Takin: A Cohort of Superior Quality Zero-shot Speech Generation Models

Everest Team, Ximalaya †

Abstract

With the advent of the big data and large language model era, zero-shot personalized rapid customization has emerged as a significant trend. In this report, we introduce Takin AudioLLM, a series of techniques and models, mainly including Takin TTS, Takin VC, and Takin Morphing, specifically designed for audiobook production. These models are capable of zero-shot speech production, generating high-quality speech that is nearly indistinguishable from real human speech and facilitating individuals to customize the speech content according to their own needs. Specifically, we first introduce Takin TTS, a neural codec language model that builds upon an enhanced neural speech codec and a multi-task training framework, capable of generating high-fidelity natural speech in a zero-shot way. For Takin VC, we advocate an effective content and timbre joint modeling approach to improve the speaker similarity, while advocating for a conditional flow matching based decoder to further enhance its naturalness and expressiveness. Last, we propose the Takin Morphing system with highly decoupled and advanced timbre and prosody modeling approaches, which enables individuals to customize speech production with their preferred timbre and prosody in a precise and controllable manner. Extensive experiments validate the effectiveness and robustness of our Takin AudioLLM series models. For detailed demos, please refer to`https://everest-ai.github.io/takinaudiollm/`.

1 Introduction

Recent advancements in large language models (LLMs) [1, 2, 3, 4], neural codecs [5, 6, 7, 8], and diffusion and flow models [9, 10, 11, 12, 13] have led to significant progress in the fields of zero-shot text-to-speech synthesis (TTS) [14, 15, 16, 17, 18], voice conversion (VC) [19, 20, 21, 22], and related areas. These innovations enable the synthesis of high-quality speech without extensive model training, thereby enhancing the accessibility and scalability of these technologies and fostering more natural and immersive user interactions.

In this context, to drive innovation and support audiobook production, we propose Takin AudioLLM—a series of models designed to allow users to customize speech content according to their specific needs while generating high-quality, near-human-like speech with exceptional naturalness and expressiveness. The Takin AudioLLM series comprises Takin TTS, Takin VC and Takin Morphing.

Firstly, inspired by the powerful contextual learning capabilities of LLMs, we present Takin TTS—a robust and effective neural codec language model for audiobook production. Takin TTS incorporates a high-fidelity, low-bandwidth neural speech codec based on efficient disentangled prompt encoders, which reduces modality heterogeneity between text and audio, thereby enhancing the LM's prediction accuracy. We introduce a five-stage multi-task training strategy that significantly improves overall LM performance, ensuring robustness and effectiveness in complex real-world scenarios. Additionally, we employ a latent diffusion model and Vocoder for token-to-speech synthesis, further improving speech quality and naturalness. Consequently, Takin TTS excels in generating high-quality, natural-sounding speech for various applications, from interactive voice response systems to sophisticated text-to-

speech frameworks. This approach greatly enhances user experience and demonstrates substantial potential in advancing generative speech modeling technology.

Secondly, Takin-VC employs a joint modeling approach that integrates timbre features with both supervised and self-supervised content representations to enhance speaker similarity and intelligibility. This design allows Takin-VC to effectively capture and reproduce the nuanced characteristics of various speakers, ensuring that converted voices closely resemble the target speakers. Furthermore, to refine speech quality and naturalness, we incorporate an efficient conditional flow matching-based decoder. This advanced decoder optimizes the alignment between timbre and content features, leading to more accurate and natural voice conversion. In this way, Takin-VC provides a powerful and versatile tool for voice conversion applications, excelling in producing high-fidelity, natural-sounding voice conversions suitable for audiobook production. It significantly enhances user experience and demonstrates its potential to advance the field of voice conversion technology.

Finally, Takin Morphing introduces an attention mechanism-based multi-reference timbre encoder for precise and detailed timbre modeling. Additionally, a language model (LM)-based prosody encoder is employed to capture prosody representations that align with timbres for unseen speakers in an auto-regressive manner. To further enhance waveform quality, we advocate a two-stage information-flow-based training method. Through these innovations, Takin Morphing enables users to utilize timbres from various unseen speakers and combine them with preferred prosody styles, thus generating personalized audiobooks with a high degree of control. This capability meets the demands of diverse speech synthesis applications, from entertainment and education to commercial contexts, offering a more natural and enriched auditory experience.

Overall, Takin AudioLLM represents a significant advancement in zero-shot speech production technology. By leveraging the sophisticated capabilities of Takin TTS, Takin VC, and Takin Morphing, this series not only advances the state-of-the-art in speech synthesis but also addresses the growing demand for personalized audiobook production, enabling users to tailor speech generation precisely to their requirements.

2 Takin TTS

2.1 Overview

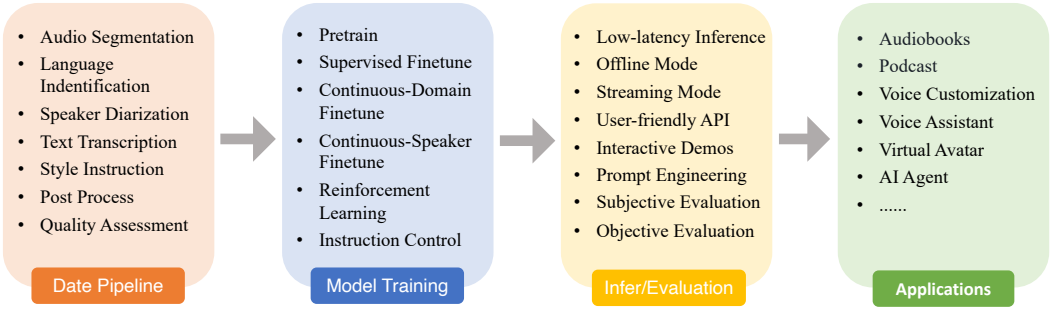

Figure 1: An overview of Takin TTS workflow

As shown in Figure 1, we take Takin TTS as an example to introduce the construction scheme of this series of large models, primarily including the construction of large-scale datasets, model training for specific tasks, the establishment of evaluation systems for voice generation models, and the commercialization of applications. Additionally, Figure 2 illustrates the overall training process of Takin TTS, and the specific training details will be gradually expanded upon as follows.

2.2 Pretrain

We use multimodal data to pretrain the Takin TTS. Specifically, we encode text and audio data into tokens and input them into the GPT model to learn relevant knowledge. For text data, we develop an internally developed G2P (Grapheme-to-Phoneme) method. This solution includes a Text

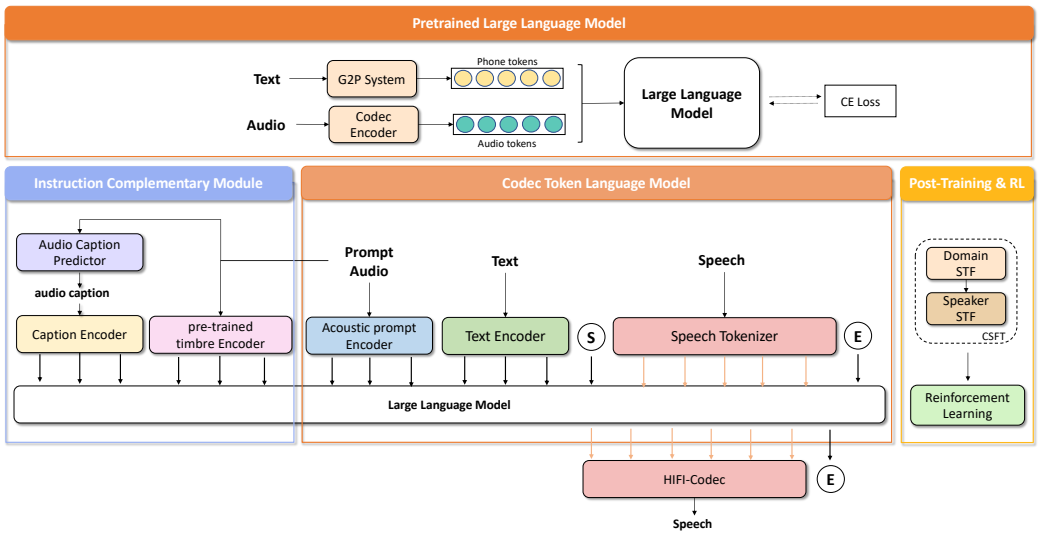

Figure 2: Overall schematic diagram of Takin TTS.

Normalization (TN) module, a Named Entity Recognition (NER) module, as well as a polyphone disambiguation module, which can convert text into phonemes and subsequently embed them into the lexical merge space. For audio data, we train an encodec system with a single codebook to convert audio content into discrete codec tokens. In this way, the multimodal data can be input into the GPT model for its understanding.

$$\hat{x}_t = \arg\max_{x_t} P_\theta(x_t \mid x_1, x_2, ..., x_{t-1}) \tag{1}$$

We use the most classic GPT training method, assuming the audio or text sequence is $X = \{x_1, x_2, ..., x_t\}$, with the pretraining objective as shown in equation 1.

2.3 Supervised Fine-tuning (SFT)

Following unsupervised learning on extensive data, our Takin TTS has developed a robust capacity to comprehend text and audio information. In the subsequent phase, akin to GPT-4 [2], we employ labeled paired data to train the Takin TTS model for downstream tasks such as TTS and Automatic Speech Recognition (ASR) [23, 24, 25], thereby enhancing its proficiency in managing text and speech tasks.

In the TTS task, zero-shot is a quite important capability of applications that requires the model to synthesize high-quality speech for unseen speakers without collecting their labeled data for training in advance. In this work, leveraging the ability of neural codec to convert speech into discrete tokens, the zero-shot TTS task is regarded as a conditional language modeling task to predict discrete codec tokens autoregressively based on given conditions.

Let $D = \{T_i, P_i, S_i\}$ denotes the training dataset, where S_i is the target speech, T_i is the text description and P_i is prompt audio which is from the same speaker with S_i. During the training process, a set of speech conditions SC_i is extracted from prompt audio P_i via acoustic prompt encoder, the text transcription T_i is converted to a phoneme sequence $TP_i = \{BP, P_{i_1}, P_{i_2}, ..., P_{i_m}, EP\}$, and BP stands for the Begin of Phone Sequence, EP stands for the End of Phone, the target speech is passed to neural codec model to get discrete codec tokens $C_i = \{C_{i_1}, C_{i_2}, ..., C_{i_n}\}$. A start identifier s and an end identifier e are inserted at the beginning and the end of codec tokens. The input training sequence is constructed as follows:

$$[SC_i, TP_i, S, C_i, E]$$

As shown in Figure 2, the language model is only trained to predict codec tokens and the end of sequence token E conditioned on phone sequence TP_i and speech conditions SC_i, which is

formulated as:

$$P(C_i|S, SC_i, TP_i) = \prod_{t=1}^{n+1} P(C_{i_t}|C_{i_{<t}}, S, SC_i, TP_i) \tag{2}$$

where C_{n+1} denotes the end of sequence token E. During inference, the language model generates tokens autoregressively based on given text and reference speech, and fed these tokens to neural codec model to generate audio.

2.4 Continual Supervised Fine-tuning (CSFT)

While Supervised Fine-Tuning (SFT) has endowed the Takin TTS model with TTS capabilities, the diverse content standards generated by TTS often lead to more frequent word omissions in Autoregressive (AR) models compared to Non-Autoregressive (NAR) models during inference [26, 27]. As a consequence, to enhance the stability of the system's TTS functionality, further Continual Supervised Fine-tuning (CSFT) joint with ASR guided training is necessary. In our method, CSFT primarily consists of two components: Domain-SFT and Speaker-SFT, which will be elaborated below.

2.4.1 Domain SFT

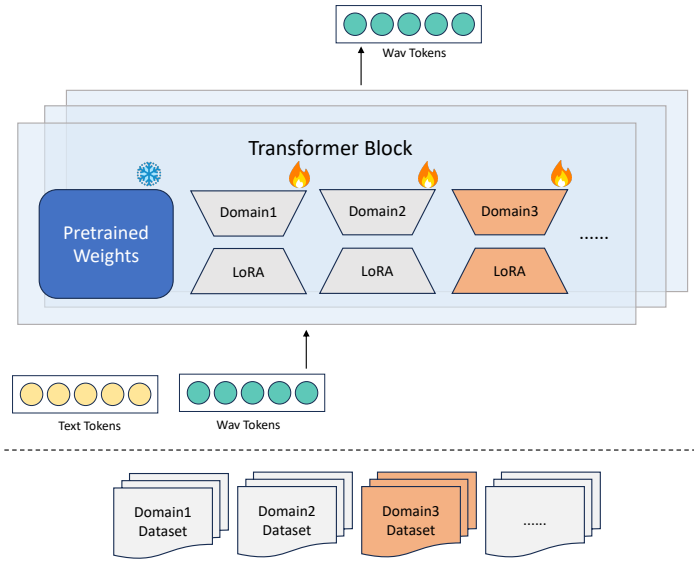

Figure 3: LoRAMoE for Domain Fine-tuning

The distribution of speech prosody diverses across various scenarios, for instance, the overall prosody of reading an audiobook is far different from that of delivering a speech. Since there usually exists an imbalanced data during pretraining phase, in order to improve the quality and accuracy of generated speech, Domain SFT is applied to our well fine-tuned models. In this phase, we only select several thousand hours of high-quality finely labeled domain data and train the propsoed approach using LoRA [28].

2.4.2 Speaker SFT

To ensure that the narration of high-quality audiobooks sounds more natural and aligns closely with the original speaker's performance style, we have further introduced the Speaker SFT Phase. In this section, we continue to use the LoRA training method. The difference here is that we freeze most of the GPT parameters to retain the model's foundational knowledge and update the parameters of the Acoustic Prompt Encoder with the Input and Output Embedding Layer parts of GPT.

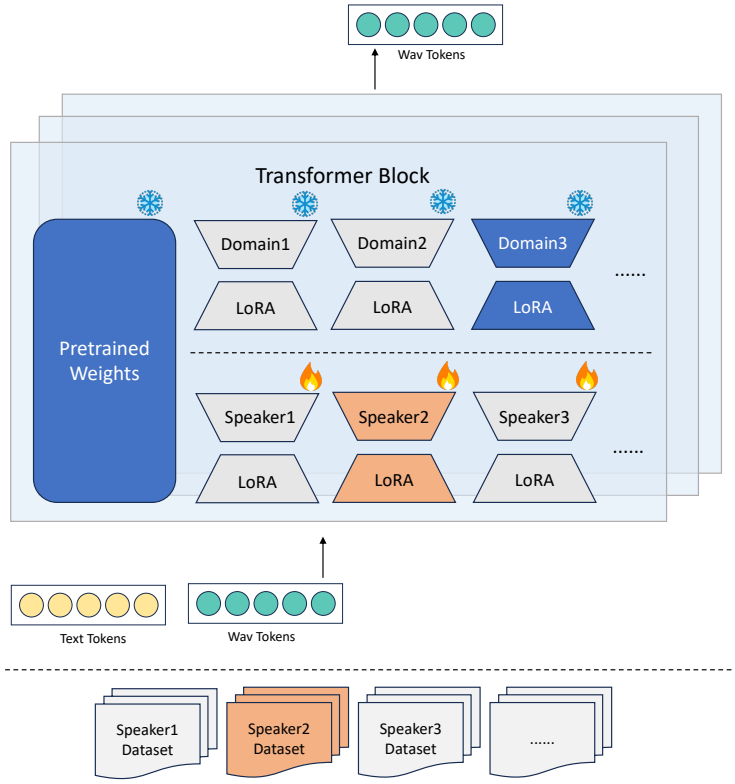

Figure 4: LoRAMoE for Speaker Fine-tuning

2.4.3 ASR guided Joint Training

To improve the accuracy of the model's output content, we incorporate ASR guidance into the model training during the finetuning process. The sequences output by GPT are fed into a codec decoder to be restored to wav format. To ensure gradient propagation and training speed, the generated wav is input into the whisper model, and its output is compared with the annotations to calculate the cross-entropy loss.

2.4.4 Reinforcement Learning

Despite the fact that the model after CSFT Process performs quite well, even surpassing human rendition levels for certain sentences by some speakers, it still faces issues with varying effectiveness among different speakers for the same text, as well as discrepancies between human and machine aesthetics. To make the generated content as closely aligned with human preferences as possible, we have introduced the concept of our RL (Reinforcement Learning) method. As shown in Figure 2, The RL is placed in the end of the whole diagram of Takin TTS to further improve the performance by aligning the model with human preference.

Currently RL methods [29, 30, 31, 32], follow a Sampling-human-annotating-learning pipeline, in which human evaluation is applied to model-generated outputs to ensure the model learns to align with subjective human preferences. The pipeline works also in speech generation task [33]. However the human ratings is labour dmanded, there are also some works studying to [34, 35] use objective metrics to replace human ratings, in order to facilitate the obtaining of preference data pairs. We also explore leveraging the human-rating only pipeline to combine it with a set of objective metrics which partly indicate human preferences, namely the Sampling-human&machine-annotating-learning pipeline.

2.4.5 Instruction Style Control

In AudioLLM paradigm, the common method of controlling speech style involves selecting different audio prompts, which generally incorporate both speaker identity and style information simultaneously. To explore the full potential of controllability, we propose TakinTTS-Instruct to synthesize speech with various styles and emotions, including rhythm, pitch, paralinguistics, etc., using natural language as style prompts which is more user-friendly than the base model of Takin TTS and could decouple the speaker and style in synthesis.

The lower-left part of Figure 2 prominently displays the core structure of TakinTTS-Instruct. To be specific, a robust pre-trained speaker verification system [36] is employed to provide additional voice characteristics, enhancing the similarity between the synthesized voice and the target speaker. Moreover, unlike previous speech emotion or speaking state recognition tasks [37, 38, 39, 40], in order to control the emotions of the generated speeches, speaking states, or other linguistic dimensions in a more user-friendly fashion, we implement a predictor that detects and classifies emotions or different speaker states in spoken language and subsequently outputs its corresponding natural language description. Furthermore, these descriptions will be parsed by SimBERT[41] into an embedding form to be incorporated into the model training.

3 Takin VC

In addition to TTS, another widely used technology in the audiobook business is VC technology. Here, we propose a novel and effective zero-shot VC approach based on DDPM or CFM. Similar to the usage conditions of TTS technology, it can achieve high-expressiveness timbre conversion with only 5-10 seconds of unseen audio.

3.1 VC Training

The input of Takin VC is composed of two parts: Phonetic Posteriorgrams (PPG), utilizing the output features of HybridFormer [42] in this case, and a truncated prompt mel fragment. For the output, Takin VC offers two alternatives: it can either produce mel spectrograms, which are subsequently converted to audio samples through a vocoder, or directly generate audio samples.

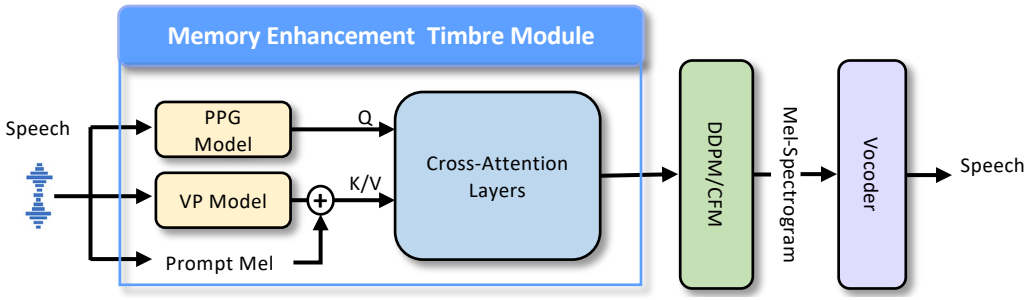

Figure 5: The Structure of Takin VC.

To elaborate, our Takin VC system mainly consists of three components: the PPG model is used to extract the decoupled content information of the input audio, and the memory augmented cross-attention based timbre modeling mechanism is used to re-populate the timbre information. Finally, we use the way of DDPM / Flow Matching to restore the converted spectral information and employ a HiFi-Gan vocoder to render it into the converted audio. Specifically, the PPG Model used here is a pre-trained Hybridformer model. As for the voiceprint model, we use a pre-trained CAM++[36] model from modelscope. The target features of DDPM / CFM here use a 129-dimensional mel-filterbank, and the prompt mel does the same. Due to the difficulty of collecting VC data from real world, similar to other VC system, we directly use normal speech data by cut of single speaker, and randomly select a segment as the prompt mel. We trained our models on 500k hours of data. During training , the PPG model and voiceprint model are frozen and only update the memory augmented timbre blocks and DDPM / CFM model are updated.

3.2 VC CSFT

Similar to parts of the Takin TTS, after pre-training on a large amount of data, fine-tuning with a small amount of high-quality data can enhance the model's performance. Furthermore, although the timbre information retained in the PPG feature is already minimal, there are still minor timbre leakage issues, resulting in suboptimal timbre conversion similarity in some cases. Therefore, we employed the TTS system to improve the performance in this regard. Due to the duration control issues, we used a traditional TTS system here to generate a small amount of parallel data, which is used to better guide the model in understanding the speech conversion task.

4 Takin Morphing

Audio Style Transfer is an important application in the field of audiobook production, which involves transforming styles while retaining the speaker's vocal timbre, thereby lowering the barrier to becoming a professional broadcaster. By using this technology, works by enthusiasts who are not yet proficient in certain broadcasting techniques can be transformed to have the style of professional broadcasters, thereby improving the quality of the works to some extent. Alternatively, it can serve as an auxiliary in teaching, guiding enthusiasts to develop their own unique broadcasting styles. Here, we introduce the Takin Morphing technology, which utilizes the form of DDPM to achieve style and rhythm transfer while maintaining the speaker's vocal timbre.

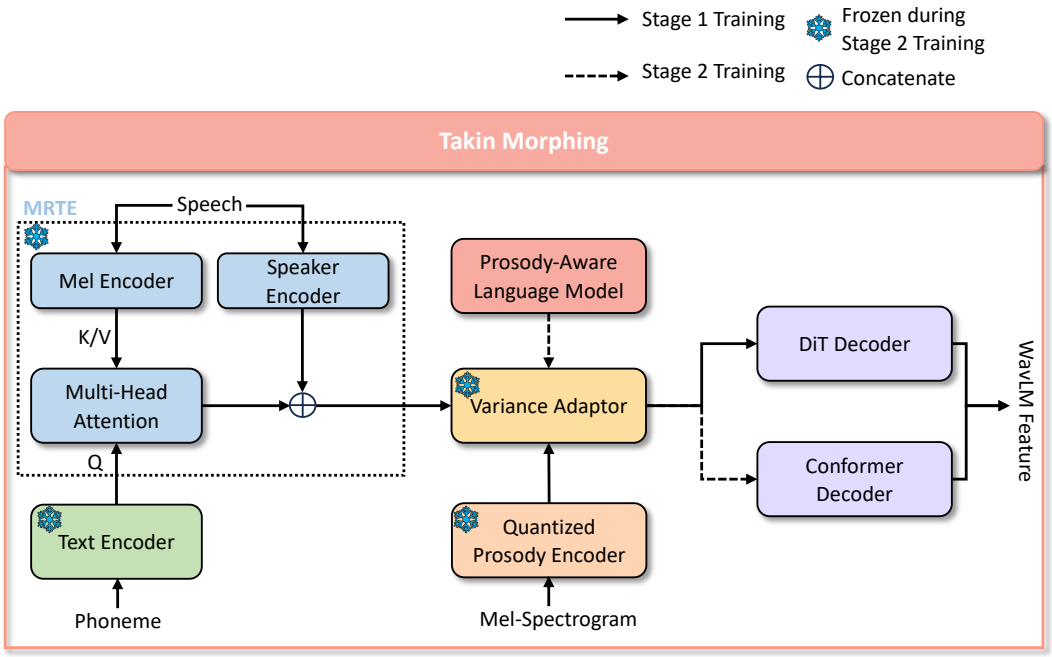

Figure 6: Training Structure of Takin Morphing.

As shown in the Figure 6, the input to the model is the Phone sequence. After feeding it into the Multi-Reference Timbre Encoder (MRTE) layer [43], we obtain a hidden matrix containing content and timbre information. This matrix, along with prosodic features, is sent into the Decoder model to restore the mel-like features. The prosodic features are VQ vectors of low-frequency Mel. Here, we replaced the commonly used mel-spectral features with wavlm features because our experiments show that wavlm features contains more expressive information than Mel Spectrum.

5 Experiments

5.1 Takin TTS Settings

Experimental Datasets To train and evaluate Takin TTS, We build a large multilingual base dataset for pretraining and 1st round of SFT training. To evaluate CSFT and RL training, several carefully human labeled datasets are built which including domain datasets and speaker datasets. The datasets are depicted as follows:

- **Base TTS Dataset**: In-house dataset including over 1M hours of speech data, which may include some labelling errors.
- **Domain TTS Dataset**: The domain dataset is of high-quality dataset with all the transcripts manually checked. There are two domains exist in the domain dataset which are audiobook and podcast. There is around 1000 hours for each domain used for Domain SFT. For speech data of each domain, 5% of the whole dataset is held out for test purpose and we make sure the held out test set does not have speaker overlap with the train set.
- **Speaker TTS Dataset**: To conduct speaker SFT based on the result of Domain SFT, a small Speaker TTS Dataset is constructed by selecting two audiobook-domain speakers and two podcast-domain speakers from our in-house dastaset. There is 1-hour speech data for each speaker, and likewise their transcripts are carefully labeled. For each speaker, 5% of speech data is held out as test set.

Evaluation Metrics To conduct objective evaluations, We employ the Phoneme Error Rate (PER) and Speaker Similarity (SIM) metrics. For PER, we pick Whisper-large-v3 [44] as the ASR model to conduct the PER test, While for SIM, we use CAM++ on the speaker verification task [45] to obtain speaker embeddings for calculating the cosine similarity of speech samples of each test utterance against reference clips. For subjective evaluations, We employ the Mean Opinion Scores (MOS) by rating different speech samples of the same content by human evaluators. The scores vary from 1 to 5 and the higher score indicates better speech quality. Besides, Bad Case Rate (BCR) is used to evaluate the overall stability of our models in RL experiments. Equation 3 is defined to compute BCR, in which B is the number of bad cases. To count the number of bad cases, We count the occurances of three types of bad cases covering prosody, pronunciation and missing or extra speech.

$$BCR = \frac{B}{100} \tag{3}$$

5.1.1 Pretraining

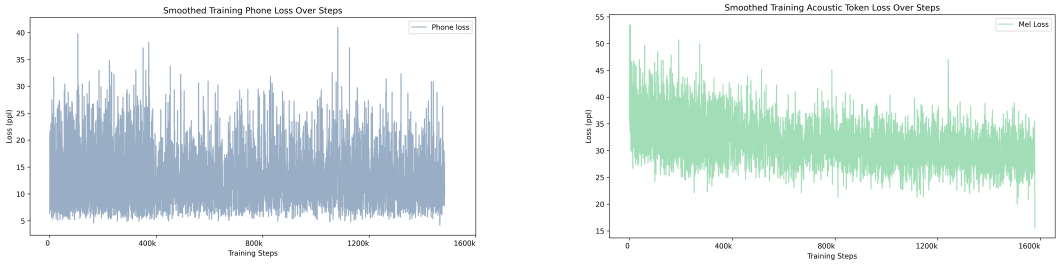

Figure 7: Training Loss of Phone and Acoustic Tokens.

As shown in the Figure 7, during the pretraining process, the phone loss can converge quickly. In comparison, the learning of acoustic tokens is slower, seemingly because the acoustic tokens contain more information and are more difficult to learn. Although we believe that if one only wants to perform TTS tasks, pre-training may not be a necessary option, and starting from the CSFT stage can also train a model with very good results. However, if you want the model to expand other multimodal capabilities, such as GPT-4o, pretraining is also a good choice.

5.1.2 CSFT on Takin TTS

After pretraining, Talkin TTS model is finetuned with *Base TTS dataset* to align the model to TTS task, namely SFT. However, this finetuned model is not prepared for real applications in terms of its stability and expressiveness as mentioned in section 2.4. CSFT is key to getting a stable model and enhance its expressiveness, especially when generating speech of a specific speaker. This section is mainly focused on the experiments of two types of CSFT which are domain SFT and speaker SFT.

Table 1: Evaluating results of domain SFT

Model	PER(↓)	SIM(↑)	MOS(↑)
w/o domain SFT	5.6	0.70	4.12 ± 0.09
w/ domain SFT (cross-domain test)	3.3	0.69	4.18 ± 0.07
w/ domain SFT (in-domain test)	**2.8**	**0.71**	$\mathbf{4.29 \pm 0.06}$

Domain SFT, unlike the full-parameter fine-tuning of CSFT, trains extra LoRA parameters of Takin TTS model, keeping backbone frozen. *Domain Dataset* is used to do domain SFT on audiobook and podcast domains respectively, which consequently results in two separate LoRA models of those two domains. The experimental results are demonstrated in Table 1, the model after SFT is denoted by *w/o domain SFT*, and *w/ domain SFT* denotes the SFT model further finetuned with domain SFT. All the objective metrics are computed in a zero-shot setup using test set in *Domain TTS Dataset*. To conduct subjective evaluation, 8 males and 8 females are randomly selected from the test set to synthesize speech, with 30 subjects participated in to rate scores from 1 to 5. We not only compare the model with and without domain SFT, but also study the influence of domain match between training and inference. *cross-domain test* labels testing a model after domain SFT with mismatched domain data. e.g., testing the model after audiobook domain SFT with podcast test data. On the contrary, *in-domain* test denotes the testing scenario with matched domain data. The Table 1 shows the PER of the model with domain SFT is superior to that without domain SFT, as well as the MOS score. For the zero-shot speaker similarity, both models with and without domain SFT share a similar SIM score. We also find that the domain consistency of training and inference further boosts the performance of the generated speech from both subjective and objective perspectives.

Table 2: Evaluating results of Speaker SFT

Model	PER(↓)	SIM(↑)	MOS(↑)
domain SFT	1.91	0.70	4.23 ± 0.06
speaker SFT	1.13	0.81	4.35 ± 0.08
domain SFT + speaker SFT	**0.89**	**0.82**	$\mathbf{4.46 \pm 0.07}$

Speaker SFT also trains extra LoRA parameters of Takin TTS model and can be stacked onto the model after Domain SFT. We conduct the experiment of speaker SFT based on *Speaker TTS Dataset* and all the evaluations are based on the test data of the four speakers in that dataset. In the experiment of speaker SFT, we study the influence of stacking domain SFT and speaker SFT, by comparing it with the models only with domain SFT and the ones with only speaker SFT.

As illustrated in Table 2, we use the same evaluations as that in domain SFT experiments. the speaker SFT obviously plays the most important part to improve the performance of a specific speaker on both objective and subjective perspectives. However, just speaker SFT alone does not achieve the best results. We suspect that is due to the model after domain SFT works like a better starting point for speaker SFT in the same domain.

5.1.3 RL Training on TTS

RL training can be employed as an extra post-training stage after either Domain SFT or Speaker SFT. Both experiments are conducted to verify the effectiveness of RL training, especially on expressiveness and BCR. To prepare training data for RL, we make a set of good / bad examples with both subjective ratings and objective metrics. As [46] shows repeated sampling is able to largely increase the pass coverage to queried problems, we get 5 samples by repeated sampling for each sentence. For objective ratings, we pick PER and UTMOS [47] as objective metrics to generate preferences considering both metrics. For subjective ratings, there are 50 human raters being

participated in to rate the best and the worst sample among 5 candidates by listening and comparing the overall speech quality. Therefore, a good / bad example pair is acquired for each sentence either by objective or subjective ratings. As a result, for the RL experiments, 50000-sentence RL pairs are created for RL training after Domain SFT and 500-sentence RL pairs are prepared for two male and two female speakers respectively after speaker SFT. The experiments are defined as follows.

- ***Domain-SFT-RL-OBJ*** denotes the RL experiments conducted based on the model after domain SFT by using objective-metric-rating data.
- ***Domain-Speaker-SFT-RL-OBJ*** denotes the RL experiments conducted based on the model after domain and speaker SFT by using objective-metric-rating data.
- ***Domain-SFT-RL-SUBJ*** denotes the RL experiments conducted based on the model after domain SFT by using human-rating data.
- ***Domain-Speaker-SFT-RL-SUBJ*** denotes the RL experiments conducted based on the model after speaker SFT by using human-rating data.

In our experiments, DPO is employed to do RL post training on *Domain-SFT-RL* and *Speaker-SFT-RL* respectively. The Table 3 shows the results comparing models with (w/) or without (w/o) RL training. To make the result comparable, the four speakers in *Speaker TTS Dataset* are picked to evaluate various metrics. The models named with *SUBJ* suffix are trained with human-rating pairs, the results of which indicate more stable speech generation in terms of *PER* and *BCR*. However there is just minor improvements on MOS which is more related to expressiveness. That might be due to RL data raters more sensitive to bad cases comparing to prosody changes.

Table 3: Objective and subject evaluating results of models with and without DPO.

Model	PER	BCR	MOS	SIM
Domain-SFT w/o RL	1.91	1.1%	4.23 ± 0.06	0.69
Domain-Speaker-SFT w/o RL	0.89	0.7%	4.46 ± 0.07	0.82
Domain-SFT-RL-SUBJ	1.79	0.9%	4.26 ± 0.07	0.71
Speaker-SFT-RL-SUBJ	0.89	0.4%	4.53 ± 0.09	0.81
Domain-SFT-RL-OBJ	1.89	1.1%	4.22 ± 0.09	0.7
Speaker-SFT-RL-OBJ	0.93	0.6%	4.55 ± 0.11	0.81

The data created by human raters are labour demanded especially for single speaker RL training. Thus, it is worthy to analyze the utility of pairs generated by using just objective metrics. According to the evaluation results from Table 3, applying RL training onto the weights in Domain SFT using data rated by objective metrics do not bring significant improvement. However, we see much larger improvement from the results of *Speaker-SFT-RL-OBJ*, though the results do not demonstrate the same level of improvement as in *Speaker-SFT-RL-SUBJ*. We further analyze those experimental result by conducting a consistency analysis over data rated by objective metrics, regarding the human rated ones as ground truth. We find there is 64% overlap in Speaker RL experiments, while there is only 55% overlap in Domain RL experiments, which is consistent with the experimental results in Table 3 for *Domain-SFT-RL-OBJ* and *Speaker-SFT-RL-OBJ*.

5.1.4 Emotion Control Based on Instructions

Data preparation Regarding the textual description, our professional data expert proposes three dimensions for annotating a voice recording: speaking emotion, speaking state, and speaking rhythm. Considering the difficulty and accuracy of annotation, the dimension of emotion is more distinctive compared to the other two dimensions, with the control of the remaining dimensions acting as supplementary control for the emotional dimension. We have established nine commonly recognized emotional directions(see in Table4) and then described them using various synonymous natural language. We have annotated approximately 100 hours of audio data, with each audio clip's corresponding textual annotation cross-validated by three different experienced data annotators. All these annotated data are utilized for supervised fine-tuning on a large language model.

Evaluation Metrics The performance of TakinTTS-Instruct compared with TakinTTS-base is shown in table4 and table5. To evaluate the accuracy of instruction controllability over speech synthesis, we randomly selected 50 different sentences with a fixed speaker prompt for each emotion, attempting to

test whether different instructions could achieve the goal of controlling the emotional style of the synthesized audio. These sample instructions were derived from user inputs, such as: **"The speaker says it with a smile, in a tone that is somewhat familiar and at a fast speed, expressing a pleasant emotion."**

Table 4: Comparison of emotion control accuracy (↑) between Takin-TTS and Takin-TTS-Instruct.

Emotion	TakinTTS	TakinTTS-Instruct	Emotion	TakinTTS	TakinTTS-Instruct
Admire	0.34±0.05	0.45±0.03	Disgust	0.53±0.06	0.59±0.02
Alert	0.27±0.06	0.35±0.04	Joy	1.00±0.00	1.00±0.00
Anger	0.67±0.03	0.73±0.08	Sad	0.71±0.11	0.85±0.03
Fear	0.55±0.04	0.79±0.02	Surprise	0.25±0.01	0.43±0.02

The research results in Table 4 indicate that TakinTTS-Instruct demonstrates strong controllability under various command inputs. Compared to the model before instruction finetuning, there is a significant improvement in the emotion similarity between the prompt and generated speech.

Table 5: Indicator of Instruction control accuracy (↑) between TakinTTS and TakinTTS-Instruct.

System	PER(↓)	MOS(↑)	SIM(↑)
TakinTTS-Instruct	1.9	4.48±0.13	0.78
TakinTTS	1.82	4.46±0.07	0.79

Furthermore, the objective metrics in Table5 displays that the quality of the generated speechs from TakinTTS-Instruct system are no less than those of the benchmark Takin TTS, and even slightly outperformed.

5.1.5 Efficient Inference and Serving

To generate speech with superior quality, we use auto-regressive LLMs and diffusion models in Takin, which are difficult and expensive to deploy. So we use various techniques and tricks to build an inference service with low latency and high concurrency. Our efforts on TTS task are as follows: Since most of the computation is spent on LLM model inference, we deploy a separate service for LLM to maximize GPU utilization and throughput for token prediction. Flash attention[48, 49] and paged attention[50] techniques are used in the prefill and the decode phases respectively, to reduce the consumption of memory and computation. Mixed precision and quantization techniques such as GPTQ[51] and AWQ[52] are also used to achieve further speedup. Besides, we adopt a suite of kernel-level optimizations, which leverage hardware-specific features and software techniques to accelerate critical computation kernels. As described above, CSFT strategy is used to improve the stability of synthesis. But it is not practical to deploy separate inference services for different domains and speakers. So we support multiple LoRAs in the same service, as well as batch inference for different LoRAs. Streaming inference is applied to scenarios such as real-time interaction, and the first packet delay is less than 300 ms.

5.2 Takin VC Experiments

5.2.1 Takin VC Datasets

Training Dataset used in Takin VC training heavily overlaps with the data in the TTS dataset, including approximately 500,000 hours of web-scraped and internal data.

Test Dataset We random select 100 out-of-set speaker speech data from the Internet. In addition, these speakers include different attributes such as gender, age, language, and emotion. Each speaker has about 1 to 3 sentences for different attributes.

5.2.2 Takin VC Performence

As shown in Table 6, our proposed Takin VC scheme surpasses the baseline solution in terms of both sound quality and speaker similarity. Our experiments were conducted under conditions of large datasets to ensure the scheme's effectiveness on a large scale.

Table 6: Comparison of Takin-VC and baselines.

System	PMOS	SMOS	UTMOS	SIM
DiffVC	3.34 ± 0.07	3.45 ± 0.07	3.48	0.61
ValleVC $\diamond$	3.48 ± 0.05	3.53 ± 0.08	3.59	0.67
NS2VC	3.31 ± 0.06	3.52 ± 0.07	3.45	0.54
DDDMVC	3.56 ± 0.07	3.61 ± 0.07	3.67	0.69
TakinVC	$\mathbf{4.02 \pm 0.04}$	$\mathbf{4.07 \pm 0.05}$	**4.16**	**0.80**

$\diamond$ stands for utilizing VC models derived from open-source repositories.

5.3 Takin Morphing Experiments

Experimental Datasets We trained Takin Morphing on a substantial corpus consisting of 20,000 hours of multilingual speech recordings in English and Chinese, consisting of in-house dataset alongside filtered portions of the WenetSpeech [53] and LibriLight [54]. To assess the performance of the proposed approach, we perform zero-shot speech synthesis and prosody transfer evaluations using in-house test sets which will be detailed below.

Evaluation Metrics To conduct an in-depth analysis of the proposed Takin Morphing approach, various objective and subjective metrics are employed. To elaborate, PER and SIM are used as objective measures as well, while quality mean option score (QMOS) is employed to assess quality, clarity, naturalness, and high-frequency details, and similarity mean option score (SMOS) is used to measure speaker similarity with respect to timbre reconstruction and prosodic patterns for subjective evaluation.

5.3.1 Zero-shot Speech Synthesis

To examine the zero-shot speech synthesis performance of the proposed Takin Morphing, we first designed two distinct test sets, referred to as the objective and the subjective test sets. The objective test set includes 2,000 samples each from in-house English (EN) and Mandarin (ZH) speech corpora, while the latter comprises 200 highly expressive in-house samples in both EN and ZH as well. Notably, each sample in the subjective test set includes a reference utterance and a target utterance spoken by the same speaker. During inference, the Takin Morphing System generates speech for the target text using the reference speech as an audio prompt. The results are presented in Table 7.

Table 7: Zero-shot speech synthesis results of Takin Morphing against real human speech.

Models	Language	PER	SIM	QMOS	SMOS
GT	EN	2.52%	0.834	4.41 ± 0.08	4.28 ± 0.12
Takin Morphing	EN	3.14%	0.846	4.09 ± 0.07	4.04 ± 0.06
GT	CN	2.16%	0.879	4.43 ± 0.11	4.32 ± 0.09
Takin Morphing	CN	3.05%	0.884	4.13 ± 0.09	4.09 ± 0.08

As shown in the table, on both Chinese and English test sets, our proposed Takin Morphing system achieved a performance level comparable to that of humans in terms of speech naturalness and speaker similarity. Notably, it slightly underperformed in the PER, QMOS, and SMOS metrics while achieving a higher score in the SIM metric. This outcome may be attributed to the fact that, even when both the real and reference speech originate from the same speaker, variations in speaking style, background environment, and other factors may still exist. In this context, Takin Morphing, when generating the target speech, accurately captures the fine-grained characteristics of the reference speech through more sophisticated and advanced timbre modeling, thereby enabling a more consistent and precise reproduction of the reference speech.

5.3.2 Prosody Transfer

To validate the prosody transfer performance of Takin Morphing, we transfer the styles from our internal dataset to audio samples from our main platform. Specifically, we randomly select 20 speakers from the main platform and choose 50 sentences for each of them. Subsequently, for each sentence of the selected speakers, we randomly choose an emotional speech clip from the internal emotional dataset and use it as the prosodic reference.

Table 8: Prosody transfer performance of Takin Morphing against real human speech.

Models	Language	PER	SIM	QMOS	SMOS
GT	EN	4.96%	0.823	4.19 ± 0.12	4.21 ± 0.09
Takin Morphing	EN	5.32%	0.835	3.94 ± 0.09	3.90 ± 0.07
GT	CN	2.99%	0.853	4.21 ± 0.09	4.24 ± 0.11
Takin Morphing	CN	3.05%	0.875	3.99 ± 0.06	3.92 ± 0.08

Table 8 presents all results for English (EN) and Chinese (CN), respectively. In terms of the objective evaluation, we can observe that Tarkin Morphing consistently achieved human-level performance with similar content recognition accuracy and better SIM score, highlighting the effectiveness of systematical design of our proposed approach. In subjective tests, Takin Morphing demonstrated a performance level in both English and Chinese that closely matches real human speech, with QMOS and SMOS scores both exceeding 3.9, underscoring the effectiveness of the proposed method in prosody interpolation.

6 Applications

6.1 Audiobook Generation

Takin TTS shows a large superiority comparing to conventional neural speech synthesis methods [55, 56, 57, 58, 59], which revolutionizes the field of AI audiobook generation. Two distinct approaches to creating immersive audio experiences using Takin TTS are explored. In the first approach, purely AI-generated audio content is produced, where different AI-powered voices act as various characters, bringing the story to life with diverse and nuanced performances. This approach allows for a consistent and scalable production process, potentially reducing costs and time associated with traditional audiobook recording. The another approach combines AI and human voices, with Takin TTS handling narration while human voice actors take on the dialogue parts. This hybrid approach leverages the efficiency and consistency of AI-generated speech for descriptive passages while preserving the emotional depth and authenticity that human actors bring to character interactions. The AI-generated audiobook samples can be listened in our demo page.

6.2 Voice Clone

In recent years, zero-shot timbre cloning technology has achieved significant advancements in voice cloning and speech synthesis, and is widely used in various fields. In voice assistants and customer service robots, it provides a more natural interaction experience; in the fields of film and entertainment content production, it is used for dubbing and creating voices for animated characters; in voice memos and recordings, it clones the voices of celebrities for future preservation. In music production, it can mimic the timbre of specific instruments; in education and training, it creates learning materials with standard pronunciations; in medical and rehabilitation, it helps patients who have lost the ability to speak regain their voices. Additionally, historical reconstructions and museum exhibits benefit from this technology. Using Takin VC's technology, the model requires only a few seconds to tens of seconds of audio samples to generate high-quality simulated voices, greatly reducing the technical threshold and making the aforementioned applications possible.

6.3 Talking head

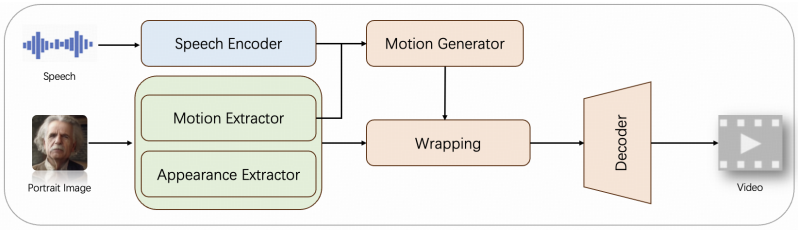

Figure 8: Talking Head Framework

By integrating LLM-based TTS systems with Portrait Animation technology, we can effortlessly create an interactive talking head system. The TTS system converts text to speech, while the Portrait Animation system generates expressive and temporally coherent animations synchronized with the speech, resulting in a lifelike animated character that communicates naturally and engagingly. The inference pipeline is illustrated in Figure 8.

7 Authors (alphabetical order of family name)

- Sijing Chen
- Yuan Feng
- Laipeng He
- Tianwei He
- Wendi He
- Yanni Hu
- Bin Lin
- Yiting Lin
- Yu Pan
- Pengfei Tan
- Chengwei Tian
- Chen Wang
- Zhicheng Wang
- Ruoye Xie
- Jixun Yao
- Quanlei Yan
- Yuguang Yang
- Jianhao Ye
- Jingjing Yin
- Yanzhen Yu
- Huimin Zhang
- Xiang Zhang
- Guangcheng Zhao
- Hongbin Zhou
- Pengpeng Zou

References

[1] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[2] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh,

Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[3] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-audio technical report, 2024.

[4] Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

[5] Cristina Gârbacea, Aäron van den Oord, Yazhe Li, Felicia SC Lim, Alejandro Luebs, Oriol Vinyals, and Thomas C Walters. Low bit-rate speech coding with vq-vae and a wavenet decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 735–739. IEEE, 2019.

[6] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[7] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[8] Yu Pan, Lei Ma, and Jianjun Zhao. Promptcodec: High-fidelity neural speech codec using disentangled representation learning based adaptive feature-aware prompt encoders, 2024.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[13] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023.

[14] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[15] Mateusz Łajszczak, Guillermo Cámbara, Yang Li, Fatih Beyhan, Arent van Korlaar, Fan Yang, Arnaud Joly, Álvaro Martín-Cortinas, Ammar Abbas, Adam Michalski, et al. Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data. *arXiv preprint arXiv:2402.08093*, 2024.

[16] James Betker. Better speech synthesis through scaling. *arXiv preprint arXiv:2305.07243*, 2023.

[17] Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, et al. Naturalspeech: End-to-end text-to-speech synthesis with human-level quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[18] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.

[19] Xintao Zhao, Feng Liu, Changhe Song, Zhiyong Wu, Shiyin Kang, Deyi Tuo, and Helen Meng. Disentangling content and fine-grained prosody information via hybrid asr bottleneck features for voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7022–7026. IEEE, 2022.

[20] Trung Dang, Dung Tran, Peter Chin, and Kazuhito Koishida. Training robust zero-shot voice conversion models with self-supervised features. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6557–6561. IEEE, 2022.

[21] Zhichao Wang, Yuanzhe Chen, Lei Xie, Qiao Tian, and Yuping Wang. Lm-vc: Zero-shot voice conversion via speech generation based on language models. *IEEE Signal Processing Letters*, 2023.

[22] Jixun Yao, Yuguang Yang, Yi Lei, Ziqian Ning, Yanni Hu, Yu Pan, Jingjing Yin, Hongbin Zhou, Heng Lu, and Lei Xie. Promptvc: Flexible stylistic voice conversion in latent space driven by natural language prompts. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10571–10575. IEEE, 2024.

[23] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[24] Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W Mahoney, and Kurt Keutzer. Squeezeformer: An efficient transformer for automatic speech recognition. *Advances in Neural Information Processing Systems*, 35:9361–9373, 2022.

[25] Yuguang Yang, Yu Pan, Jingjing Yin, and Heng Lu. Lmec: Learnable multiplicative absolute position embedding based conformer for speech recognition. *arXiv preprint arXiv:2212.02099*, 2022.

[26] Yi Ren, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Revisiting over-smoothness in text to speech. *arXiv preprint arXiv:2202.13066*, 2022.

[27] Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

[28] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[29] Direct Preference Optimization: Your Language Model is Secretly a Reward Model — proceedings.neurips.cc. `https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html`. [Accessed 18-09-2024].

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[31] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[32] Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*, 2024.

[33] SpeechAlign: Aligning Speech Generation to Human Preferences — arxiv.org. `https://arxiv.org/abs/2404.05600`. [Accessed 18-09-2024].

[34] Enhancing Zero-shot Text-to-Speech Synthesis with Human Feedback — arxiv.org. `https://arxiv.org/abs/2406.00654`. [Accessed 18-09-2024].

[35] Robust Zero-Shot Text-to-Speech Synthesis with Reverse Inference Optimization — arxiv.org. `https://arxiv.org/abs/2407.02243`. [Accessed 18-09-2024].

[36] Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, and Qian Chen. Cam++: A fast and efficient network for speaker verification using context-aware masking. *arXiv preprint arXiv:2303.00332*, 2023.

[37] Yu Pan, Yanni Hu, Yuguang Yang, Wen Fei, Jixun Yao, Heng Lu, Lei Ma, and Jianjun Zhao. Gemo-clap: Gender-attribute-enhanced contrastive language-audio pretraining for accurate speech emotion recognition. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10021–10025, 2024.

[38] Yu Pan, Yuguang Yang, Heng Lu, Lei Ma, and Jianjun Zhao. Gmp-tl: Gender-augmented multi-scale pseudo-label enhanced transfer learning for speech emotion recognition, 2024.

[39] Yu Pan, Yuguang Yang, Yuheng Huang, Jixun Yao, Jingjing Yin, Yanni Hu, Heng Lu, Lei Ma, and Jianjun Zhao. Msac: Multiple speech attribute control method for reliable speech emotion recognition, 2024.

[40] Guofeng Yi, Yuguang Yang, Yu Pan, Yuhang Cao, Jixun Yao, Xiang Lv, Cunhang Fan, Zhao Lv, Jianhua Tao, Shan Liang, et al. Exploring the power of cross-contextual large language model in mimic emotion prediction. In *Proceedings of the 4th on Multimodal Sentiment Analysis Challenge and Workshop: Mimicked Emotions, Humour and Personalisation*, pages 19–26, 2023.

[41] Jianlin Su. Simbert: Integrating retrieval and generation into bert. Technical report, ZhuiyiTechnology, 2020.

[42] Yuguang Yang, Yu Pan, Jingjing Yin, Jiangyu Han, Lei Ma, and Heng Lu. Hybridformer: Improving squeezeformer with hybrid attention and nsr mechanism. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[43] Ziyue Jiang, Yi Ren, Zhenhui Ye, Jinglin Liu, Chen Zhang, Qian Yang, Shengpeng Ji, Rongjie Huang, Chunfeng Wang, Xiang Yin, et al. Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias. *arXiv preprint arXiv:2306.03509*, 2023.

[44] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[45] Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, and Qian Chen. Cam++: A fast and efficient network for speaker verification using context-aware masking. In *INTERSPEECH*, 03 2023.

[46] Large Language Monkeys: Scaling Inference Compute with Repeated Sampling — arxiv.org. `https://arxiv.org/abs/2407.21787`. [Accessed 18-09-2024].

[47] Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*, 2022.

[48] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

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

[50] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[51] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[52] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: activationaware weight quantization for llm compression and acceleration. corr, abs/2306.00978, 2023. doi: 10.48550. *arXiv preprint ARXIV.2306.00978*, 2023.

[53] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. Wenetspeech: A 10000+ hours multi-domain mandarin

corpus for speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186. IEEE, 2022.

[54] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE, 2020.

[55] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerrv Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *ICASSP*, pages 4779–4783, 2018.

[56] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International conference on machine learning*, pages 195–204. PMLR, 2017.

[57] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[58] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

[59] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *ICML*, pages 5530–5540, 2021.