currently limits modeling to 10 seconds of speech input and performs well for 4-5 turns of conversation. Extending the context window would allow for modeling of longer audio segments and handling of more complex, multi-turn conversations.