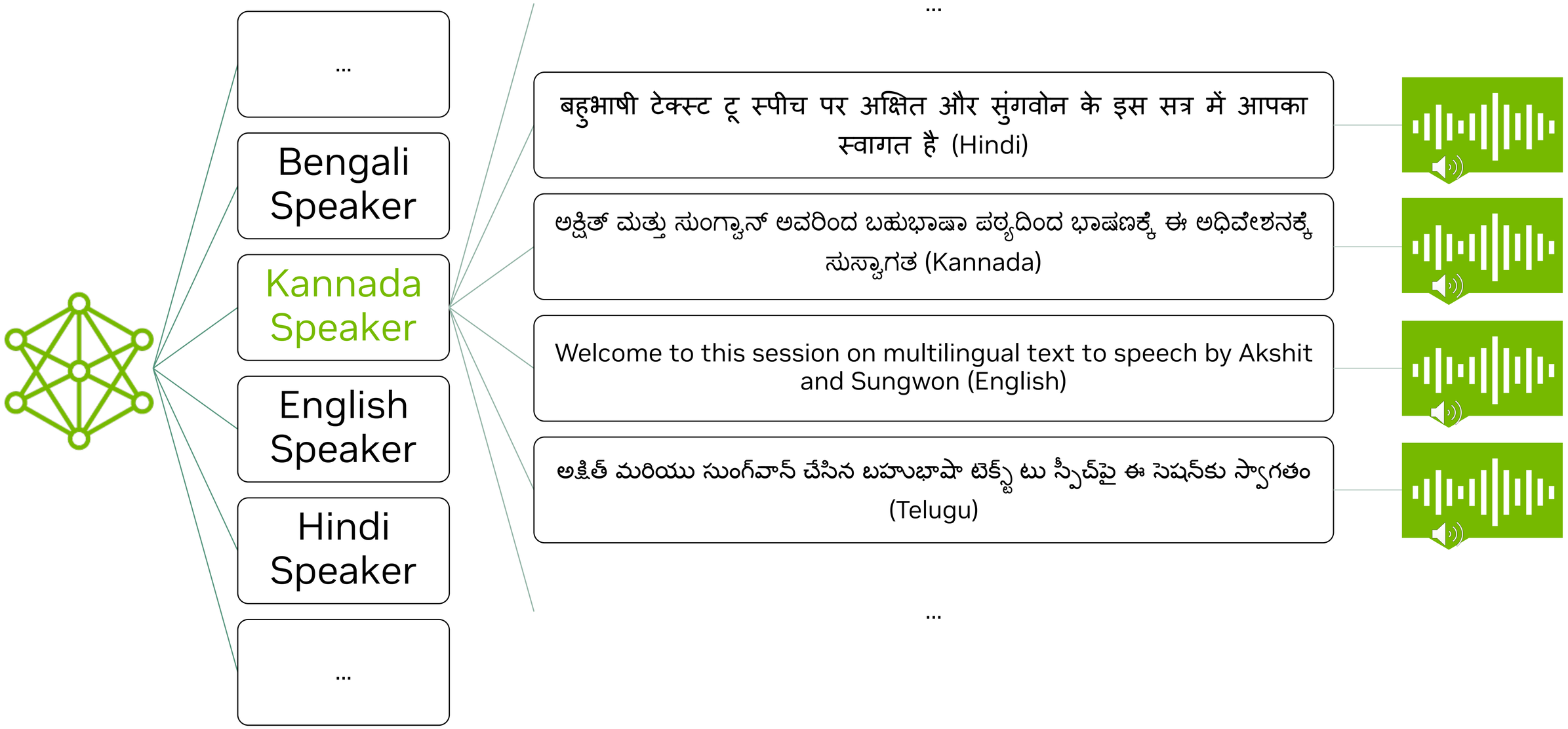
Scaling NVIDIA's Multi-Speaker Multi-Lingual TTS Systems with Zero-Shot TTS to Indic Languages

Akshit Arora, Rohan Badlani, **Sungwon Kim**, **Rafael Valle**, Bryan Catanzaro

ICASSP 2024 / 17 April 2024

RAD-MMM

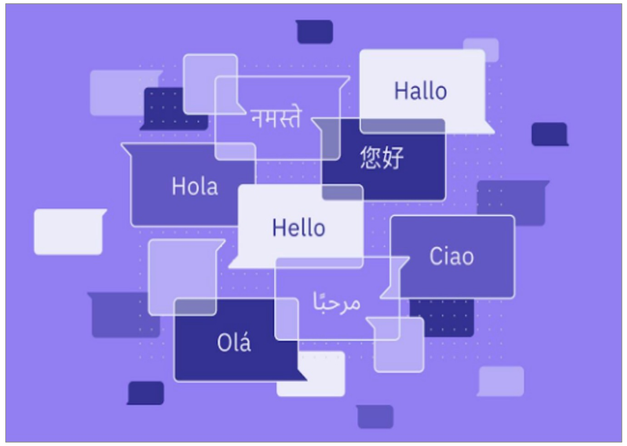
Language transfer with native accent and preserving speaker's voice



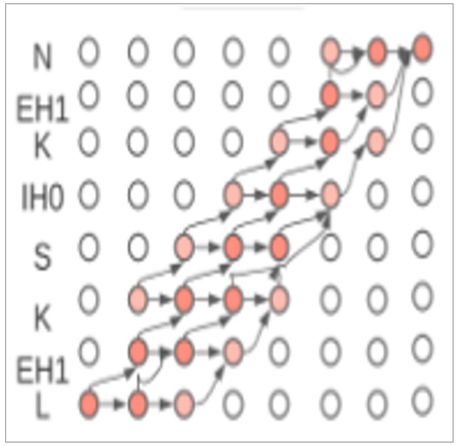
Why is multi-lingual multi-accented TTS hard ?

Technical Challenges

- **Major Technical Challenges:**



Each language has its own alphabet



Obtaining speech-text alignment is hard

Speaks English text

Speaks with en accent

Speaks with specific en prosody

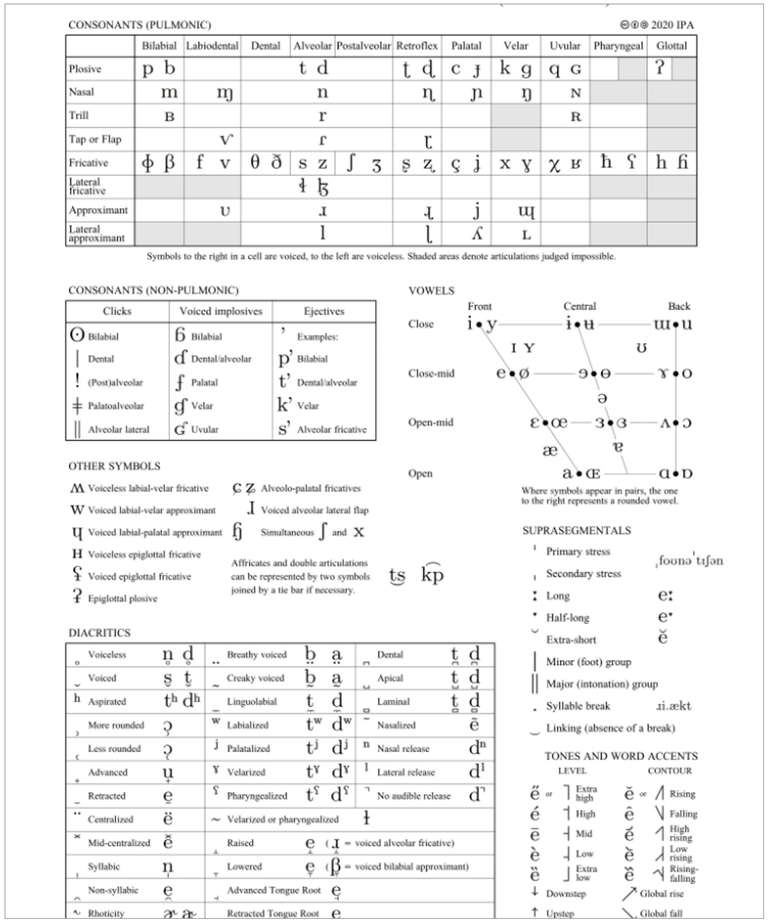
Entangled attributes

- **Desired Characteristics of Multi-lingual TTS System**

1. Control accent of the synthesized speech → remove user's accent when synthesizing in target language and apply target language accent.
2. Fine-grained prosody control → control over features like F0, emphasis, durations per phoneme, rhythm for flexibility.
3. One Model to Rule them all → all languages, all speakers, all accents. Should be easy to scale to new languages.

One Alphabet to Rule them All

Shared Text Representation for all languages



Source: https://en.wikipedia.org/wiki/International_Phonetic_Alphabet#/media/File:IPA_chart_2020.svg

Each language has independent alphabet. Shared alphabet set like IPA is beneficial to support:

Simplified text processing

Simplified speech-text alignment learning

Easy scaling to new languages

Shared token set **doesn't introduce entanglement between symbols and speakers**. (esp 1 spk/language)

Support **for code-switched text to speech** (mixed languages within same prompt)

One Alignment to Rule Them All

Speech-text alignment for accented TTS

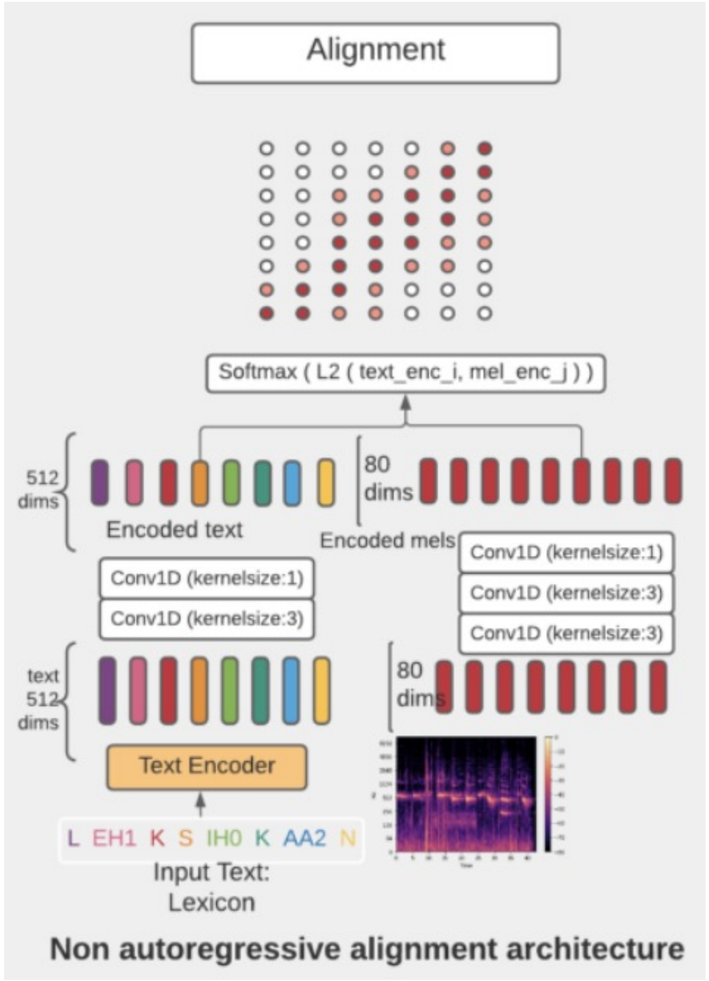
Alignment from TTS Model modelling likelihood

$P(s_t \mid x_t ; \theta)$

s_t: token at frame t
x_t: mel at frame t

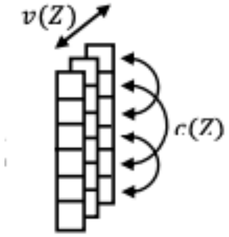
Speech has thick accent, condition on accent to handle multi-modality: same text can be spoken in different ways.

$P(<s_t, accent> \mid x_t ; \theta)$



Several disentanglement strategies to synthesize them all

Disentangling Attributes to synthesize (accent, speaker, language, text) for better quality synthesis



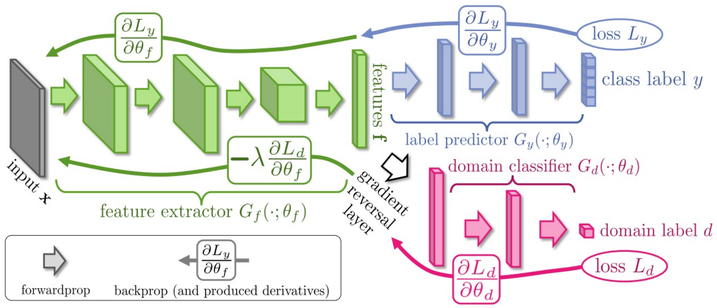
1. v(z): Maximize variance of embeddings

2. c(z): Minimize off-diagonal elements of covariance matrix for z

3. cross-corr(z1, z2) : Minimize covariance across attributes (spk, accent)

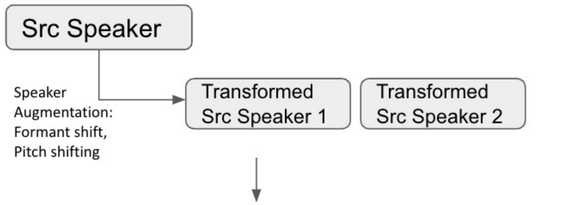
Regularization of embeddings

Boosts Speaker Identity Retention and more natural accent



Adversarial loss to disentangle text (language) and speaker

Boosts Speaker Identity Retention with slightly worse content quality



Data Augmentation

Boosts Speaker Identity Retention

Results

LANGUAGE TRANSFER

With few hours of training on speaker's Telugu data

Making a Hindi speaker speak Telugu without any Telugu data for speaker

LIghtweight, Multi-speaker, Multi-lingual Indic TTS – LIMMITS'23 – ICASSP

Rank	NATURALNESS		SPEAKER SIMILARITY	
	Team name	Average score	Team name	Average score
1	SJTU-X-LANCE	4.77	VAANI	3.98
2	VAANI	4.71	SIPLAB-IITH	3.96
3	SIPLAB-IITH	4.46	SJTU-X-LANCE	3.86
4	TSUP	4.4	TSUP	3.72
5	IIIT-TTS	4.27	IIIT-TTS	3.28
6	UTokyo-SaruLab	4.16	UTokyo-SaruLab	3.25

[*Naturalness: Score 5 = Human-like sound … Score 1 = Extremely intolerable]
[* Speaker similarity: Score 5 = Identical … Score 1 = Very different]

Telugu - ఒక్క అడుగు ప్రారంభిస్తే వేయి మైళ్ళ ప్రయాణమైనా పూర్తి అవుతుంది.
English - A journey of a thousand miles can be completed if one step is taken.

Total Participating teams: ~38
Total countries participating in challenge: 11

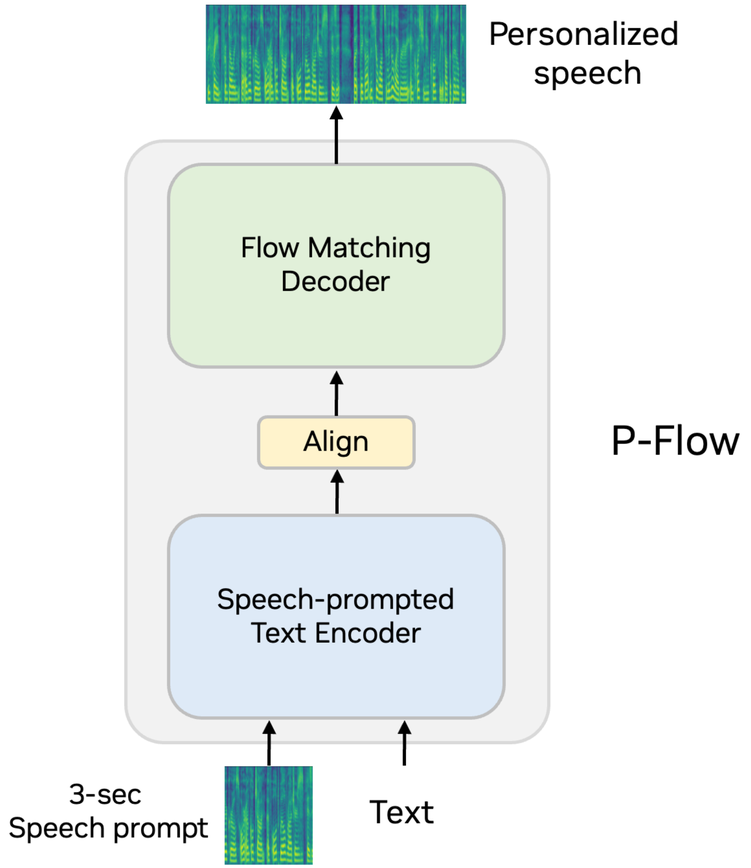
For more details: aroraakshit.github.io/vani

P-Flow: Non-autoregressive zero-shot TTS through speech prompting

Speech prompting mechanism + Non-autoregressive TTS

P-Flow: A Fast and Data-efficient Zero-shot TTS through Speech Prompting

- NAR-TTS model that takes 3-second reference data of the target speaker for zero-shot TTS
- P-Flow can perform zero-shot TTS without transcript of 3-second reference data



Rank 1 for both evaluation metrics in Track 3

NVIDIA

Background

Zero-shot Personalized Text-to-Speech

- **Zero-shot Personalized TTS (Track 3: Build zero-shot TTS for Indic languages)**
  - TTS model that generates personalized samples without fine-tuning given a reference speech data for the target speaker
  - **Goal**: High speaker similarity, Using only a small amount of reference speech, …

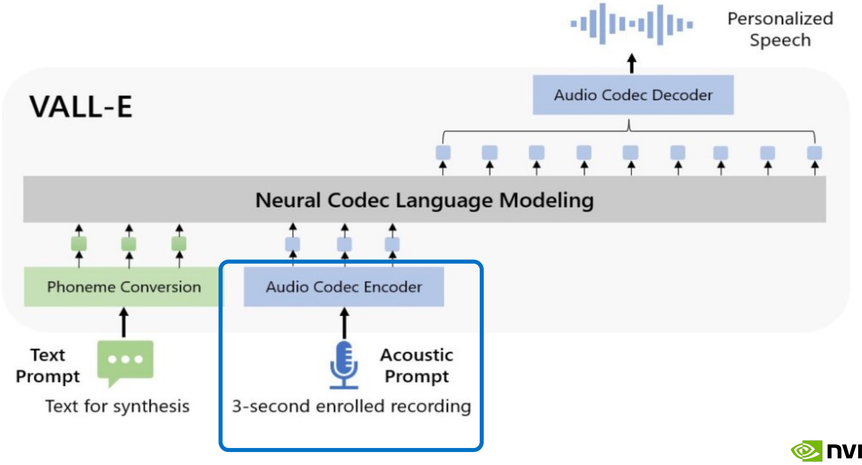
- **Speaker embedding approach**
  - Speaker encoder: Reference speech → speaker embedding $e_S \in R^d$
  - Rely on a single speaker embedding vector for zero-shot personalized TTS → **Low Speaker Similarity**

Reference speech

$$x \sim p(speech|text, e_S)$$

- **Language Model for TTS + Speech prompting approach**
  - **VALL-E, SPEAR-TTS:** Text-conditional Language Modeling for speech
  - **(Inference)** Speech prompting: Use reference speech as a prompt for LM
  - **Breakthrough in Zero-shot Personalized TTS**
    - 3-second reference speech → high speaker similarity

  - **(-) Slow sampling speed, Robustness issues**



P-Flow: A Fast and Data-Efficient Zero-shot TTS

Non-autoresgressive TTS + Speech prompting

- **P-Flow: A Fast and Data-Efficient Zero-shot TTS through Speech Prompting**
  - **Hypothesis:** Speech prompting-based speaker adaptation is the key for zero-shot TTS

1. Introduce **Speech prompting mechanism** into non-autoregressive zero-shot TTS
   - ***Speech prompted text encoder*** for speaker adaptation

2. Introduce **flow matching generative model**, for fast and high-quality speech generation
   - ***Flow matching generative model*** for fast speech generation

	VALL-E	P-Flow
Speech representation	Audio codec code	Mel-spectrogram
Generative Model	Language Model	Flow Matching Generative Model
Training Data	60,000 hours	260 hours
In-context Learning	O	O

→ **Fast**: 20x faster than VALL-E

→ **Data-Efficient**: Less than 0.01x VALL-E's training dataset

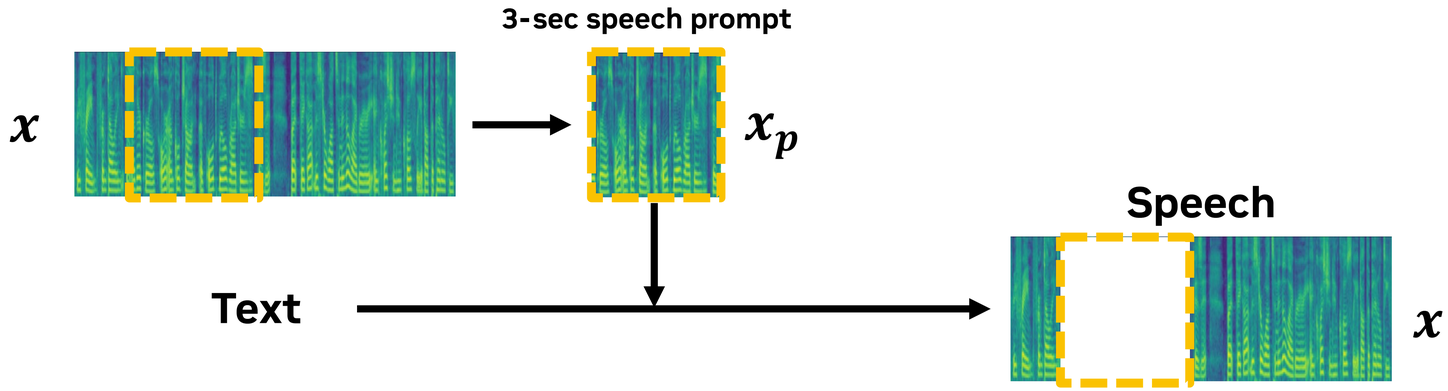
→ **Zero-shot**: Comparable to VALL-E

Speech prompting for non-autoregressive TTS model

- **Introduce speech prompting mechanism to non-autoregressive TTS model**
  - Provide 3-second reference speech as a prompt for zero-shot speaker adaptation.
  - **How?** Consider zero-shot personalized TTS as a masked-autoencoder

1. Sample 3-sec random segment $\boldsymbol{x_p}$ from the speech data $\boldsymbol{x}$
2. Reconstruct the data $\boldsymbol{x}$ from speech prompt $\boldsymbol{x_p}$ and text

  - To prevent learning copy & paste, mask the loss for prompted segment



P-Flow

Speech prompted text encoder + Flow matching generative decoder

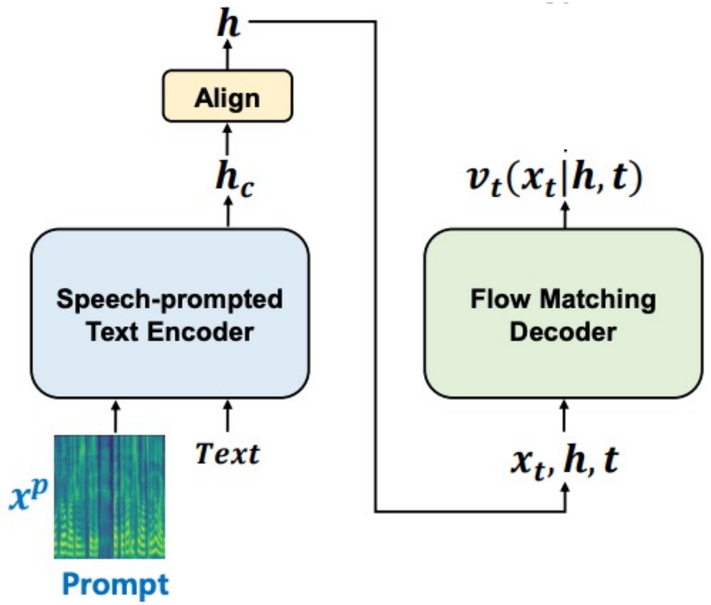
P-Flow: Speech prompted text encoder + Duration predictor + Flow matching generative decoder

1. ***Speech prompted text encoder*** for speaker adaptation
   - 3-sec speech prompt $\boldsymbol{x_p}$ + Text → Personalized hidden representation $\boldsymbol{h_c}$ → Expanded hidden representation $\boldsymbol{h}$

2. ***Flow matching generative model*** for fast speech generation
   - Personalized hidden representation $\boldsymbol{h}$ → Mel-spectrogram

   → **10 Euler steps (Significantly faster than VALL-E)**

For LIMMITS challenge, we used Diffusion Transformer (DiT) architecture as a decoder



P-Flow

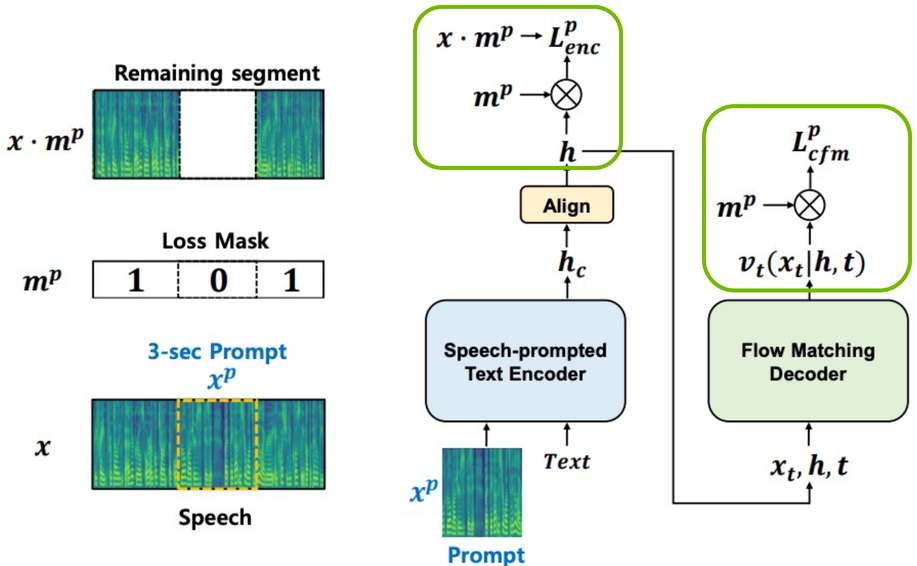
Speech prompted text encoder + Flow matching generative decoder

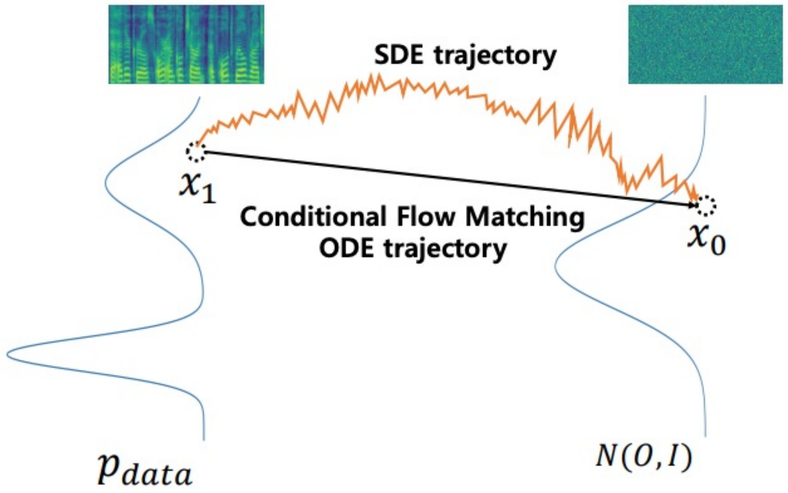
P-Flow: Speech prompted text encoder + Duration predictor + Flow matching generative decoder

1. ***Speech prompted text encoder***: Regression on $\boldsymbol{x}$ using the speech prompt $\boldsymbol{x_p}$ and text input
   - Encoder: 3-sec speech prompt $\boldsymbol{x_p}$ + Text → Expanded hidden representation $\boldsymbol{h}$
   - **Encoder loss:** $L^p_{enc} = L_2(h * m^p, x * m^p)$

2. ***Flow matching generative model***: Estimate the vector field $\boldsymbol{u_t}$ between the data and noise
   - Define the flow $\phi: p_0(x_0) \rightarrow p_1(x_1), x_1$: data, x_0: noise
   - Decoder v_t estimates the vector field of the flow $u_t = \frac{d\phi_t}{dt}$ given h
   - **Flow matching loss:** $L^p_{cfm} = L_2(v_t * m^p, u_t * m^p)$

$$x_0 \sim \mathcal{N}(0, I); \quad x_{t+\frac{1}{N}} = x_t + \frac{1}{N}\hat{v}_\theta(x_t, h, t) \quad \textbf{(Inference)}$$





Results

P-Flow: A Fast and Data-Efficient Zero-shot TTS !

- **Comparison with zero-shot TTS baselines**
  - **YourTTS:** VITS + Speaker embedding
  - **VALL-E:** Autoregressive LM for speech + Speech prompting
  - **P-Flow:** Flow matching model + Speech prompting, Sampling: 10 ODE steps

  - **Sample quality, Pronunciation accuracy:** P-Flow > VALL-E
  - **Fast**: 20x faster than VALL-E
  - **Data-Efficient**: Less than 0.01x VALL-E's training dataset
  - **Zero-shot**: Comparable to VALL-E

  → **P-Flow: A Fast and Data-efficient Zero-shot TTS !**

Model	Data (hours)	WER↓	SECS↑	Inference Latency(s)↓
GT (Hifi-GAN)		2.4	0.64	
YourTTS†	500+	7.7	0.337	
VALL-E†	60,000	5.9	**0.580**	2.515 ± 0.040
VALL-E Continual†	60,000	3.8	0.508	2.515 ± 0.040
P-Flow (proposed)	**260**	**2.6**	0.544	$\mathbf{0.115 \pm 0.004}$

Data-efficient **20x faster**

Model	CMOS↑	SMOS↑
P-Flow vs VALL-E	$\mathbf{0.27} \pm 0.10$	$\mathbf{0.23} \pm 0.13$

Zero-shot TTS for low-resource Indic Languages

Language transfer with native accent and preserving speaker's voice without fine-tuning

Track 3 in LIMMITS 2024: Zero-shot TTS for Indic Languages

Goal: Build **zero-shot TTS model for Indic languages**

Core challenge: The LIMMITS dataset has **a very limited number of speakers**

- 2 speakers (1 male, 1 female speaker) per each Indic language → a total of 14 speakers (560 hours of data)
- Allow to use external multi-speaker datasets for pre-training

We expanded NVIDIA's zero-shot TTS model, P-Flow, to 7 Indic languages of the challenge.

- **We ranked first in both naturalness and speaker similarity in Track 3.**

Train P-Flow on LibriTTS + LIMMITS dataset

Leveraging multi-speaker English dataset to achieve zero-shot personalization capabilities

How to build zero-shot TTS for Indic languages with P-Flow?

- **P-Flow:** Trained on LibriTTS performs well on zero-shot TTS in English

- **Expand P-Flow to enable zero-shot TTS in Indic languages**
  - How? Simply train P-Flow on LibriTTS + LIMMITS datasets

LIMMITS

2 speakers x 7 Indic languages
560 hours of data
→ Enable to pronounce Indic languages correctly

LibriTTS

2456 speakers, English
585 hours of data
→ Enable to learn zero-shot personalization capabilities

Track 3: P-Flow as NVIDIA submission (zero-shot with pretraining)

Team name	MOS(avg)	MOS(std)
NVIDIA	4.4	0.73
SJTU_XLANCE_VC	4.23	0.79
TalTech	3.93	1.16
reply_2024	3.12	1.16
Shabdh	3.09	1.1
LIMITLESS	2.82	1.42
nwpu	2.31	1.26

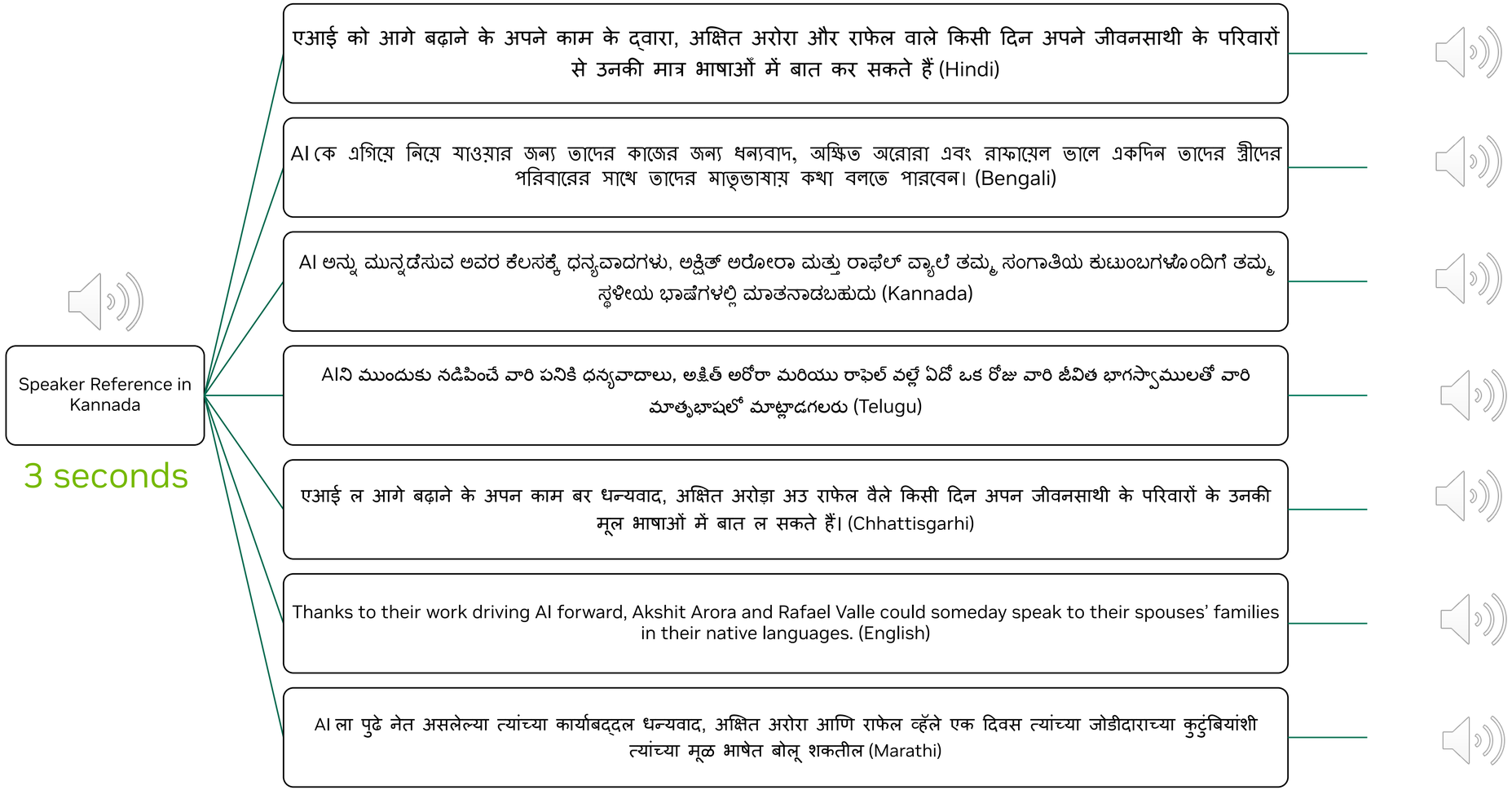
Naturalness of speech in target language

Team name	Score(avg)	Score(std)
NVIDIA	3.62	1.3076
Shabdh	3.44	1.3296
LIMITLESS	3.37	1.4172
TalTech	3.12	1.3261
reply_2024	3.04	1.27
nwpu	2.38	1.3003
SJTU_XLANCE_VC	2.26	1.1823

Speaker Preservation in target language

P-Flow

demo samples using 3 seconds from unseen speaker



RAD-MMM and P-Flow

Resources

Samples presented today: https://aroraakshit.github.io/mmitsvc-2024/

Multilingual Multiaccented Multispeaker TTS with RADTTS (RADMMM)

Rohan Badlani, Rafael Valle, Kevin J. Shih, João Felipe Santos, Siddharth Gururani, Bryan Catanzaro

https://arxiv.org/abs/2301.10335

Code & tutorial notebooks available here - https://github.com/NVIDIA/RAD-MMM

P-Flow: A Fast and Data-Efficient Zero-Shot TTS through Speech Prompting

Sungwon Kim, Kevin J. Shih, Rohan Badlani, João Felipe Santos, Evelina Bhakturina, Mikyas Desta, Rafael Valle, Sungroh Yoon, Bryan Catanzaro

https://openreview.net/pdf?id=zNA7u7wtIN

P-Flow will be available on Riva soon: https://www.nvidia.com/en-us/ai-data-science/products/riva/

ધન્યવાદ

ধন্যবাদ

Thank you!

ధన్యవాదాలు

ಧನ್ಯವಾದ

linkedin.com/in/aakshit/
linkedin.com/in/sungwon-kim-dsail/