

Supplementary Materials: VoxInstruct: Expressive Human Instruction-to-Speech Generation with Unified Multilingual Codec Language Modelling

Anonymous Authors

A DEMO WEBPAGE

The webpage displays generated samples from VoxInstruct, and we highly recommend that the reviewers take a listen. We provide more samples and situations than the experimental results in the paper. The webpage is available at: <http://voxinstruct.github.io/VoxInstruct>. Please open the demo webpage in *Chrome* for an enhanced experience.

B SPEECH ANNOTATION SYSTEM (DATASET)

In section 5.1 of the paper, we annotated raw speech corpus with a novel automatic speech annotation system for building the instruction-speech paired dataset. The speech annotation system we used can be referred to in the attachment “[speech_annotation_system.pdf](#)” in supplementary materials, which describes how to design an automatic speech annotation system for expressiveness interpretation that annotates in-the-wild speech clips with expressive and vivid human language descriptions. Speech audios are firstly processed by a series of expert classifiers and captioning models to capture diverse speech characteristics, followed by a fine-tuned LLaMA for customized annotation generation. This dataset work is the data premise of our proposed VoxInstruct, and has also been *anonymously submitted* to ACM MM 24 as concurrent work. It should be noted that, our dataset includes publicly available parts from this dataset paper as well as some internal data and crawled data, and the annotation process is the same as this paper.

C DETAILED EXPERIMENTAL SETTINGS

C.1 Model configuration

Our VoxInstruct consists of an Encodec acoustic encoder, an MT5 text encoder, an AR codec language model, a NAR codec language model, and a Vocos-based vocoder. We use the official models and parameters of Encodec and Vocos with the configuration of 24 kHz. The MT5-base text encoder comprises 12 transformer blocks with 768 hidden sizes. We insert the trainable LoRA adapters with $r = 16$ and $\alpha = 16$ into this pre-trained text encoder. The text encoder supports multilingual text input and encodes the original text into a sub-word-level text embedding sequence. Both the AR and NAR codec language models are a 12-layer transformer with 1024 hidden sizes and 4096 feed-forward network dimensions. The flash attention module, rotary positional embeddings and SwiGLU activation functions are used in codec language models. The maximum sequence length of codec language model is set to 2560, where the instruction text embedding sequence occupies 512, and the ST and AT sequences occupy 2048. Padding is used to fill the shortfall of text embedding sequence part. We provide detailed hyper-parameter settings about the model configuration in Table 1.

Table 1: Hyper-parameters of VoxInstruct model

Module	Item	Scale/Size
MT5-base Text Encoder	Vocab Size	250112
	Encoder Layers	12
	Hidden Size	768
	FFN Intermediate Size	2048
	LoRA r	16
	LoRA α	16
Codec Language Model AR (NAR)	Vocab Size	1530 (12240)
	Decoder Layers	12
	Hidden Size	1024
	FFN Intermediate Size	4096
	Attention Heads	16
Total num. of parameters		709M
Total trainable num. of parameters		432M

C.2 Inference settings

Using different inference sampling strategies in the codec language models of VoxInstruct affects the generated speech results, and the CFG values of our proposed multiple CFG strategies also impact speech performance.

In the AR model, we use a greedy strategy for generating the semantic token (ST) sequence to ensure content accuracy. For the generation of the coarse-grained acoustic token (AT) sequence, we aim to increase diversity, so we use a combined sampling strategy of topK ($K = 50$) and TopP ($P = 0.95$). In the NAR model, we use greedy sampling in each forward pass, which means taking the maximum probability of token candidates as the confidence score for each position.

For the setting of CFG values, we adjust them based on the speech performance required by the specific task. For instance, when conducting human instruction-to-speech generation, we choose $\gamma = 1.5$, $\alpha = 3.0$, $\beta = 1.5$, which strongly emphasizes the adherence of generating coarse-grained AT to the human instruction, enhancing style control over speech. Conversely, when conducting voice cloning, we select $\gamma = 1.0$, $\alpha = 1.0$, $\beta = 1.5$, slightly increasing the adherence of generating AT to ST, to stabilize the content of the synthesized speech. The selections of CFG values are only rough references; we did not finely adjust these values to obtain the best results in our experiments.

C.3 Implementation details in subjective evaluations

Screenshots of three subjective MOS test systems are shown in Fig. 1 (MOS-I), Fig. 2 (MOS-Q), and Fig. 3 (MOS-S). The text parts in the screenshots provided the human instructions corresponding to each sample and explained how participants should rate the audio samples. Participants can select a score from 1 to 5 for each audio

sample from the specific aspect. Additionally, for the MOS-S test, speech prompts are provided to assist in comparing the speaker’s similarity to the reference voice.

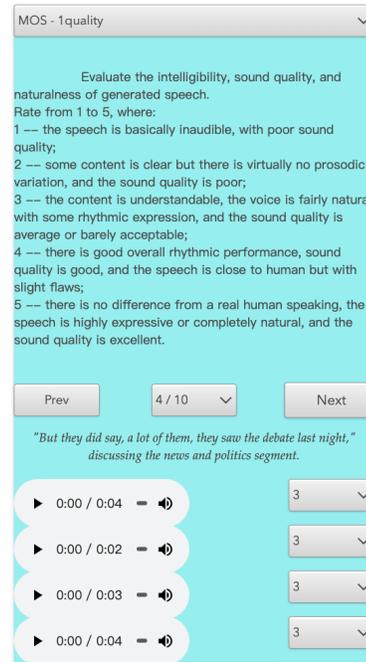


Figure 2: The screenshot of MOS-Q test system.

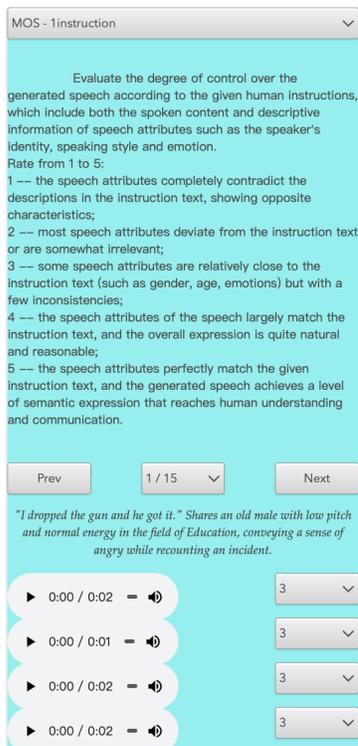


Figure 1: The screenshot of MOS-I test system.

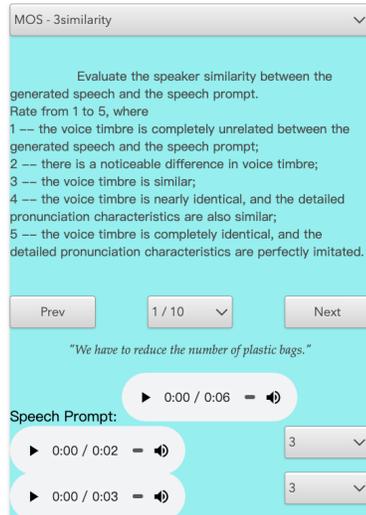


Figure 3: The screenshot of MOS-S test system.