Towards Natural and Efficient Speech Synthesis: Perspectives on Modeling, Alignment, and Representation

Yuancheng Wang
The Chinese University of Hong Kong, Shenzhen

About Me

- Education:
  - I am a second-year Ph.D. student in the School of Data Science at The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), supervised by Professor Zhizheng Wu.
  - Before that, I received my B.S. degree at CUHK-Shenzhen in 2023.
- Research:
  - Speech synthesis: NaturalSpeech 3, MaskGCT, Metis, INTP…
  - Speech Representation: FACodec, TaDiCodec, DualCodec…
  - Other Work in Audio, Speech, and Multimodality: AUDIT, SD-Eval…

Contents

Contents

- **Unifying Speech Synthesis through Masked Language Modeling**
  - Introduction to our paper: ***Metis: A Foundation Speech Generation Model with Masked Generative Pre-training***
    - Arxiv: https://arxiv.org/abs/2502.03128
    - Demo page: https://metis-demo.github.io/
    - Huggingface: https://huggingface.co/amphion/Metis

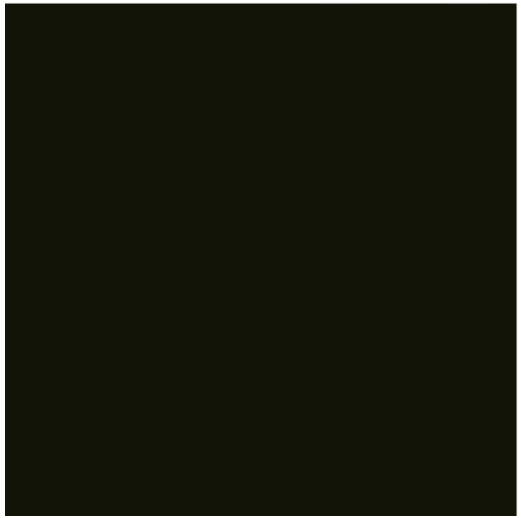
Wang, Y., Zheng, J., Zhang, J., Zhang, X., Liao, H., & Wu, Z. Metis: A Foundation Speech Generation Model with Masked Generative Pre-training. ArXiv preprint arXiv:2502.03128.

Background: MGM for Speech Generation

- What are masked generative models?
  - Also known as masked diffusion models or discrete diffusion models (a special case)...
  - BERT-style language models:
    - Text: Gemini Diffusion, LLaDA
    - Vision: MaskGIT, MAGVIT, LLaDA-V, MMaDA
    - Audio: SoundStorm, NaturalSpeech 3, MaskGCT, AnyEnhance

Background: MGM for Speech Generation

Autoregressive Decoding



MaskGIT's Parallel Decoding

From MaskGIT.

Chang, H., Zhang, H., Jiang, L., Liu, C., & Freeman, W. T.. Maskgit: Masked generative image transformer. CVPR 2022.

Background: MGM for Speech Generation

- Compare with autoregressive models:

Autoregressive Model:

Next token prediction

Causal attention

Sequential decoding

Left-to-right generation

Masked Generative Model:

Masked token prediction

Bi-direction attention

Parallel decoding

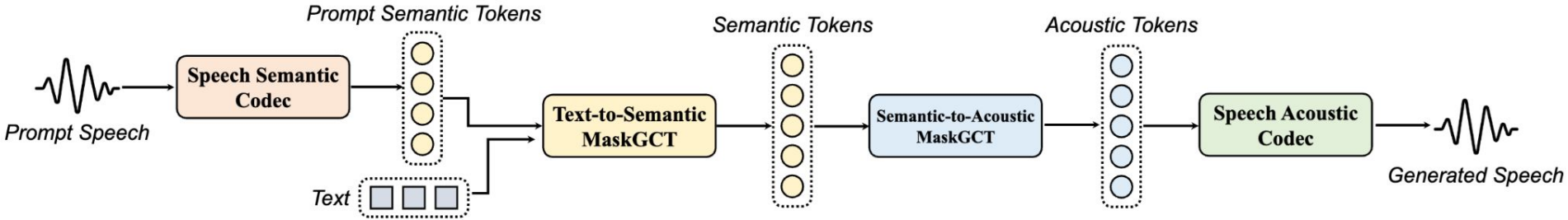
Random generation

Background: MGM for Speech Generation

- Why use MGM for speech generation:
  - (Potentially) faster inference speed
  - Can control duration
  - Exploring new generative paradigms

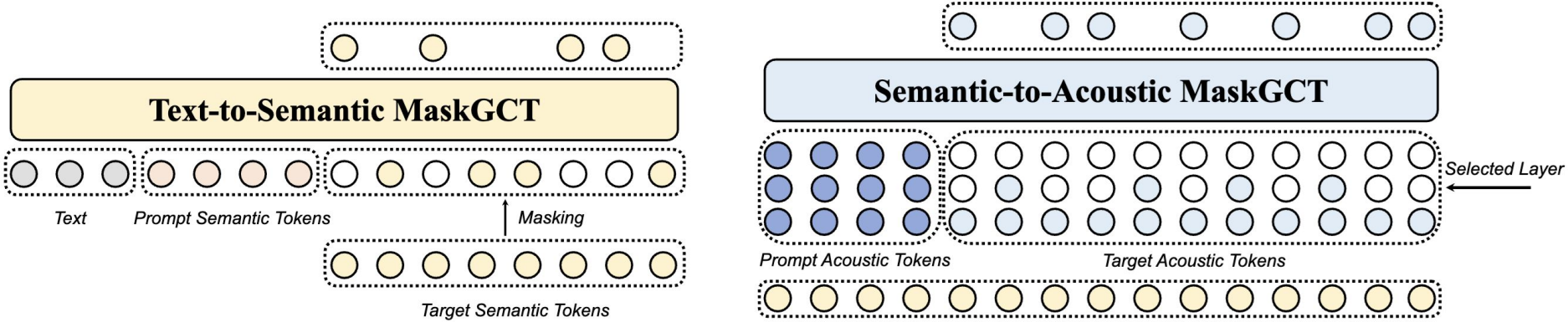
Background: MGM for Speech Generation

- MaskGCT: MGM for Zero-Shot Text-to-Speech



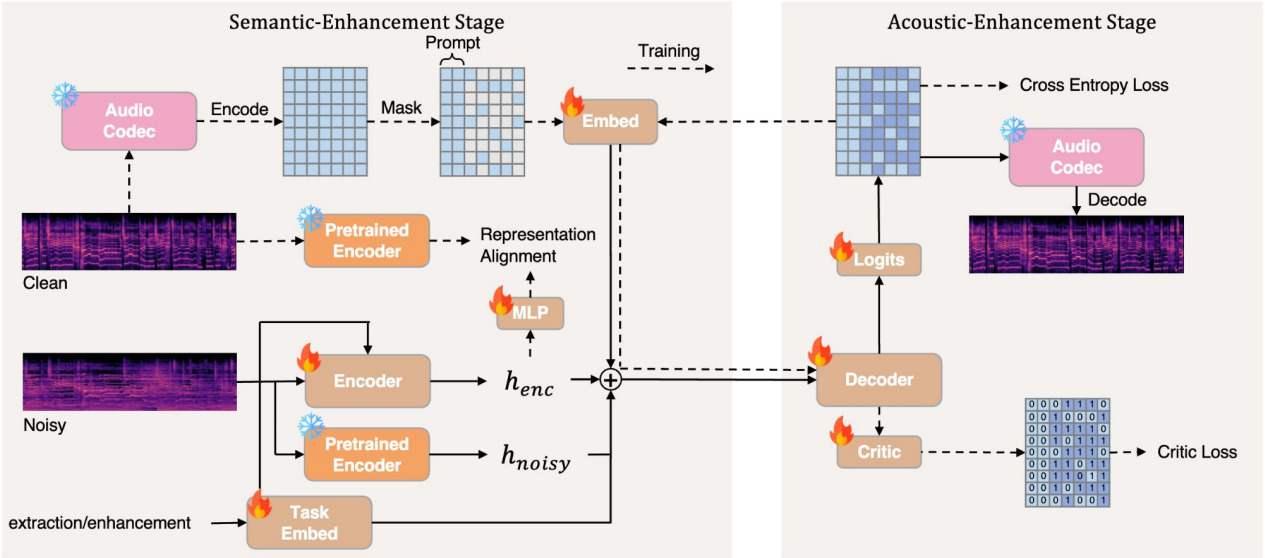
Background: MGM for Speech Generation

- MaskGCT: MGM for Zero-Shot Text-to-Speech



Background: MGM for Speech Generation

- AnyEnhance: MGM for SE and TSE



Zhang, J., Yang, J., Fang, Z., Wang, Y., Zhang, Z., Wang, Z., ... & Wu, Z. AnyEnhance: A Unified Generative Model with Prompt-Guidance and Self-Critic for Voice Enhancement. TASLP.

Motivation: Unified Speech Generation

- Our goal: design a unified speech generation framework
  - Most speech generation models are expert models that require extensive task-specific designs
  - Some works attempt to use autoregressive language models for multiple speech generation tasks: UniAudio, SpeechX…

Key question: How to design a **unified speech generation framework** that leverages large-scale unlabeled speech data for **pre-training** and **efficiently adapts** to diverse speech generation tasks through **fine-tuning**.

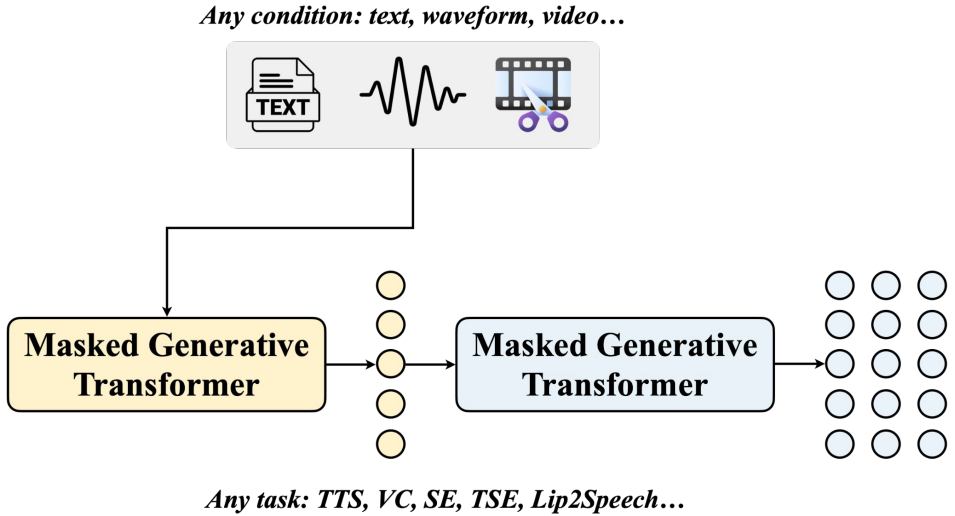
Motivation: Unified Speech Generation

- Revisiting the speech generation process:
  - **Task-Specific Process**: generating **intermediate representations** from **task-specific conditions**
    - the intermediate representations are typically discrete units quantized from self-supervised speech features (SSL tokens; or semantic tokens)
  - **Task-Independent Process**: synthesizing **acoustic representations** from these **intermediates**
    - it can be achieved effectively using fully self-supervised diffusion models
    - acoustic representations are easily used to reconstruct high-quality waveforms

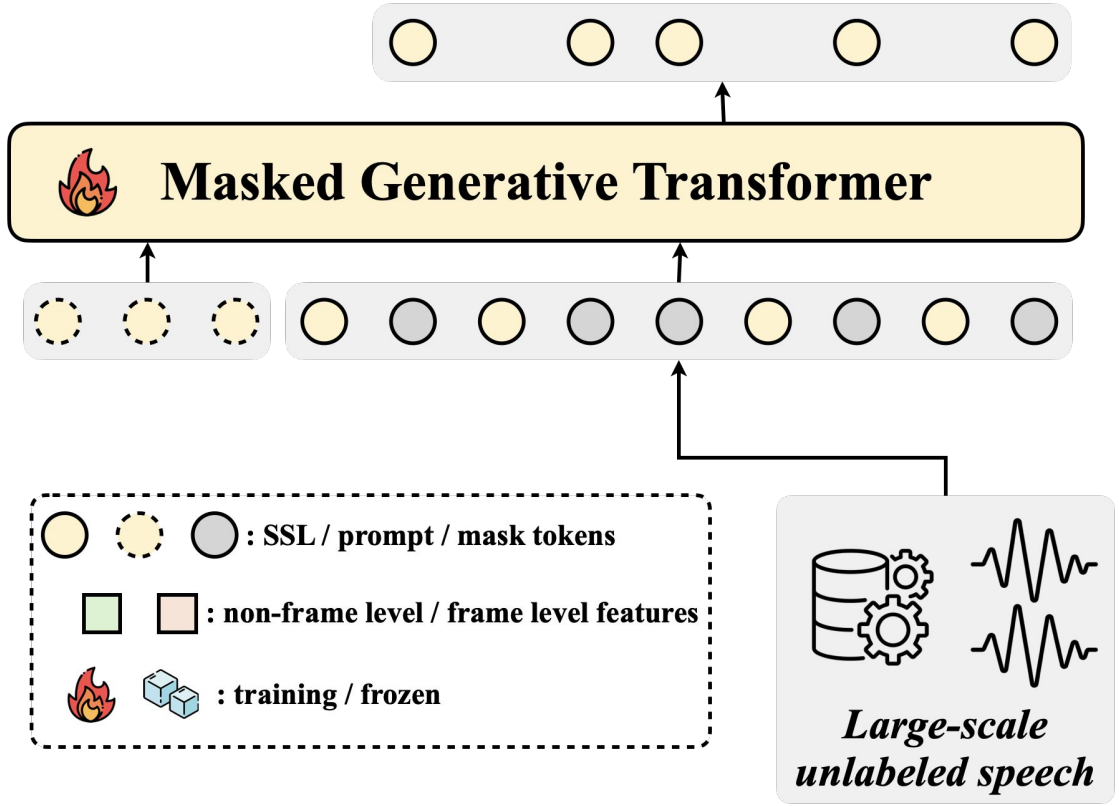
Our objective: simplified to designing a **generative pre-training mechanism** for the first stage, enabling the pre-trained model to **effectively generalize** across various speech generation tasks.

What We Do

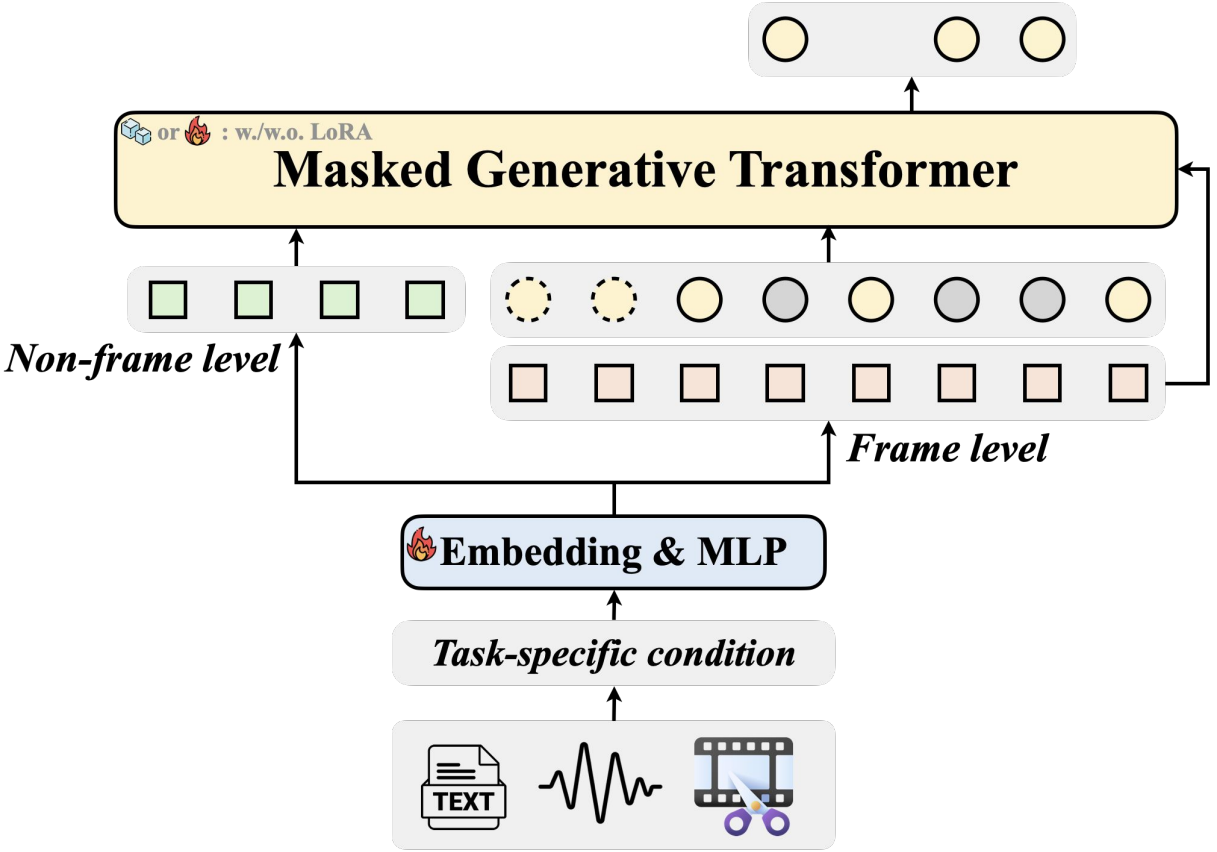
- Adopt masked generative models for unconditional pre-training
  - Masked prediction pre-training is typically used for representation learning
  - Using a straightforward masked token prediction without any condition
  - The pre-trained model can be **fine-tuned with limited data and parameters** (with LoRA) by incorporating task-specific conditions



Method: Pre-training



Method: Fine-tuning



Result: Evaluation on Zero-shot TTS and VC

Model	Training Data	WER(↓)	SIM(↑)	DNSMOS(↑)
	SeedTTS test-en			
Ground Truth	-	2.14	0.73	3.53
VALL-E (Wang et al., 2023a)	45K EN	6.13	0.43	3.39
VoiceCraft (Peng et al., 2024)	9K EN	7.55	0.47	3.37
CosyVoice (Du et al., 2024)	170K Multi.	4.08	0.64	3.64
XTTS-v2 (Casanova et al., 2024)	27K Multi.	3.25	0.46	3.45
MaskGCT (Wang et al., 2024c)	100K Multi.	2.47	0.72	3.51
Metis-TTS LoRA 32	1K Multi.	4.78	0.72	3.47
Metis-TTS LoRA 32	10K Multi.	**4.55**	0.72	3.45
Metis-TTS LoRA 32	0.58K[1] EN.	4.63	0.70	3.51
Metis-TTS fine-tune	0.58K[1] EN.	**3.04**	0.68	3.47
Metis-TTS fine-tune	1K Multi.	3.86	0.71	3.46
Metis-TTS fine-tune	10K Multi.	**2.28**	0.72	3.47
Metis-TTS w.o. pre-train	10K Multi.	4.91	0.69	3.42
	SeedTTS test-zh			
Ground Truth	-	1.25	0.75	3.51
CosyVoice (Du et al., 2024)	170K Multi.	4.09	0.75	3.71
XTTS-v2 (Casanova et al., 2024)	27K Multi.	2.88	0.63	3.44
MaskGCT (Wang et al., 2024c)	100K Multi.	2.18	0.77	3.58
Metis-TTS LoRA 32	1K Multi.	5.21	0.77	3.57
Metis-TTS LoRA 32	10K Multi.	**4.48**	0.77	3.55
Metis-TTS fine-tune	1K Multi.	4.23	0.77	3.54
Metis-TTS fine-tune	10K Multi.	**2.30**	0.77	3.55
Metis-TTS w.o. pre-train	10K Multi.	4.98	0.73	3.51

[1] This version of the model is trained on LibriTTS (Zen et al., 2019a).

Model	Training Data	WER(↓)	SIM(↑)	DNSMOS(↑)	NISQA(↑)
		VCTK			
HierSpeech++ (Lee et al., 2023)	2.8K	4.87	0.38	3.40	3.79
LM-VC (Wang et al., 2023b)	1.4K	8.35	0.29	3.46	3.93
UniAudio (Yang et al., 2023)	60K	9.00	0.25	3.47	4.28
Vevo (Zhang et al., 2024)	60K	**3.48**	0.38	3.47	4.30
Metis-VC LoRA 16, cfg = 0.0	0.4K	<u>4.49</u>	<u>0.50</u>	<u>3.48</u>	<u>4.46</u>
Metis-VC LoRA 16, cfg = 2.0	0.4K	7.90	**0.55**	3.46	4.42
Metis-VC fine-tune, cfg = 0.0	0.4K	6.65	0.48	**3.49**	**4.47**

[1] For LoRA 16, we train on one A100 GPU for 10K steps. For fine-tuning, we train on one A100 GPU for 5K steps.

[2] The best and the second best result is shown in **bold** and by <u>underlined</u>.

Result: Evaluation on TSE and SE

Model	WER(↓)	SIG(↑)	BAK(↑)	OVRL(↑)	NISQA(↑)	SIM(↑)
			LibriMix test			
Ground Truth	4.27	3.62	4.03	3.32	4.11	0.76
UniAudio (Yang et al., 2023)	20.08	3.64	**4.15**	3.33	4.32	0.66
VoiceFilter (Wang et al., 2018)	20.10	3.27	3.77	2.91	2.97	0.68
WeSep (Wang et al., 2024a)	**6.19**	3.56	3.93	3.23	4.04	0.73
TSELM (Tang et al., 2024)	9.20	3.55	4.08	3.29	4.03	0.27
Metis-TSE LoRA 4	13.57	**3.66**	4.02	3.34	4.40	**0.75**
Metis-TSE LoRA 16	12.52	**3.66**	4.02	**3.35**	**4.41**	**0.75**
Metis-TSE LoRA 32	9.65	**3.66**	4.02	**3.35**	4.38	**0.75**
Metis-TSE fine-tune	6.31	3.65	4.02	3.34	4.36	0.74
			EmiliaMix test			
Ground Truth	0.00	3.57	4.01	3.27	3.83	0.86
UniAudio (Yang et al., 2023)	32.51	3.55	4.03	3.20	4.01	0.55
VoiceFilter (Wang et al., 2018)	24.10	3.21	3.64	2.78	2.15	0.71
WeSep (Wang et al., 2024a)	**5.58**	3.50	3.85	3.12	3.86	**0.81**
TSELM (Tang et al., 2024)	52.12	3.48	**4.05**	3.20	3.87	0.26
Metis-TSE LoRA 4	11.55	3.59	3.84	**3.21**	4.04	0.77
Metis-TSE LoRA 16	10.48	**3.60**	3.85	**3.21**	**4.06**	0.77
Metis-TSE LoRA 32	8.72	**3.60**	3.86	**3.21**	4.05	0.78
Metis-TSE fine-tune	6.87	**3.60**	3.85	**3.21**	**4.06**	0.78

[1] The best and the second best result is shown in **bold** and by underlined.

[2] For LibriMix, we use the original text for computing WER. For EmiliaMix, we use the ASR-transcribed text of the ground truth speech for WER calculation.

Model	SIG(↑)	BAK(↑)	OVRL(↑)	NISQA(↑)	SIM(↑)
		DNS2020 with reverb			
TF-GridNet (Wang et al., 2023c)	3.11	3.23	2.51	2.61	0.69
VoiceFixer (Liu et al., 2022)	3.43	4.02	3.13	3.82	0.91
SELM (Wang et al., 2024d)	3.16	3.58	2.70	-	-
MaskSR (Li et al., 2024)	3.53	4.07	3.25	-	-
Metis-SE LoRA 4	3.67	4.13	3.43	4.54	0.93
Metis-SE LoRA 16	3.67	4.13	3.43	**4.57**	**0.94**
Metis-SE LoRA 32	**3.68**	**4.14**	3.43	4.48	**0.94**
Metis-SE fine-tune	**3.68**	**4.14**	**3.44**	4.56	**0.94**
		DNS2020 no reverb			
TF-GridNet (Wang et al., 2023c)	3.54	4.05	3.27	4.35	0.68
VoiceFixer (Liu et al., 2022)	3.50	4.11	3.25	4.27	0.96
SELM (Wang et al., 2024d)	3.51	4.10	3.26	-	-
MaskSR (Li et al., 2024)	3.60	4.15	3.37	-	-
Metis-SE LoRA 4	3.65	4.15	3.43	**4.82**	**0.97**
Metis-SE LoRA 16	3.65	4.16	3.43	4.81	**0.97**
Metis-SE LoRA 32	**3.66**	**4.17**	**3.44**	4.77	**0.97**
Metis-SE fine-tune	3.64	**4.17**	3.43	4.76	**0.97**
		DNS2020 Real Recording			
VoiceFixer (Liu et al., 2021)	3.31	3.93	3.00	3.66	-
SELM (Wang et al., 2024d)	3.59	3.44	3.12	-	-
MaskSR (Li et al., 2024)	3.43	4.03	3.14	-	-
Metis-SE LoRA 4	**3.60**	4.02	3.29	3.94	-
Metis-SE LoRA 16	**3.60**	**4.04**	**3.30**	**3.97**	-
Metis-SE LoRA 32	3.59	3.99	3.26	3.92	-
Metis-SE fine-tune	3.59	4.01	3.27	3.95	-

[1] The best and the second best result is shown in **bold** and by underlined.

Result: Evaluation on Lip2Speech

Model	WER(↓)	DNSMOS(↑)	NISQA(↑)	SIM(↑)
		LRS2		
Lip2Speech-Unit (Kim et al., 2023)	33.64	3.01	2.70	29.34
Metis-L2S fine-tune	**32.28**	**3.23**	**3.71**	**59.73**
		LRS3		
Lip2Speech-Unit (Kim et al., 2023)	38.34	2.28	1.92	32.05
Metis-L2S fine-tune	**31.03**	**3.09**	**3.75**	**56.74**

Result: Omni-Model

Task	Dataset	Model	Performance	
			Metrics	Results
Zero-shot TTS	***SeedTTS-en***	MaskGCT (Wang et al., 2024c)	WER(↓) | SIM(↑) | DNSMOS(↑)	2.47 | 0.72 | 3.52
		Metis-TTS		**2.41** | **0.72** | **3.57**
		Metis-Omni		4.78 | 0.71 | **3.57**
	SeedTTS-zh	MaskGCT (Wang et al., 2024c)	WER(↓) | SIM(↑) | DNSMOS(↑)	2.18 | 0.77 | 3.58
		Metis-TTS		**2.30** | **0.77** | 3.55
		Metis-Omni		4.39 | **0.77** | **3.60**
VC	***VCTK***	LM-VC (Wang et al., 2023b)	WER(↓) | SIM(↑) |DNSMOS(↑)	8.35 | 0.29 | 3.46
		Metis-VC		6.65 | **0.48** | 3.49
		Metis-Omni		**3.52** | 0.34 | **3.51**
SE	***DNS2020 with reverb***	TF-Grident (Wang et al., 2023c)	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.11 | 3.23 | 2.51 | 2.61
		Metis-SE		**3.68** | **4.14** | **3.44** | **4.56**
		Metis-Omni		**3.68** | 4.13 | **3.44** | 4.53
	DNS2020 no reverb	TF-Grident (Wang et al., 2023c)	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.54 | 4.05 | 3.27 | 4.35
		Metis-SE		3.64 | **4.17** | 3.43 | 4.77
		Metis-Omni		**3.65** | **4.17** | **3.44** | **4.79**
	Real Recording	VoiceFixer (Liu et al., 2022)	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.31 | 3.93 | 3.00 | 3.66
		Metis-SE		3.59 | **4.01** | **3.27** | 3.95
		Metis-Omni		**3.60** | 4.00 | **3.27** | **3.96**
TSE	***LibriMix test***	WeSep (Wang et al., 2024a)	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.56 | 3.39 | 3.23 | 4.04
		Metis-TSE		3.65 | 4.02 | 3.34 | 4.36
		Metis-Omni		**3.66** | **4.03** | **3.36** | **4.40**
	EmiliaMix test	WeSep (Wang et al., 2024a)	SIG(↑) | BAK(↑) | OVRL(↑) | NISQA(↑)	3.50 | 3.85 | 3.12 | 3.86
		Metis-TSE		3.60 | **3.85** | 3.21 | 4.06
		Metis-Omni		**3.62** | 3.82 | **3.24** | **4.11**

Samples

- Demo page: https://metis-demo.github.io/

Conclusion and Discussion

- We propose Metis, a foundation model for unified speech generation that:
  - leverages large-scale unlabeled speech data for pre-training
  - can be effectively adapted to diverse speech generation tasks through fine-tuning
- Can we use MGM to unify speech generation and understanding?
  - There is some try in vision: LLaDA-V, MMaDA
  - Bi-direction attention can learn dependencies (text-speech) between different modalities simultaneously
- Support streaming inference:
  - block-wise MGM (Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. ICLR 2025 Oral)

Contents

- **Enhancing Speech Synthesis with Performance Alignment**
  - Introduction to our ACL 2025 paper: ***Advancing Zero-shot Text-to-Speech Intelligibility across Diverse Domains via Preference Alignment***
    - Arxiv: https://arxiv.org/abs/2505.04113
    - Demo page: https://intalign.github.io/
    - Huggingface: coming soon

Zhang X*, **Wang Y***, Wang C, Li Z, Chen Z, Wu Z. Advancing Zero-shot Text-to-Speech Intelligibility across Diverse Domains via Preference Alignment. ACL 2025.

Motivation

- Modern (academic) zero-shot TTS systems still lack robustness during real-world applications
  - Struggling to meet even the most fundamental requirement of speech synthesis – ***intelligibility*** in several scenarios: articulatory, code-switching, cross-lingual…
- Why? Primarily due to the problem of **out-of-distribution**
  - For example, in cross-lingual cases, there is a significant mismatch between **monolingual pre-training** and **cross-lingual inference**
  - Naive idea: collecting high-quality data for challenging cases (but difficult).
- Use performance alignment to mitigate OOD
  - PA's customized post-training on human-expected distributions helps mitigate OOD
  - PA needs only paired samples with **relative preferences** – **even synthetic data can lead to large improvements**, thus significantly simplifying data collection for challenging scenarios

What We Do

- Establish a synthetic Intelligibility Preference Speech Dataset (INTP):
  - Comprising about 250K preference pairs (over 2K hours) of diverse domains
  - Building preference pairs using various strategies to avoid **reward hacking**:
    - Using different open-source TTS systems with different generative models
    - Introducing perturbations to create human-guided negative samples (guided by human knowledge and LLMs)

What We Do

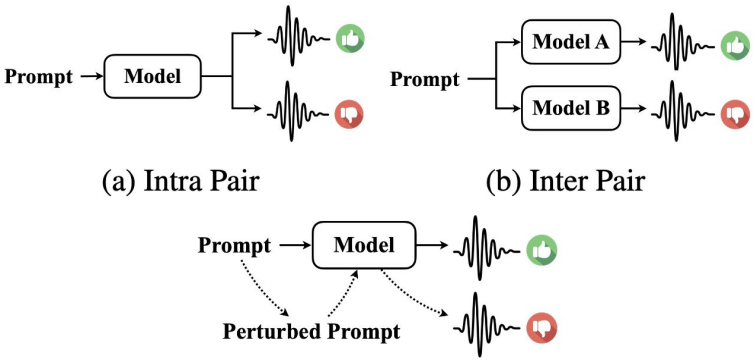
- Adopt DPO to enhance TTS systems:
  - Employing the vanilla DPO algorithm for AR-based TTS models
  - Extending DPO to flow-matching and masked generative models (aka. discrete diffusion models)
- A Key Question: Why do we use performance alignment instead of directly optimizing single metrics or rules such as WER?
  - **Incorporating human feedback beyond WER**: preference datasets allow us to inject human knowledge beyond WER, such as using human-guided negative samples to better reflect real-world preferences
  - We can create preference pairs using different models to increase data diversity and reduce the risk of **reward hacking** common in single-metric optimization

INTP Dataset: Overview

- Each triplet comprises:
  - Prompt: consisting of **target text** and **reference speech**
  - A pair of synthesized speech samples: the **preferred (positive)** and **dispreferred (negative)** outputs conditioned on the prompt
- Prompt construction:
  - Prompt text: regular, repeated, code-switching (both monolingual and cross-lingual)
  - Reference speech: cover a wide range of speakers
  - Data source: Emilia-Large (He et al., 2024, 2025)
- Model selection:
  - ARS (Wang et al., 2025a): discrete tokens, AR for text-to-token, NAR for token-to-waveform
  - F5-TTS (Chen et al., 2024c): continuous tokens, NAR (flow matching)
  - MaskGCT (Wang et al., 2025a): discrete tokens, NAR (masked generative models)

INTP Dataset: Preference Pairs Construction

- Three categories of pairs
  - **Intra pair**: sampling multiple times, filtering
  - **Inter pair**: choosing “best of the best” samples
  - **Perturbed pair**: leveraging human knowledge and LLMs to create human-guided negative samples (pronunciation and punctuation)



System Prompt:
假设你是一个 Text To Speech (TTS) 领域的专家，现在，让我们对一个 TTS 系统进行攻击。具体地：我输入一个文本，请你修改这条文本里面的若干词语，从而使 TTS 系统更容易出错。例如：你可以修改为把某些字修改为容易读错的形近字、把多音字做替换，等等，但你不要增加和删除原有的文本。注意：你只需要返回给我转换后的结果，不需要任何解释。

例子1:
【我的输入】我今天很高兴
【你的输出】窝锦添狠搞醒

例子2:
【我的输入】目前，爱心人士正在种作寄养的小猫已经五个月大了。而本人的种作寄养申请单需要进一步审核。为了避免小猫多次转手，治疗者们对小猫的种作寄养提出了严格要求：申请人需年满二十三岁。
【你的输出】幕前，爱信人士正在重作寄扬的削猫已经伍个月大了。而本人的重作寄扬神情但需要进一步审核。为了闭面削猫多次转售，治理者们对削猫的重作寄扬提出了阉割要求：申情人需年慢贰拾叁岁。

INTP Dataset

	Regular	Repeated	Code-Switching	Pronunciation-perturbed	Punctuation-perturbed	#Total
ARS (Wang et al., 2025a)	8,219	8,852	8,300	7,325	8,036	40,732
F5-TTS (Chen et al., 2024c)	8,425	8,555	7,976	7,909	6,667	39,532
MaskGCT (Wang et al., 2025a)	9,055	10,263	8,289	7,604	7,686	42,897
Intra Pairs	25,699	27,670	24,565	22,838	22,389	123,161
Inter Pairs	27,008	27,676	24,651	25,045	23,970	128,350
#Total	52,707	55,346	49,216	47,883	46,359	251,511

Distribution of preference pairs

Text Type	Example
Regular	*A panda eats shoots and leaves.*
Repeated	*A panda panda eats shoots and leaves and leaves and leaves.*
Code-Switching	熊猫吃 *shoots* 和 *leaves*。
Pronunciation-perturbed	*A pan duh eights shots n leafs.*
Punctuation-perturbed	*A panda eats, shoots, and leaves.*

Examples of different types for a text, *“A panda eats shoots and leaves”*

INTP Dataset: Human Perception Verification

	ARS	F5-TTS	MaskGCT	CosyVoice 2
Positive Samples	73.0%	88.1%	90.9%	100.0%
Negative Samples	45.7%	15.8%	47.1%	75.0%
All	59.7%	53.7%	64.3%	90.4%

Human-annotated reading accuracy.

	Naturalness Winner	Naturalness Tie	Naturalness Loser
INTP winner	72%	15%	13%

Agreement between INTP preference and human naturalness preference.

Methods: DPO for TTS

- SFT loss:

$$\mathcal{L}_{\text{SFT}}(\pi_\theta) = -\mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\log \pi_\theta(y_w \mid x)\right]$$

- DPO loss:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

- DPO update:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big]$$

KL SFT Unlearning

Rafailov R, Sharma A, Mitchell E, Manning CD, Ermon S, Finn C. Direct preference optimization: Your language model is secretly a reward model. NeurIPS 2023.

Methods: Extending DPO to Flow-Matching

RL optimization objective

$$\max_{p_\theta} \mathbb{E}_{x,y\sim p_\theta(y|x)}[r(x,y)] - \beta D_{\mathrm{KL}}[p_\theta(y|x) \parallel p_{\mathrm{ref}}(y|x)]$$

Reward modeling

$$\mathcal{L}_{\mathrm{R}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x,y_w) - r_\phi(x,y_l)\right)\right]$$

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]$$

Rewriting RL optimization objective for FM

$$\max_{p_\theta} \mathbb{E}_{y_1\sim p_\theta(y_1|x),t,x}[r(y_1,x)]$$
$$- \beta\mathbb{D}_{\mathrm{KL}}[p_\theta(y_1|y_t,t,x)||p_{\mathrm{ref}}(y_1|y_t,t,x)]$$

Methods: Extending DPO to Flow-Matching

Rewriting RL optimization objective for FM

$$\max_{p_\theta} \mathbb{E}_{y_1 \sim p_\theta(y_1|x),t,x}[r(y_1,x)] - \beta \mathbb{D}_{\mathrm{KL}}[p_\theta(y_1|y_t,t,x)||p_{\mathrm{ref}}(y_1|y_t,t,x)]$$

DPO loss for FM

$$\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w,y_1^l,x)\sim\mathcal{D},t} \log\sigma\left(\beta\left(\log\frac{p_\theta(y_1^w|y_t^w,t,x)}{p_{\mathrm{ref}}(y_1^w|y_t^w,t,x)} - \log\frac{p_\theta(y_1^l|y_t^l,t,x)}{p_{\mathrm{ref}}(y_1^l|y_t^l,t,x)}\right)\right)$$

DPO loss for FM (v prediction)

$$\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w,y_1^l,x)\sim\mathcal{D},t} \log\sigma\Big(-\beta \Big(\|v_\theta(y_t^w,t,x)-(y_1^w-y_0^w)\|_2^2 - \|v_{\mathrm{ref}}(y_t^w,t,x)-(y_1^w-y_0^w)\|_2^2\Big) - \Big(\|v_\theta(y_t^l,t,x)-(y_1^l-y_0^l)\|_2^2 - \|v_{\mathrm{ref}}(y_t^l,t,x)-(y_1^l-y_0^l)\|_2^2\Big)\Big)$$

Methods: Extending DPO to MGM

Rewriting RL optimization objective for MGM

$$\max_{p_\theta} \quad \mathbb{E}_{y_0 \sim p_\theta(y_0|x),t,x}\left[r(y_0,x)\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[p_\theta(y_0|y_t,x) \,||\, p_{\mathrm{ref}}(y_0|y_t,x)\right]$$

Reward modeling

$$\mathcal{L}_{\mathrm{R}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x,y_w) - r_\phi(x,y_l)\right)\right]$$

$$\mathcal{L}_{\text{DPO-MGM}} = -\mathbb{E}_{(y^w,y^l,x)\sim\mathcal{D},t} \log\sigma\left(\beta\left(\log\frac{p_\theta(y_0^w|y_t^w,x)}{p_{\mathrm{ref}}(y_0^w|y_t^w,x)} - \log\frac{p_\theta(y_0^l|y_t^l,x)}{p_{\mathrm{ref}}(y_0^l|y_t^l,x)}\right)\right)$$

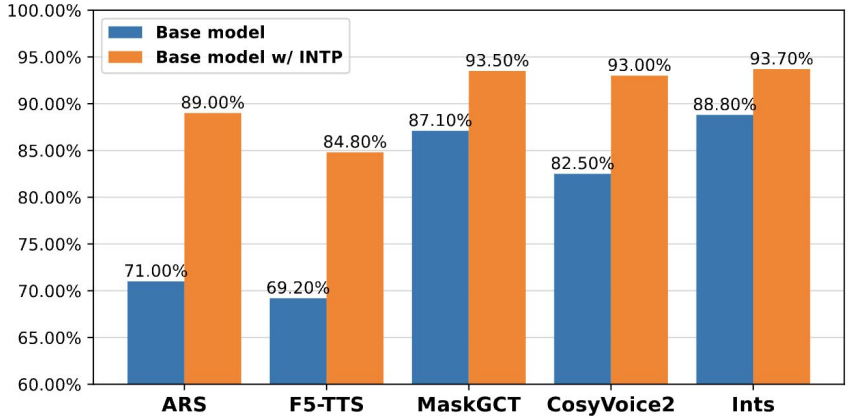
DPO loss for MGM

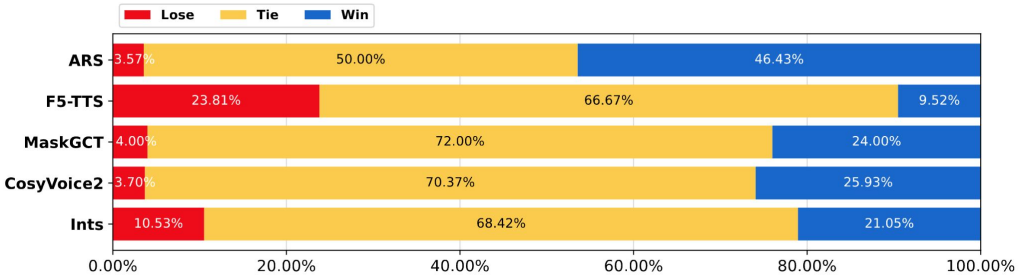
Results: Main and Weak-to-Strong Generalization

Model	Regular cases			Articulatory cases			Code-switching cases			Cross-lingual cases			Avg		
	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS
ARS	3.96	0.717	-	20.03	0.693	-	54.15	0.693	-	19.76	0.630	-	24.47	0.683	-
w/ INTP	2.32	0.727	0.47 ± 0.22	12.83	0.713	0.64 ± 0.31	36.91	0.698	0.63 ± 0.34	9.57	0.632	0.82 ± 0.28	15.41	0.692	0.64 ± 0.12
F5-TTS	3.44	0.670	-	16.84	0.635	-	33.99	0.609	-	16.86	0.546	-	17.78	0.615	-
w/ INTP	2.38	0.652	0.38 ± 0.26	12.97	0.628	0.30 ± 0.23	15.98	0.576	0.67 ± 0.36	7.13	0.509	0.47 ± 0.30	9.62	0.591	0.44 ± 0.12
MaskGCT	2.34	0.738	-	12.43	0.714	-	29.06	0.696	-	12.34	0.629	-	14.04	0.694	-
w/ INTP	2.23	0.737	0.23 ± 0.20	9.13	0.722	0.57 ± 0.36	19.70	0.704	0.19 ± 0.16	7.87	0.633	0.29 ± 0.18	9.73	0.699	0.32 ± 0.15
CosyVoice 2	2.09	0.709	-	8.12	0.696	-	33.36	0.672	-	8.78	0.600	-	13.09	0.669	-
w/ INTP	1.65	0.709	0.24 ± 0.25	6.87	0.696	0.20 ± 0.16	28.31	0.671	0.63 ± 0.30	5.39	0.603	0.28 ± 0.31	10.56	0.670	0.33 ± 0.12
Ints	3.14	0.688	-	12.08	0.666	-	22.88	0.646	-	9.78	0.572	-	11.97	0.643	-
w/ INTP	2.36	0.686	0.20 ± 0.36	9.38	0.664	0.11 ± 0.22	13.80	0.642	0.20 ± 0.38	6.28	0.571	0.18 ± 0.23	7.96	0.641	0.17 ± 0.15

Improvements of DPO with INTP for different models: ARS, F5-TTS, MaskGCT, CosyVoice 2, and Ints.

Results: Subjective Evaluation





Comparison of reading accuracy: INTP alignment enhances all five models in terms of both intelligibility.

Comparison of speaker similarity: INTP alignment also enhances similarity (more Tie/Win percentages).

Samples: INTP Dataset and INTP Alignment

- Demo page: https://intalign.github.io/

More Findings and Discussion

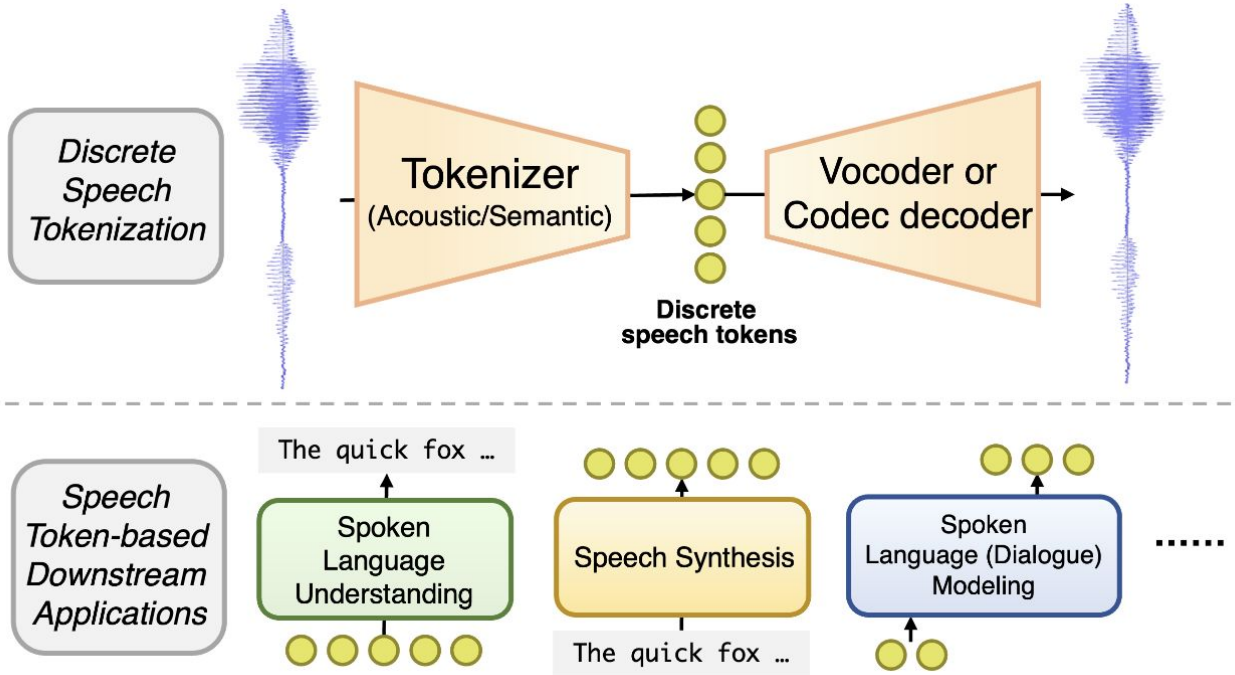
- Generalizability
  - Intelligibility preference alignment can also improve **naturalness, speaker similarity and quality**
  - Intelligibility preference alignment can generalize between **different languages**
- Directly applying SFT on positive samples
  - Not very useful
- GRPO or other online RL algorithms
  - We have tried to train a reward model based on INTP and tried to optimize ints with GRPO and REINFORCE
- Effect of PA on other speech generation tasks

Contents

- **Designing Generative-Friendly Speech Representations**
  - Introduction of ***TaDiCodec***, a speech tokenizer designed from the perspective of speech synthesis, with a single codebook and a token rate of 6.25 Hz
    - Paper: https://hecheng0625.github.io/assets/pdf/Arxiv_TaDiCodec.pdf
    - Demo: https://tadicodec.github.io/

Li J, Lin X, Li Z, Huang S, Wang Y, Wang C, Zhan Z, Wu Z. DualCodec: A Low-Frame-Rate, Semantically-Enhanced Neural Audio Codec for Speech Generation. Interspeech 2025.

Background: Discrete Speech Tokens

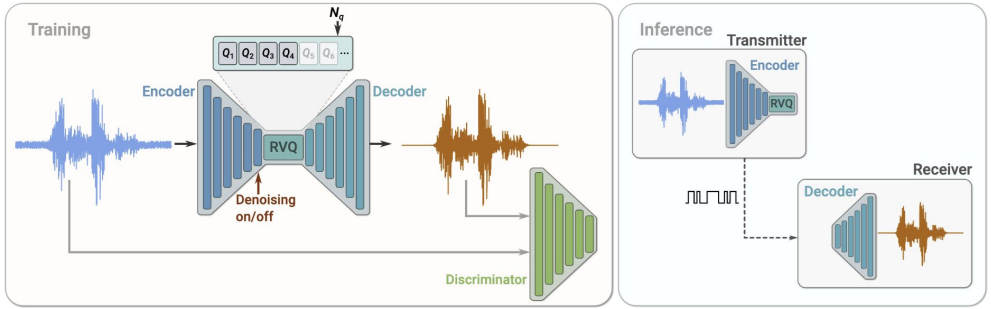


Y. Guo, Z. Li, H. Wang, B. Li, C. Shao, H. Zhang, C. Du, X. Chen, S. Liu, and K. Yu. Recent advances in discrete speech tokens: A review. ArXiv preprint arXiv:2502.06490, 2025.

Background: Brief Overview of Previous Works

- The early speech tokenizers primarily:
  - designing for compression and transmission
  - relying on multi-layer residual vector quantization (RVQ)
  - operating at high frame rates (e.g., over 50 Hz) and bitrate
  - focusing on acoustic information reconstruction

These characteristics make modeling with language models challenging and inefficient



Zeghidour N, Luebs A, Omran A, et al. Soundstream: An end-to-end neural audio codec. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021.

Background: Brief Overview of Previous Works

- Some works have explored techniques for single codebook:
  - e.g., BigCodec, TAAE, WavTokenizer…
  - still fall short in reconstruction quality compared to RVQ-based tokenizers
  - often maintain high token rates (BigCodec: 80 Hz, WavTokenizer: 65 Hz)
  - depend on complex loss designs and adversarial training and primarily optimize for **acoustic-level reconstruction**

Background: Brief Overview of Previous Works

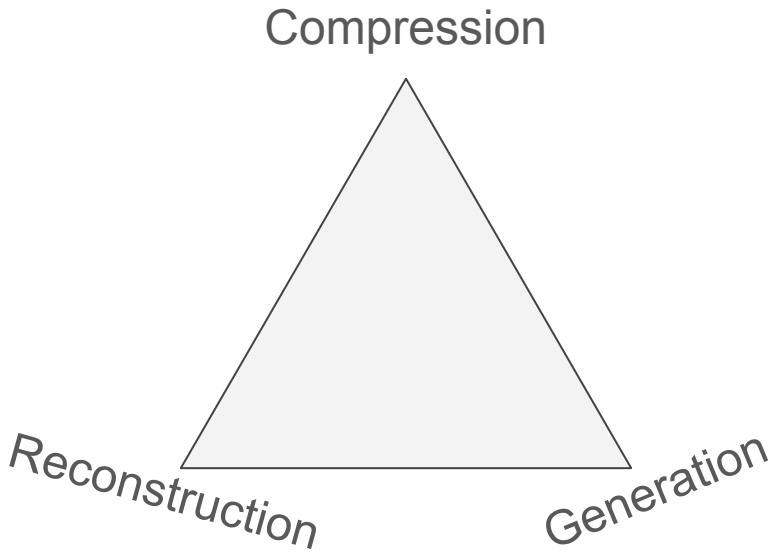
- Effective speech tokens for language modeling should exhibit **low frame rates** and **semantic richness**:
  - Enhancing semantics via **semantic distillation**: SpeechTokenizer, Mimi, DualCodec…
  - Adopting a **two-stage design**:
    - quantizing SSL-derived features
    - training a diffusion model to reconstruct speech from the tokens
    - e.g., CosyVoice, SeedTTS, FireRedTTS, Vevo…
    - several limitations: two-stage training (not end2end), external dependency, struggle with extreme compression…

Motivation: Designing Speech Tokenizers for SLM

- Elucidating the design space of speech tokenizers from the perspective of speech language modeling:
  - **Compression**: single codebook, low frame rate, low bitrate (mainly depends on the first two)
  - **Reconstruction**: capable of high-quality reconstruction in terms of intelligibility, similarity, and sound quality
  - **Generation**: can be easily modeled by language models (e.g., AR-based TTS models)
  - **Understanding**: providing sufficient information for understanding (beyond today's scope)
  - **Other good properties**: end2end training, simple architecture and training objective…

Motivation: Designing Speech Tokenizers for SLM

- Elucidating the design space of speech tokenizers from the perspective of speech language modeling:



Motivation: Designing Speech Tokenizers for SLM

- 🎯 Our goal in designing speech tokenizers:
  - achieving a **sufficiently low frame rate** for efficient modeling (6.25 Hz to 12.5 Hz)
  - enabling high-quality speech reconstruction
  - facilitating generation tasks: Language models can predict discrete tokens, which are then decoded into speech by the tokenizer decoder

Key challenge: to achieve high-quality reconstruction, tokens must carry **rich information**, including: semantic, prosody, timbre, and acoustic details under **significant information bottleneck**.

Motivation: Designing Speech Tokenizers for SLM

- Rethinking speech language modeling…
  - In zero-shot TTS scenarios: the target text is inherently available
  - In end-to-end spoken language systems: speech and text tokens are often generated jointly

Our insight: Text and (speech prompts) can be directly used as inputs to the tokenizer decoder, greatly reducing the information burden on tokens at low token rates.

Motivation: Designing Speech Tokenizers for SLM

- From the perspective of generative models:
  - Most previous tokenizers mainly relied on GANs, which perform well at high token rates but struggle at low token rates:
    - weak in completing missing information
    - unstable during training
    - less capable of effectively absorbing prompts
  - We can use more powerful generative models, such as diffusion models

We use an **end-to-end trained diffusion autoencoder** to jointly optimize quantization and reconstruction, and **incorporate text and speech prefixes as prompts**.

Overview of TaDiCodec

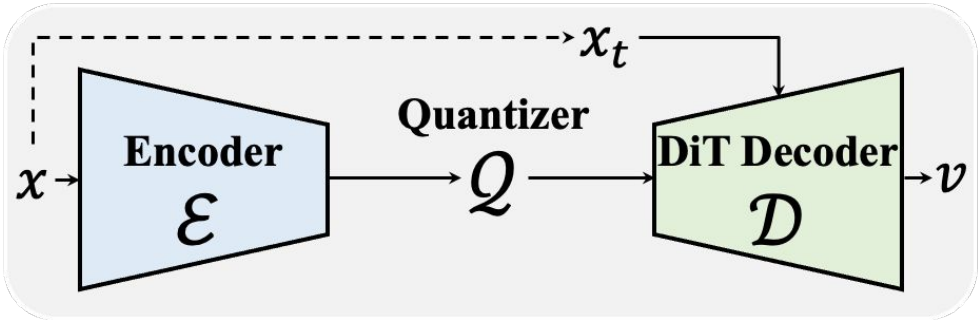
Extremely compression: achieves a frame rate of **6.25 Hz** using a **single codebook** (6.25 tokens one second), corresponding to a bitrate of **0.0875 kbps** for 24 kHz speech.

Unifies quantization and reconstruction within an **end-to-end training diffusion autoencoder**, removing the need for separate semantic distillation complex adversarial objective.

Text-aware speech tokenizers: enhances reconstruction quality and compression efficiency by **incorporating text and prompt guidance** into the diffusion decoder.

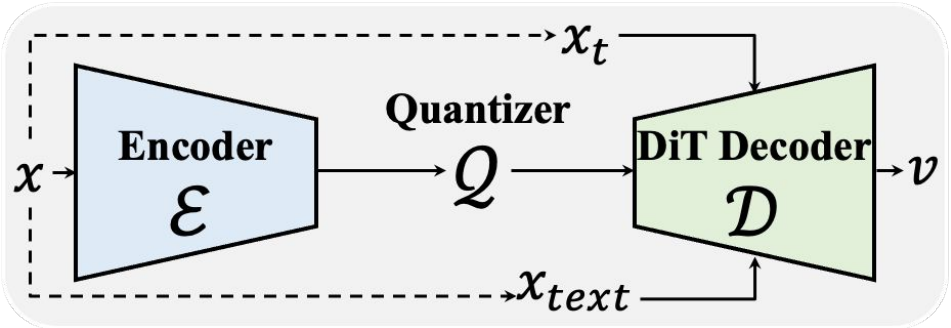
Method: Diffusion Transformer Autoencoder

- Simple model architecture:
  - Fully Llama-style transformers are used for both the encoder and the diffusion decoder
  - Timesteps and text are provided as additional inputs to the decoder
- Simple training objective:
  - End2end training for quantization and reconstruction
  - The model is trained using only flow-matching loss on the mel-spectrogram
  - Mel-to-waveform conversion is trivial with a vocoder



Method: Text-aware De-Tokenization

- Text-aware de-tokenization strategy:
  - Using the corresponding text sequence to guide the diffusion decoder and further improve reconstruction quality at extremely low compression rates
  - We directly add text to the prefix of the speech representation sequence (like VALLE, MaskGCT)



Method: Some Tricks

- Use **Binary Spherical Quantization (BSQ)** for quantization:
  - Does not rely on an explicit learnable codebook
  - Quantization error is bounded, can be trained without a commitment loss
- Prompt mechanism:
  - Randomly sample a prefix from the input mel-spectrogram as prompt without adding noise
  - During inference, we can use classifier free guidance (cfg) to improve reconstruction

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}_{text}),\boldsymbol{\epsilon},t}\left[\|(\boldsymbol{x}-\boldsymbol{\epsilon}) - \mathcal{D}_{\phi}(Q(\mathcal{E}_{\theta}(\boldsymbol{x})),\boldsymbol{x}_t,t,\boldsymbol{x}_{text})\|\right]$$

Method: Ablation Study

System	Recon. *Seed en*			Recon. *Seed zh*		
	WER	**SIM**	**UTMOS**	**WER**	**SIM**	**UTMOS**
TaDiCodec	3.02	0.67	3.68	1.11	0.74	2.70
bsq → vq	3.30	0.64	3.44	1.25	0.72	2.46
w. prompt → wo. prompt	8.63	0.52	3.26	5.42	0.59	2.28
decoder size: 320M → 160M	7.96	0.63	3.60	2.02	0.73	2.89
decoder size: 320M → 480M	2.90	0.69	3.68	1.02	0.75	2.73
frame rate: 6.25 hz → 12.5 hz	2.57	0.69	3.58	1.09	0.75	2.68
Inference steps: 50	2.87	0.68	3.66	1.07	0.75	2.68
Inference steps: 10	3.85	0.67	3.65	1.23	0.74	2.69
Inference steps: 5	7.89	0.65	3.19	1.96	0.73	2.35
w. decoder continued-training	2.73	0.69	3.73	0.94	0.75	2.69

Method: Zero-Shot TTS with TaDiCodec

- Training:
  - We train both AR-based and MGM-based (NAR) TTS models
  - We train different sizes: 0.2B, 0.5B, 1B, 3B, 4B…
- Evaluation:
  - We evaluate the models on SeedTTS test sets and INTP test sets (more challenge)

Result: Speech Reconstruction

System	Frame Rate	Token Rate	Bitrate (kbps)	Codebook Number	Semantic Distill Free	Reconstruction Quality		
						WER (↓)	SIM (↑)	UTMOS (↑)
Token rate less than 150								
EnCodec [19]	75	150	1.5	2	✓	5.36	0.48	1.54
DAC (RVQ) [21]	25	75	0.75	3	✓	20.08	0.39	1.75
DAC (VQ) [21]	75	75	0.75	1	✓	12.74	0.45	2.08
SpeechTokenizer [27]	50	100	1	2	✗	7.98	0.46	2.47
Mimi [10]	12.5	75	0.825	6	✗	4.51	0.52	3.09
			1.1	8	✗	3.99	0.57	3.21
DualCodec [26]	12.5	75	0.925	6	✗	2.63	0.62	3.78
			1.225	8	✗	2.57	0.64	3.78
BiCodec [6] *16 kHz*	50	50	0.65	1	✗	3.05	0.61	3.68
X-codec 2 [5] *16 kHz*	50	50	0.8	1	✗	2.63	0.62	3.68
WavTokenizer [23]	75	75	0.9	1	✓	6.65	0.48	3.36
BigCodec [22] *16 kHz*	80	80	1.04	1	✓	3.25	0.61	3.59
TAAE [24] *16 kHz*	25	25	0.4	1	✓	11.08	0.41	3.87
Two stage, Diffusion decoder								
SemantiCodec [40]	25	50	0.675	2	✗	5.11	0.49	2.83
Vevo Tokenizer [32]	50	50	0.65	1	✗	3.04	0.53	3.50
FireRedTTS Tokenizer [4]	25	25	0.35	1	✗	3.35	0.59	3.40
CosyVoice Tokenizer [3]	25	25	0.3	1	✗	5.63	0.47	3.65
CosyVoice 2 Tokenizer [41]	25	25	0.325	1	✗	4.10	0.68	3.65
Token rate less than 20								
Two stage, Diffusion decoder								
Ints Tokenizer [68]	12.5	12.5	0.175	1	✗	7.14	0.67	3.37
One stage, Diffusion decoder								
TaDiCodec	**6.25**	**6.25**	**0.0875**	**1**	✓	**3.02**	**0.67**	**3.68**
TaDiCodec (w. dct)*	**6.25**	**6.25**	**0.0875**	**1**	✓	**2.73**	**0.69**	**3.73**

* "w. dct" denotes continued training of the decoder for 400K additional steps, with the encoder and VQ module frozen.

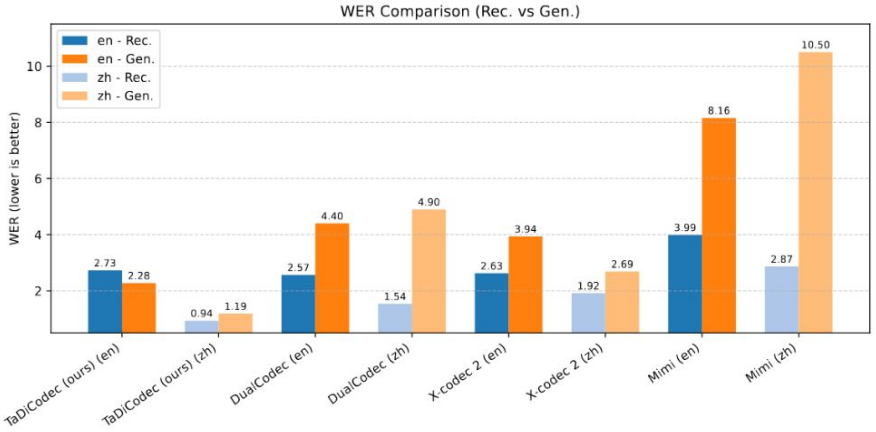
Result: Zero-Shot TTS

System	Frame Rate	Regular				Articulatory				Code-switching				Cross-lingual			
		en		zh		en		zh		en		zh		zh2en		zh2en	
		WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM
Baseline systems																	
NAR																	
MaskGCT [7]	50	2.40	0.71	2.28	0.77	14.50	0.69	10.35	0.74	38.39	0.63	19.73	0.76	8.47	0.70	16.22	0.56
F5-TTS [56]	93.75	3.02	0.63	3.87	0.71	14.13	0.61	19.54	0.66	35.35	0.54	32.63	0.68	19.93	0.64	13.78	0.46
AR																	
ARS [7]	50	3.55	0.68	4.37	0.75	15.98	0.68	24.07	0.71	48.59	0.63	59.71	0.76	15.22	0.70	24.30	0.56
CosyVoice 2 [41]	25	2.89	0.66	1.29	0.76	8.63	0.66	7.60	0.74	28.32	0.59	38.39	0.75	9.98	0.67	7.59	0.53
FireRedTTS [4]	25	8.53	0.46	1.27	0.65	14.47	0.45	18.81	0.64	15.03	0.38	23.97	0.63	3.87	0.34	9.04	0.48
Ints [68]	12.5	3.43	0.65	2.85	0.73	12.75	0.65	11.41	0.69	26.30	0.57	19.46	0.73	9.43	0.65	10.13	0.49
SparkTTS [6] *16 kHz*	50	2.50	0.57	1.78	0.66	10.19	0.57	13.37	0.65	15.12	0.46	16.86	0.65	9.73	0.58	4.88	0.40
Llasa [5] *16 kHz*	50	3.94	0.58	8.02	0.64	11.36	0.55	21.20	0.58	17.56	0.46	26.98	0.59	26.47	0.49	9.18	0.41
Ours																	
NAR																	
TaDiCodec-MGM *25 steps*	6.25	3.69	0.65	1.51	0.75	10.67	0.63	8.97	0.71	14.76	0.57	20.01	0.73	9.95	0.65	4.75	0.48
TaDiCodec-MGM *10 steps*	6.25	3.85	0.65	1.69	0.75	10.78	0.63	9.81	0.70	14.94	0.57	20.78	0.73	11.08	0.65	4.66	0.48
AR																	
TaDiCodec-AR	**6.25**	**2.28**	**0.65**	**1.19**	**0.75**	**8.23**	**0.63**	**8.74**	**0.70**	**9.16**	**0.57**	**16.09**	**0.73**	**7.67**	**0.64**	**2.91**	**0.48**

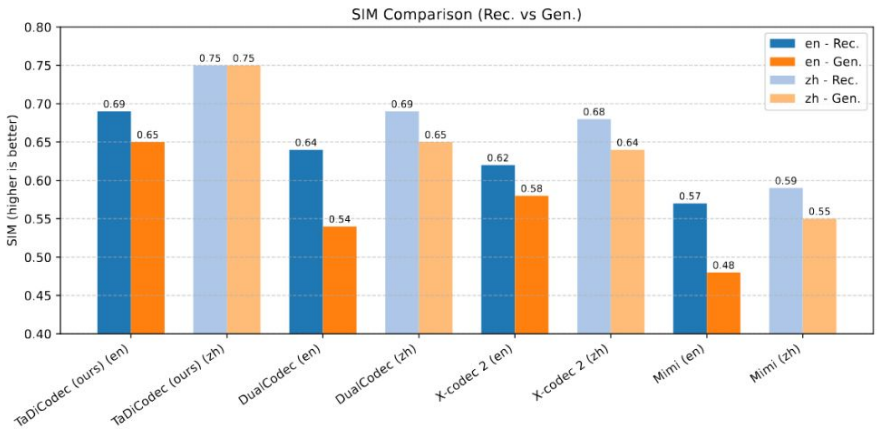
Result: RTF Analysis for TTS Model Size Scaling

System	Model Size	RTF	Regular				Articulatory				Code-switching				Cross-lingual			
			en		zh		en		zh		en		zh		zh2en		zh2en	
			WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM	WER	SIM
Baseline systems																		
CosyVoice 2 [3]	0.5B	0.47	2.89	0.66	1.29	0.76	8.63	0.66	7.60	0.74	28.32	0.59	38.39	0.75	9.98	0.67	7.59	0.53
SparkTTS [3]	0.5B	0.59	2.50	0.57	1.78	0.66	10.19	0.57	13.37	0.65	15.12	0.46	16.86	0.65	9.73	0.58	4.88	0.40
Llasa [41]	1.0B	0.42	3.94	0.58	8.02	0.64	11.36	0.55	21.20	0.58	17.56	0.46	26.98	0.59	26.47	0.49	9.18	0.41
Ours																		
TaDiCodec-MGM	0.6B	0.12	3.69	0.65	1.51	0.75	10.67	0.63	8.97	0.71	14.76	0.57	20.01	0.73	9.95	0.65	4.75	0.48
TaDiCodec-AR-0.2B	0.2B	0.20	7.68	0.64	1.48	0.74	16.06	0.63	12.54	0.70	16.38	0.56	23.91	0.72	13.40	0.64	4.26	0.48
TaDiCodec-AR-0.5B	0.5B	0.22	3.88	0.65	1.15	0.75	12.09	0.63	9.04	0.70	13.58	0.57	17.10	0.73	8.79	0.64	4.07	0.48
TaDiCodec-AR-3B	3.0B	0.25	3.24	0.65	1.23	0.75	8.34	0.63	8.52	0.70	11.31	0.57	15.47	0.73	7.85	0.65	3.99	0.48
TaDiCodec-AR-4B	4.0B	0.29	2.28	0.65	1.19	0.75	8.23	0.63	8.74	0.70	9.16	0.57	16.09	0.73	7.67	0.64	2.91	0.48
TaDiCodec-AR-4B w. vllm		0.13																

Result: Reconstruction and Generation Gap



(a) WER gap between reconstruction and generation.



(b) SIM gap between reconstruction and generation.

Samples: Reconstruction and TTS

- Demo page: https://tadicodec.github.io/

Conclusion and Future Vision

- Modeling
  - Discrete diffusion models (masked generative models) offer new opportunities to advance speech generation
  - Exploring preference alignment beyond intelligibility
  - Generation with thinking
- Representation
  - Build a unified representation for sound, speech, singing and music
  - Designing a unified representation for compression, generation, and understanding
- Omni Model
  - One model for all modality: text, vision, and audio
  - Especially for audio, we aim to build a model that can 1) understand and 2) generate audio, 3) engage in low-latency, end-to-end dialogue, and 4) speak before (or while) thinking