CS753 SEMINAR PRESENTATION

Topic Name:

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

Team Details:

DO_DIN_MAI_RESEARCH_DOUBLE

Member	*ID*
Badri Vishal Kasuba	22M2119
Meet Narendra Doshi	22M0742
Sankara Sri Raghava Muddu	22M0764
Palash Dilip Moon	22M0816

PAPER DETAILS

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

Paper Links: OpenReview, NeurIPS Poster, NVIDIA ADLR Demo and GitHub

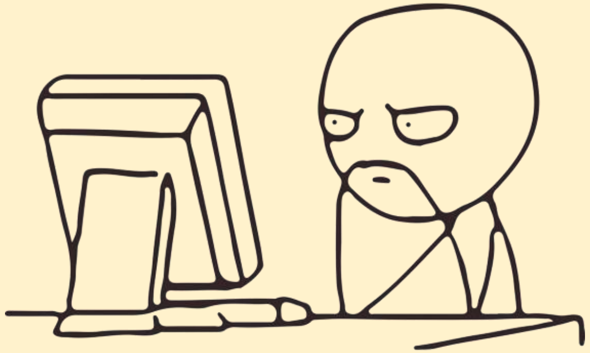
Paper Accepted at NeurIPS 2023 Conference

No. Of Citations: 1 till now

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

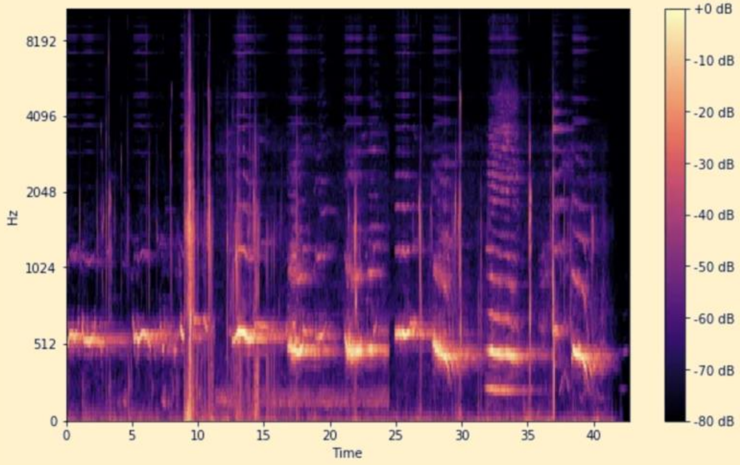
Affliation: NVIDIA, Seoul National University

Our Flow of P-Flow paper



KEY TERMS Introduction

1. What is **Zero-shot TTS** task?
2. What is meant by **Speech Prompt**?
3. What are ***mel-spectograms***? blog
4. What is a Neural Codec Language Model?: NCLM article
5. What are Normalizing Flows? Paper
6. What is **Flow Matching**? FM Explained
7. What is Simulation Free Training?
8. What are **WER, SECS, CMOS, SMOS** scores



Zero Shot TTS

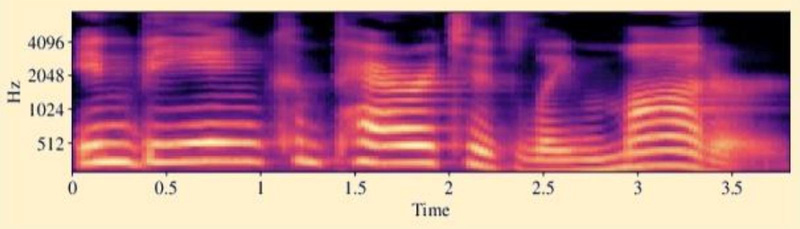
1. Reference speech segment *Xp (Speech Prompt)*:

   Eg: Voice of Trump

19. Trump on why people would vote for him

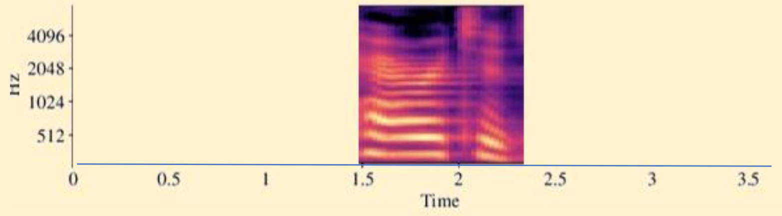
"To be blunt, people would vote for me. They just would. Why? Maybe because I'm so good looking."

New York Times, 19/9/99



Reference speech: *X*

Part of the reference speech: *Xp*



2. Text to speech input:

"Sorry losers and haters, but my IQ is one of the highest - and you all know it! Please don't feel so stupid or insecure, it's not your fault." : Trump did say this...

TTS Model

Voice generated by: https://fakeyou.com/tts

Background: Generative Models and TTS

- TTS Methods
  - **Cascaded** [An Acoustic model and Vocoder using mel-spectograms (intermediate)]
  - **End-to-End** [Jointly optimize Acoustic model and Vocoder]

- Most work has been done in the CASCADED setting for TTS from past
- Growing interest for Multi-lingual zero-shot TTS and also need of it
- Works include effective speaker encoding based methods, advanced speaker embeddings-based models were proposed

- Current Research trend in this area is
  - Efficient TTS models
  - Multi-lingual and Multi-Speaker adaptation both efficiently and with possible less samples of diverse speakers
  - As usual reducing the model cost and inference costs

Motivational background: Does efficient solution exists??

1. Large dataset
2. Complicated training setups
3. Additional pretraining tasks
4. Additional quantization steps
5. Computationally expensive autoregressive formulations

Basically
VALL-E

1. Simple Training pipeline
2. Significantly few data
3. Faster Inference
4. Good Performance
5. Retain high speaker similarity like VALL-E

The Proposal we need

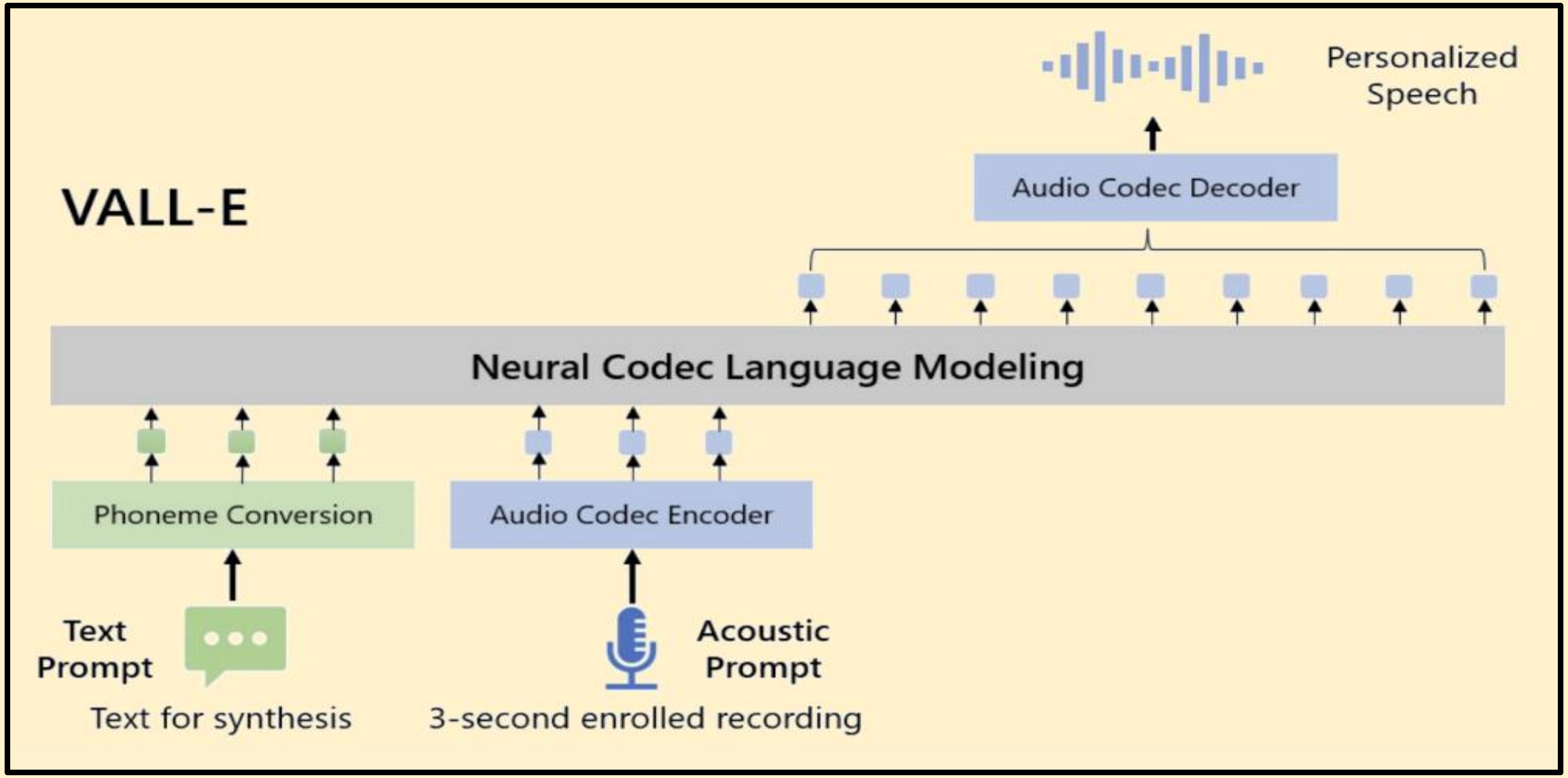
Pictures borrowed from : [link]

Previous Approaches

Name	Conference	Architecture	Training Data	Sampling speed	Problems
Vall-E	Microsoft Research	Language Model	LibriLight, LibriSpeech	Slow	AR, Codec
YourTTS	ICML	Transformer	VCTK, LibriTTS, MLS-PT	Slow	Instability in stochastic duration predictor, mispronunciations in Portuguese.
GlowTTS	Nips 2020	Transformer	LJspeech, LibriTTS	Fast	No prompting or Zero shot TTS
SpearTTS	TACL	Transformer cascaded	Librilight + LibriTTS	Very Slow	Cascaded decoupled system
GradTTS	ICML 2021	Diffusion based	LJSpeech	Fast	No prompting or Zero shot TTS
AudioLM	Google Research(2022)	Language Model	LibriLight, LibriSpeech		
A^3T	ICML 2022	Transformer, uses spectograms	LJSpeech, VCTK, LibriTTS		☹
StyleTTS2	Neurips 2023	Generative Model	VCTK, LibriTTS		

Background: VALL-E

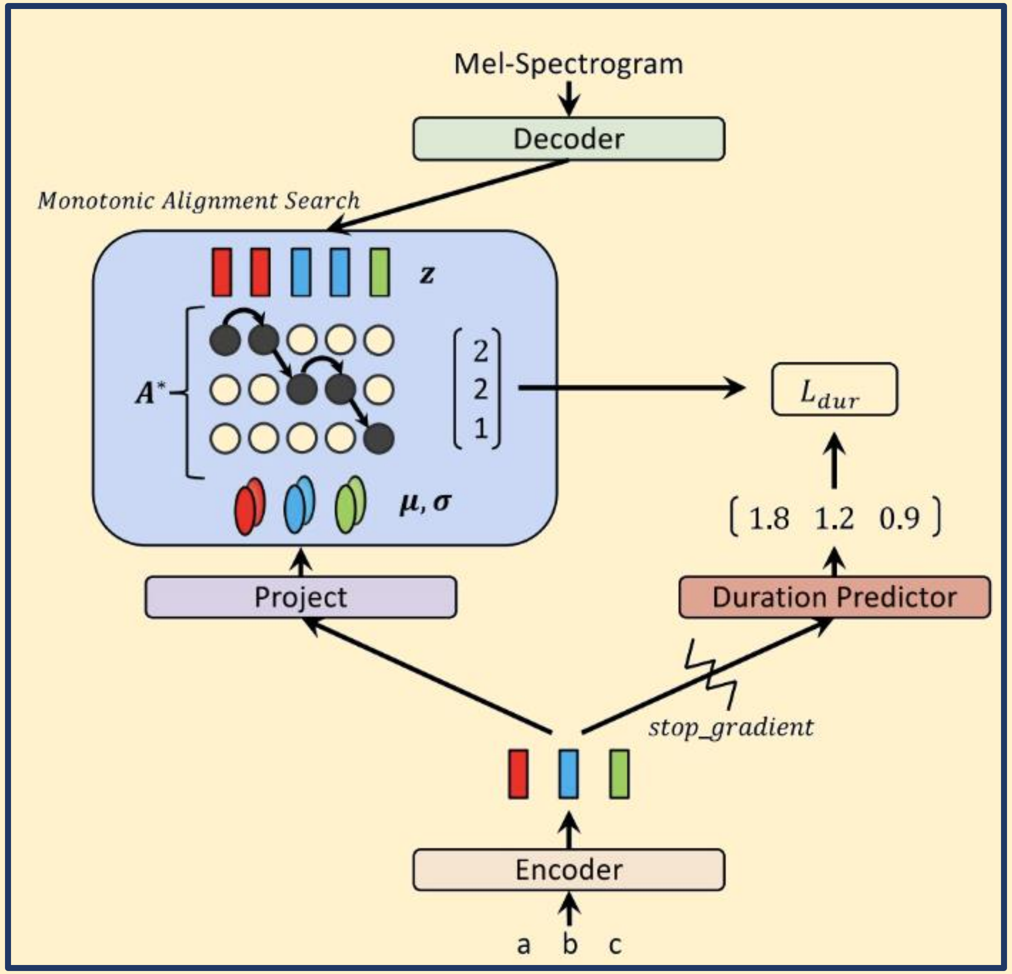
- VALL-E (Wang et al., 2023) [Link]



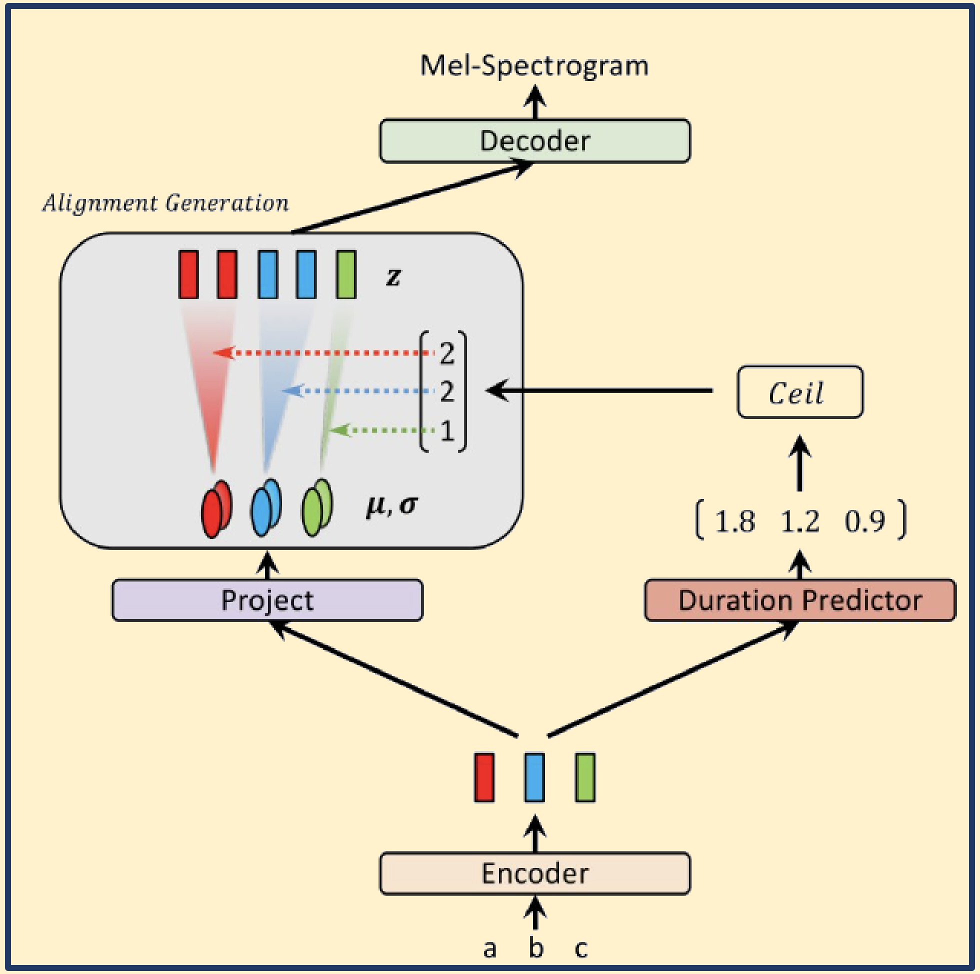
Background: Glow TTS

- GlowTTS (Kim et al., 2020) [Link]

GlowTTS Training



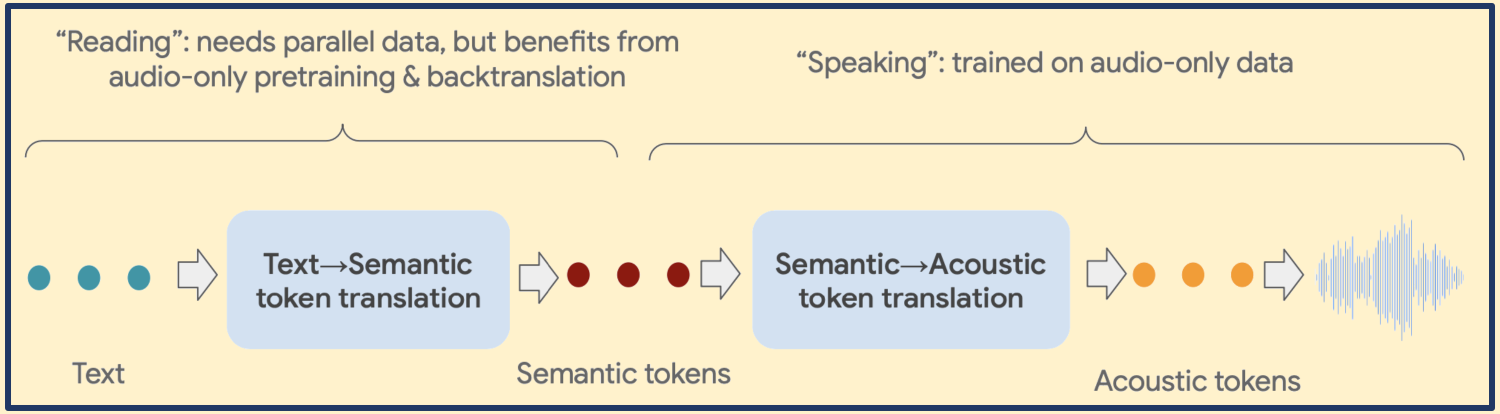
GlowTTS Inference



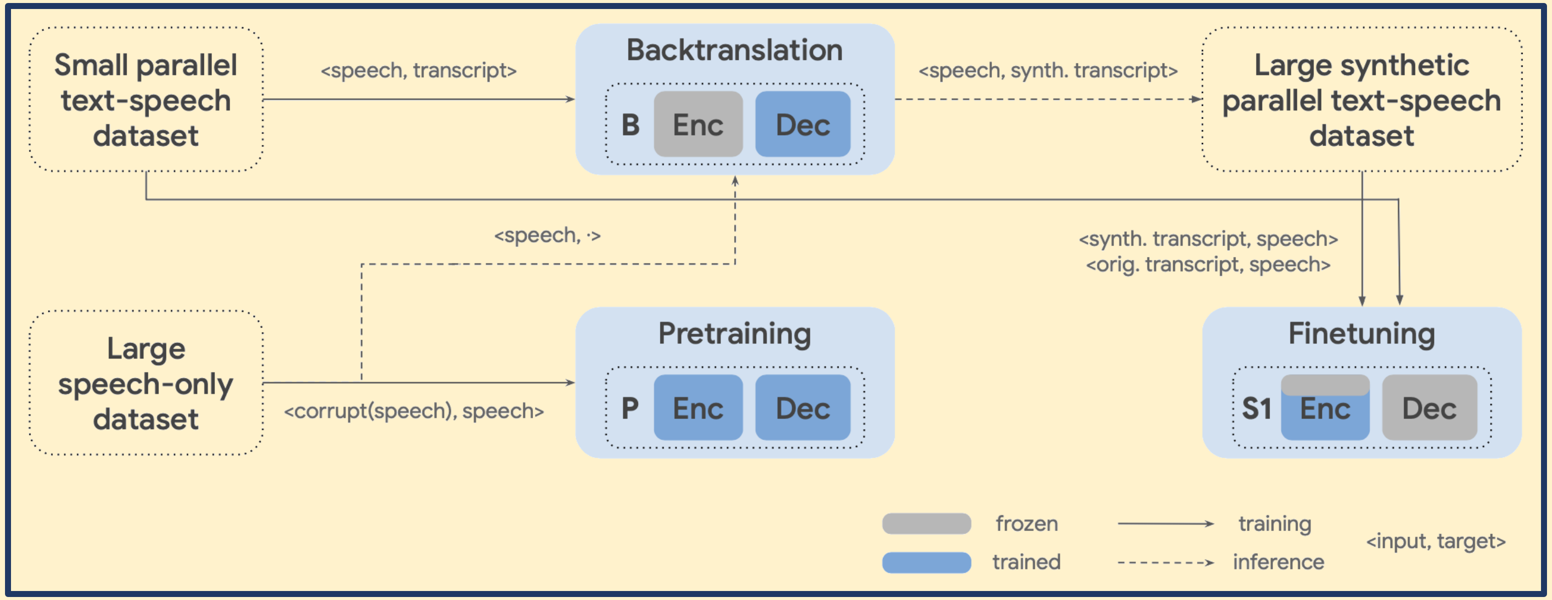
Background: SpearTTS

- Spear-TTS (Kharitonov et al., 2023) [Link]

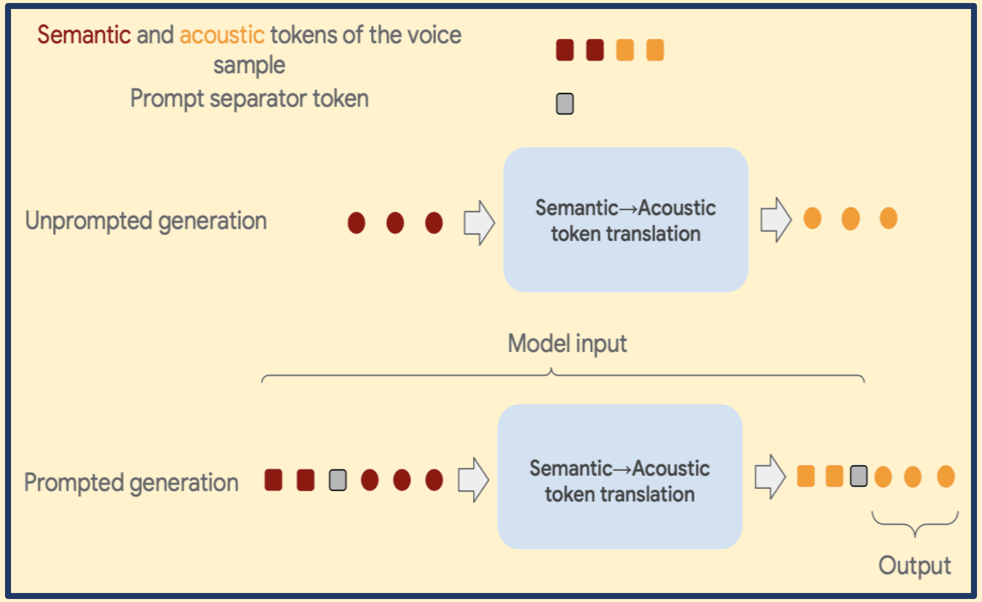
Pipeline: S1+S2



S1 Training



S2 Training



Observational Overview

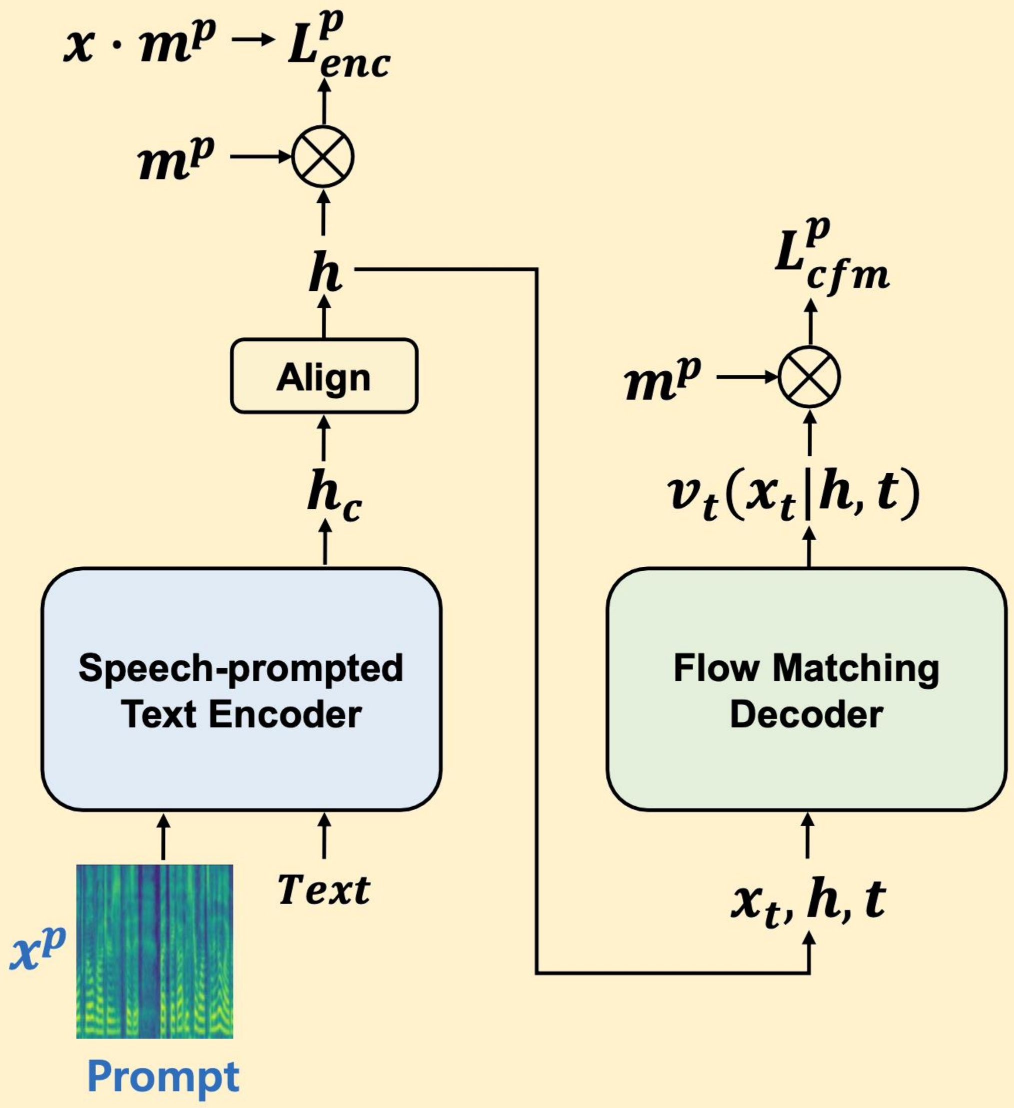
- **P-flow** raises a **challenge** to the recent trends of using feature extraction approaches for speech synthesis

- Main contribution is to achieve high speaker similarity performance with very less training and also achieving with fast inference, especially on zero-shot TTS task

- This with possible with the novel proposal of a speech prompted text encoder to generate speaker-conditional text representation for speaker adaptation

- Another contribution is the introduction of Flow matching based generative decoder for fast and efficient speech synthesis with even very few training data

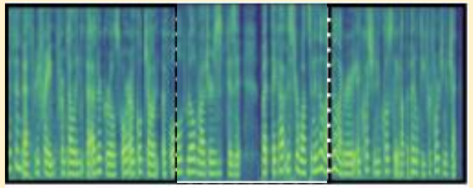
METHODOLOGY - OVERVIEW



Architectural Diagram of proposed P-FLOW model

Methodology: Speech-prompted Text encoder

1 Original Speech

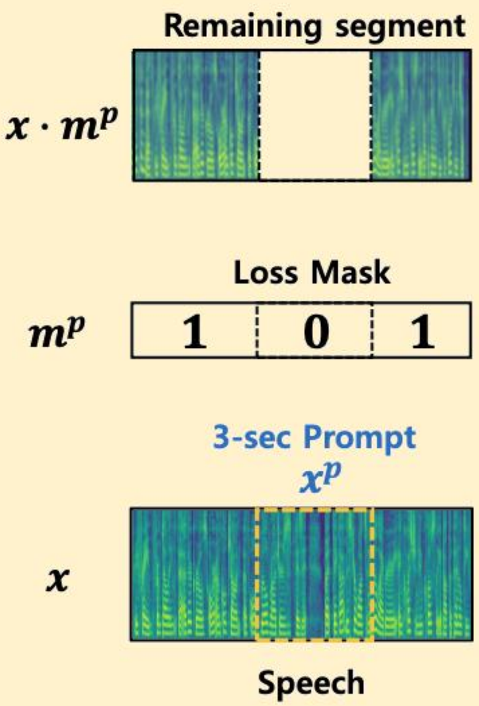


x : mel-spectogram of input speech

c : original text

m^p : Indicator mask on seqeunce x

2

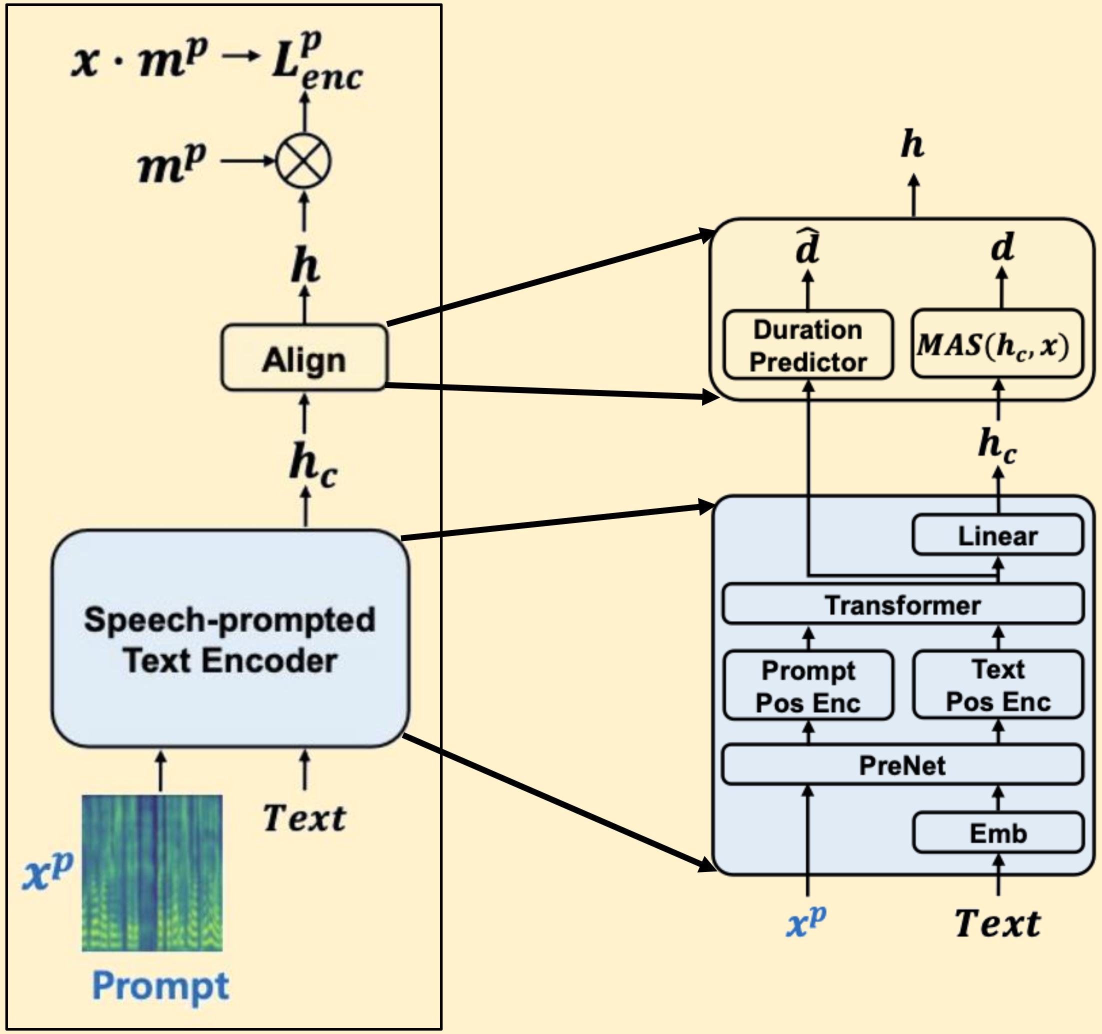


3

4

5

6



Monotonic Alignment search

Input:
h_c = (Y) speech prompted text tokens (N_tokens) and ***X*** mel spectrogram frames (M_frames)

Output:
Monotonic alignment **A*** between text tokens and speech frames

Algorithm:

Compute the first row: $Q_{1,j}$ = Σk=1...M_frames(log *N* (X[k] , µY[1], σY[1]))

for j = 2 to M_frames:
  for i = 2 to min(j, N_tokens):
    Qi,j = max(Qi-1,j-1, Qi,j-1) + log *N* (X[j] , µY[i], σY[i]) ← **DP**
  end for
end for

for j = M_frames-1 to 1; do
  A*[j] = argmax $_{i \in \{ A^*[j+1]-1,\, A^*[j+1] \}}$ **$Q_{i,j}$** ← **Backtrack**
end for

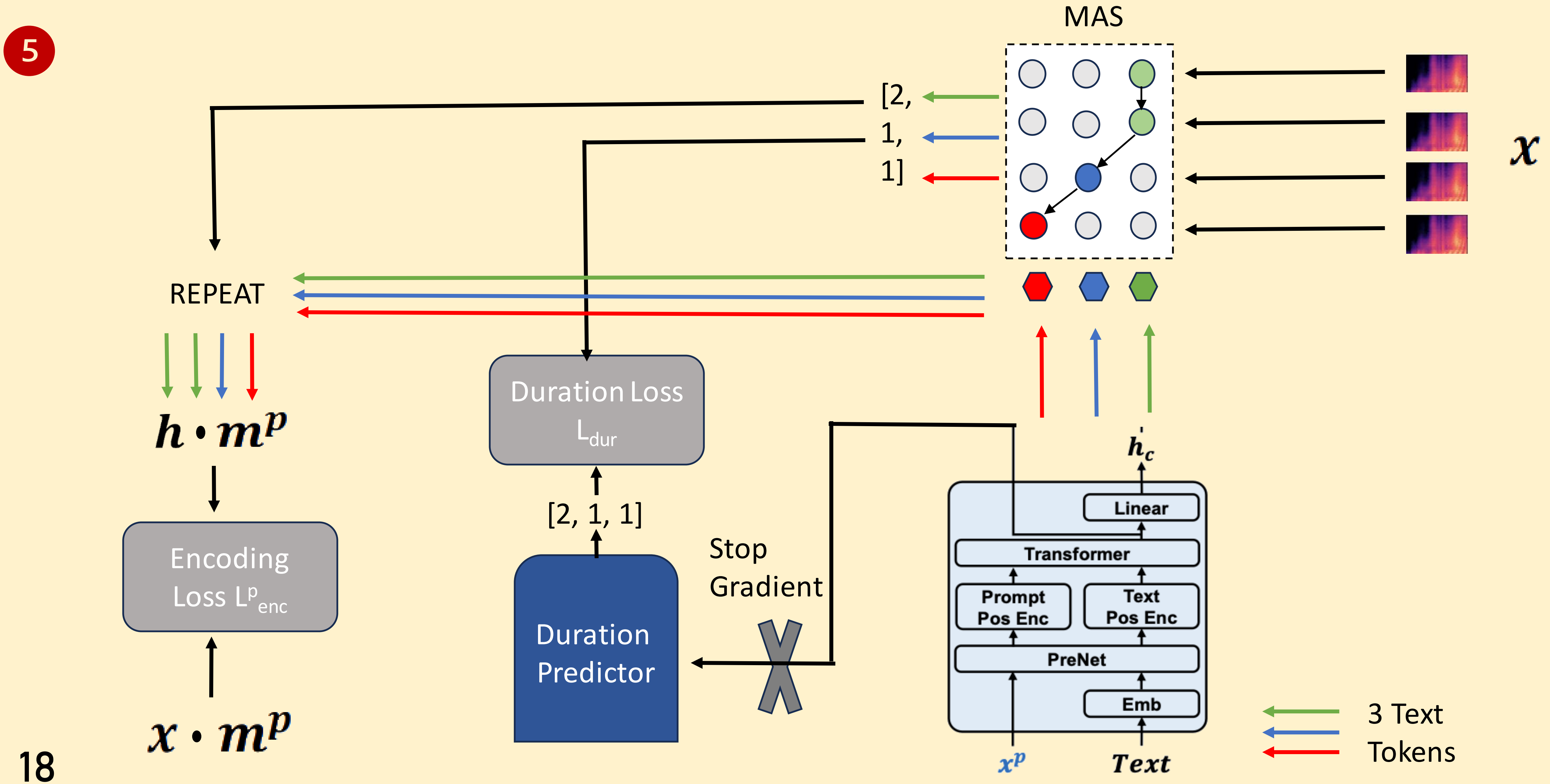
Q matrix
Shape = [N_tokens, M_frames]

-∞	-∞	-∞	-∞
-∞	-∞	-∞	-∞
-∞	-∞	-∞	-∞

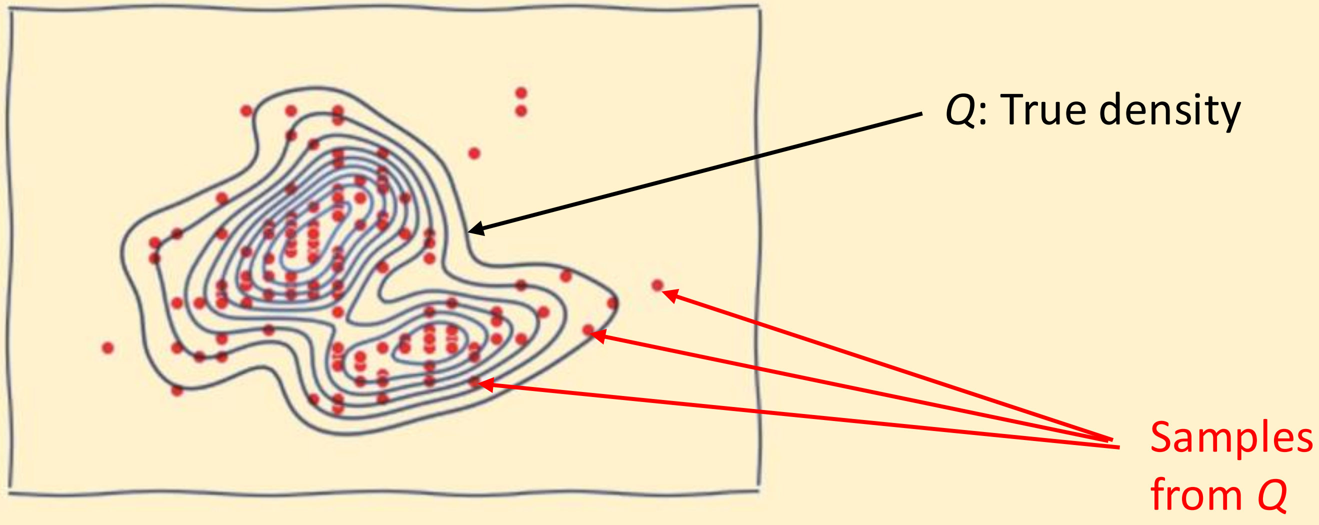
1	1	2	3

A* alignments
Len = [M_frames]

Monotonic Alignment search



Background: Generative Models

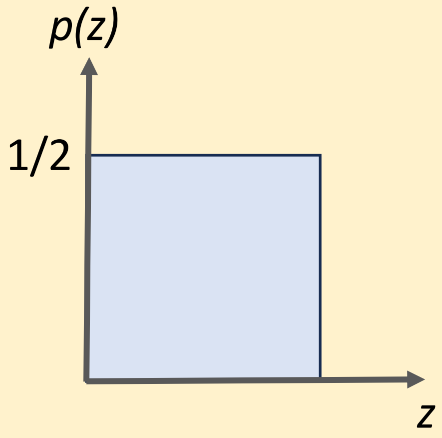


Task 1:
How to estimate the density of samples.
Eg: Consider a distribution Φ that models the images of dogs.
What is $\Pr_\Phi$(X = Golden retriever)?

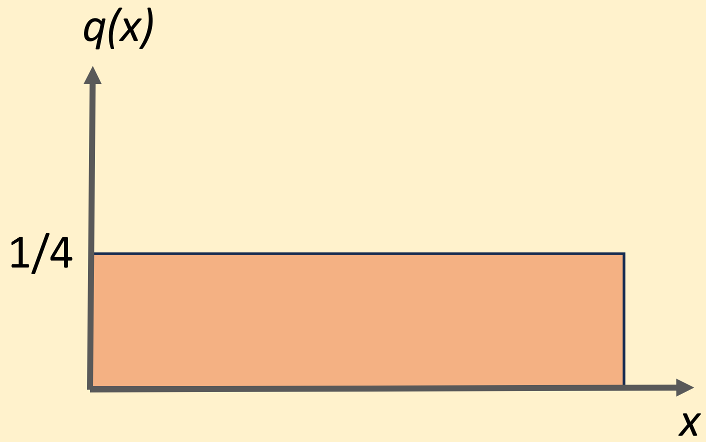
Task 2:
Generating new samples from Q by training on existing samples which are assumed to be generated from the *Q distribution.*

Images borrowed from: Link

Background: Normalizing Flows



Invertible mapping function
T



$$T : Z \to X$$

$$q(x) = p(z)\left|\frac{dz}{dx}\right|$$

Using change of variables

$$q(x) = p(z)\left|\frac{\partial T(z)}{\partial z}\right|^{-1}$$

Background: Normalizing Flows

$$T : Z \rightarrow X$$

$$q(x) = p(z)\left|\frac{\partial T(z)}{\partial z}\right|^{-1}$$

For n samples

$$\prod_{i=1}^{n} q(\mathbf{x}_i) = \prod_{i=1}^{n} p(\mathbf{z}_i)\left|det\left(\nabla_{\mathbf{z}}\mathbf{T}(\mathbf{z}_i)\right)\right|^{-1}$$

Maximize for T

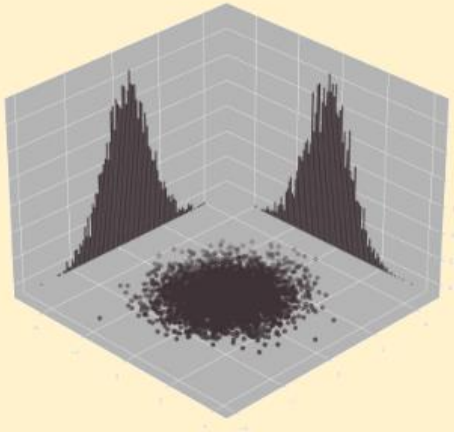
$$\hat{\mathbf{T}} := \arg\max_{\mathbf{T}} \prod_{i=1}^{n} p(\mathbf{z}_i)\left|det\left(\nabla_{\mathbf{z}}\mathbf{T}(\mathbf{z}_i)\right)\right|^{-1}$$

Maximize Log likelihood

$$\hat{\mathbf{T}} := \arg\max_{\mathbf{T}} \sum_{i=1}^{n} \log p(\mathbf{z}_i) - \log\left|det\left(\nabla_{\mathbf{z}}\mathbf{T}(\mathbf{z}_i)\right)\right|$$

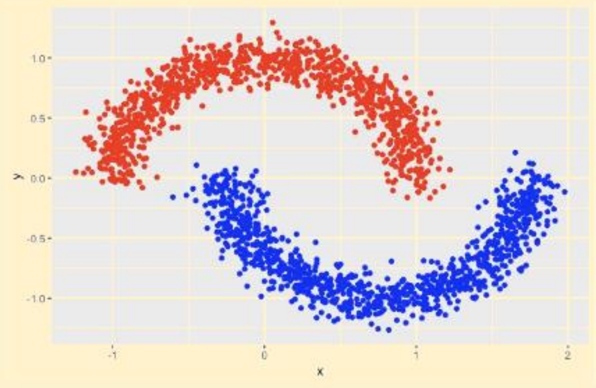
Transformations follow LOTUS: Gives rise to exact log likelihood evaluation

Background: Normalizing Flows



Invertible mapping function
T

Bijective



P(z)

Q(x)

How to estimate density?

$$q(\mathbf{x}) = p(\mathbf{z}) \left| det\left(\nabla_{\mathbf{z}} \mathbf{T}(\mathbf{z}) \right) \right|^{-1}$$

How to sample a new x from Q?

Sample z ~ P(z)
Compute f(z)

More reading: [Link]()

Background: Continuous NFs

- Neural Ordinary Differential Equations (Chen et al., 2018, Link)
- This paper gives a way to backprop through ODE solvers using constant memory requirements.
- Also introduce Continuous time dynamics for Normalizing Flows.
- Theorem (or at least the crux of it): The change in the log probability of a continuous time RV is equal to the negative trace of the Jacobian of the transformation function wrt the RV.

Background: Continuous NFs

Normalizing flow

$$\vec{z}_0 \sim p_0(\vec{z}_0)$$

$$\vec{z}_1 = \vec{f}(\vec{z}_0)$$

$\vec{f}$ invertible and smooth

$$\log p_1(\vec{z}_1) = \log p_0(\vec{z}_0) - \log \left| \det \frac{\partial \vec{f}}{\partial \vec{z}_0} \right|$$

Continuous normalizing flow

$$\vec{z}(t_0) = \vec{z}_0 \sim p_0(\vec{z}_0)$$

$$\frac{d\vec{z}}{dt} = \vec{f}(\vec{z}(t), t)$$

$\vec{f}$ uniformly Lipschitz continuous in z and continuous in t

$$\frac{d \log p(\vec{z}(t))}{dt} = -\,tr\left(\frac{\partial \vec{f}}{\partial \vec{z}}\right)$$

Readers are encouraged to view the slides: Link

Background: Flow Matching in CNFs

General conditional probability path

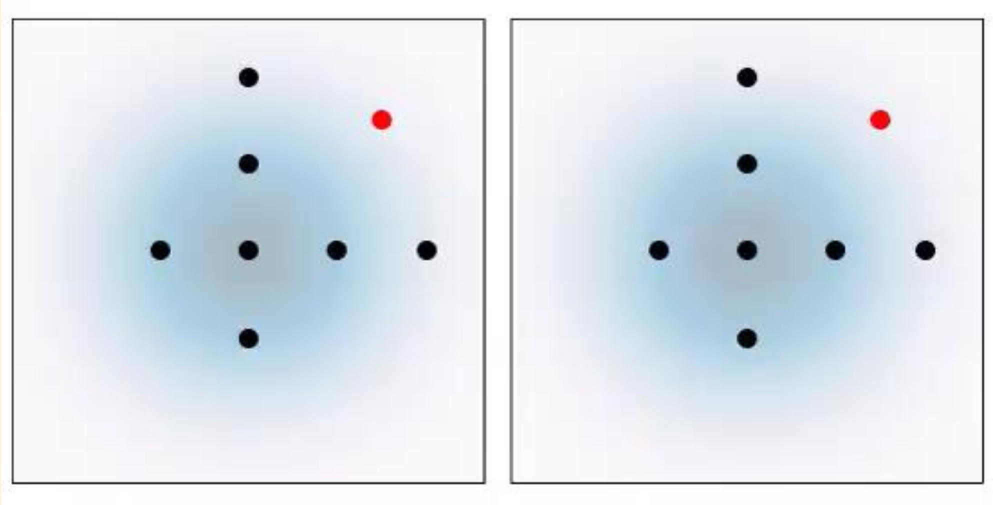
$$p(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mu_t(\mathbf{x}_0), \sigma_t^2(\mathbf{x}_0)I)$$

Training

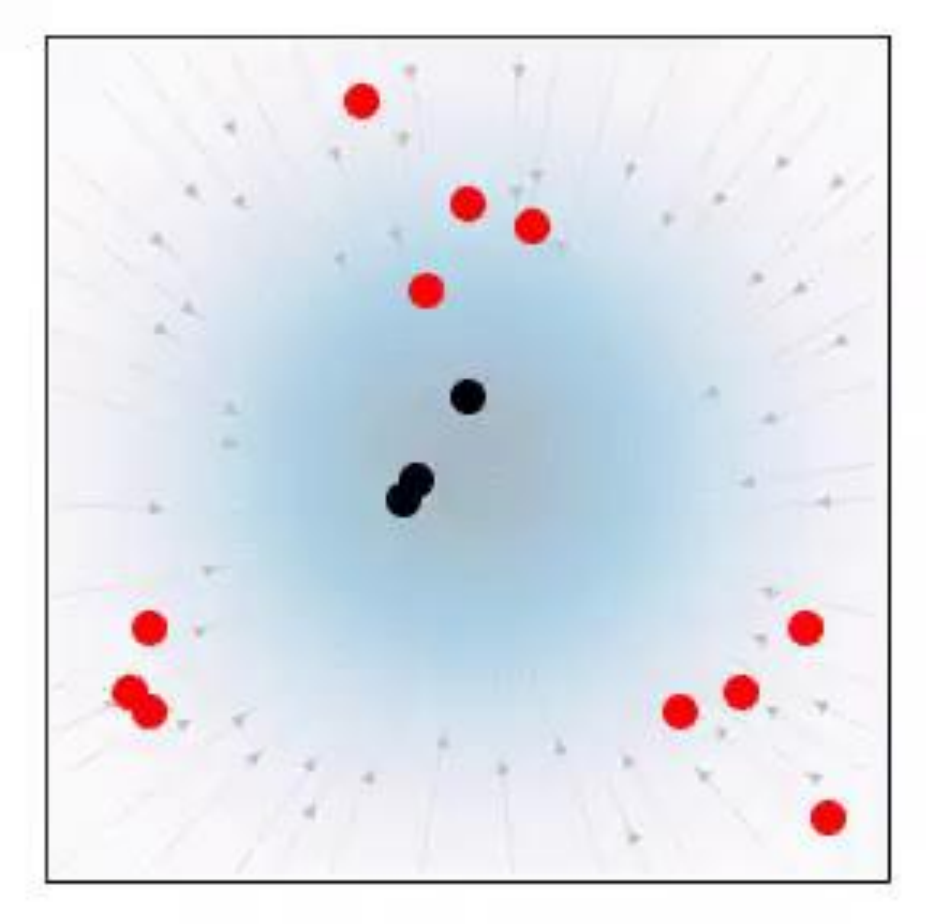
$$\left\| v_\theta(\mathbf{x}_t) - u_t(\mathbf{x}_t \mid \mathbf{x}_0) \right\|^2$$

Sampling

$$\dot{\mathbf{x}}_t = v_\theta(\mathbf{x}_t)$$



In contrast to variance preserving diffusion models (left) CNFs do not overshoot in the final step (right)



Flow Matching directly regresses over the vector fields of probability paths.

GIF borrowed from: [Link]

Background: Flow Matching in CNFs

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \| v_t(x) - u_t(x) \|^2$$

Flow matching objective. Ideal Pt (Path of the flow) and Ut (corresponding vector field) are not known.

$$p_t(x) = \int p_t(x|x_1) q(x_1) dx_1$$

Target probability path can be constructed using simple probability paths.

$$u_t(x) = \int u_t(x|x_1) \frac{p_t(x|x_1) q(x_1)}{p_t(x)} dx_1$$

The marginal vector field generates the marginal probability path.

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \left\| v_t(x) - u_t(x|x_1) \right\|^2$$

CFM objective is equivalent (in expectation) to optimizing the FM objective

Gradients coincide

Background: Flow Matching in CNFs

Algorithm 1: Flow Matching training.

Input : dataset q, noise p
Initialize v^θ
while *not converged* **do**
  $t \sim \mathcal{U}([0,1])$ ▷ sample time
  $x_1 \sim q(x_1)$ ▷ sample data
  $x_0 \sim \boxed{p(x_0)}$ ▷ sample noise
  $x_t = \boxed{\Psi_t(x_0|x_1)}$ ▷ conditional flow
  Gradient step with $\nabla_\theta \| v_t^\theta(x_t) - \dot{x}_t \|^2$
Output: v^θ

Algorithm 2: Flow Matching sampling.

Input : trained model v^θ
$x_0 \sim p(x_0)$ ▷ sample "noise"
Numerically solve ODE $\dot{x}_t = v_t^\theta(x_t)$
Output: x_1

GIF borrowed from: Link

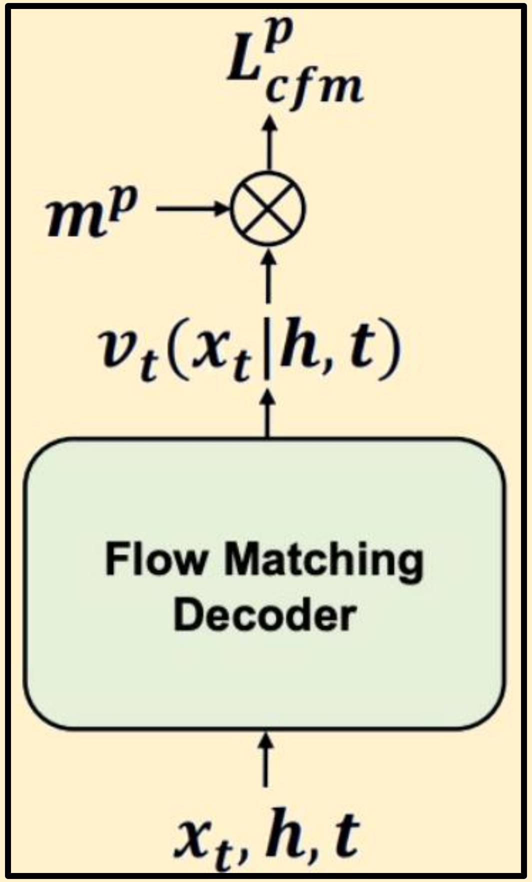
METHODOLOGY : DECODER

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)); \quad \phi_0(x) = x$$

ODE Defining the transformation

$$\phi_{t,x_1}(x) = \sigma_t(x_1)x + \mu_t(x_1)$$

$$\mu_t(x) = tx_1, \ \sigma_t(x) = 1 - (1 - \sigma_{\min})t$$



$$L_{CFM}(\theta) = \mathbb{E}_{t\sim U[0,1], x_1\sim q(x_1), x_0\sim p(x_0)} \| v_t(\phi_{t,x_1}(x_0); \theta) - \frac{d}{dt}\phi_{t,x_1}(x_0) \|^2$$

Methodology : DECODER

$$L_{CFM}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \| v_t(\phi_{t,x_1}(x_0);\theta) - (x_1 - (1-\sigma_{\min})x_0) \|^2$$

Sampling

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

After Guided Sampling:

$$x_{t+\frac{1}{N}} = x_t + \frac{1}{N}(\hat{v}_\theta(x_t, h, t) + \gamma(\hat{v}_\theta(x_t, h, t) - \hat{v}_\theta(x_t, \bar{h}, t))$$

Loss Function of P-Flow

$$L_{enc} = MSE(h, x)$$

$$L^p_{enc} = MSE(h \cdot m^p, x \cdot m^p)$$

L^p_{cfm} : Loss from the Flow Matching Decoder front

L_{dur} : minimise MSE(log(d)) obtained through MAS while training

Overall Training Loss: $$L = L^p_{enc} + L^p_{cfm} + L_{dur}$$

Datasets Details

Dataset Name	Data size	No.of speakers	Languages	Download Link
LibriLight	60,000hrs	7000+	English	https://github.com/facebookresearch/libri-light
Librispeech	982hrs	2484	English	https://www.openslr.org/12
LibriTTS	585hrs	2456	English	https://www.openslr.org/60
VCTK	44hrs	110	English	https://datashare.ed.ac.uk/handle/10283/2651

VALL-E vs P-Flow Methodological Differences

	VALL-E	P-Flow
Speech Representation	Audio Codec Code	Mel-spectograms
Generative Model	Language Model	Flow Matching Generative Model
Training Data	60,000 hours	260 hours
In-context learning	✓	✓
Dataset used	LibriLight	LibriTTS
Evaluation datasets	LibriTTS, LibriSpeech, VCTK	LibriSpeech, VCTK

FAST: 20x times faster than VALL-E

Data-efficient: Less than 0.01x VALL-E's training dataset

Zero-shot: Comparable to VALL-E

Sample Quality, Pronunciation accuracy: P-Flow > VALL-E

Images borrowed from: [Link]

Experiments and ablation study

DATASET Details

LibreTTS dataset: 580hrs of data(2456 speakers) -> 256hrs subset (need 3sec for prompting)

Evaluation on LibriSpeech

Training Configurations	**Important**
Used 1 NVIDIA A100 GPU	Euler steps : 10
Learning rate : 0.0001	Guidance scale: 1
Optimizer : Adam	Preprocessing: G2P model into IPA format
Batch size : 64	Postprocessing: HIfi-GAN mel-spectogram to audio wav file WS : 1024, hop length: 256, 22kHz representation

Hyperparameters of Model for Experiments

Speech-prompted Text Encoder	Phoneme Embedding Dim	192
	PreNet Conv Layers	3
	PreNet Hidden Dim	192
	PreNet Kernel Size	5
	PreNet Dropout	0.5
	Transformer Layers	6
	Transformer Hidden Dim	192
	Transformer Feed-forward Hidden Dim	768
	Transformer Attention Heads	2
	Transformer Dropout	0.1
	Prompt Embedding Dim	192
	Number of Parameters	3.37M
Duration Predictor	Conv Layers	3
	Conv Hidden Dim	256
	LayerNorm Layers	2
	Dropout	0.1
	Number of Parameters	0.36M
Flow Matching Decoder	WaveNet Residual Channel Size	512
	WaveNet Residual Blocks	18
	WaveNet Dilated Layers	3
	WaveNet Dilation Rate	2
	Number of Parameters	40.68M

Evaluation metrics

OBJECTIVE METRICS

1. **Word Error Rate (WER):**
   The number of errors(Insertion, substitution, deletion) divided by the total words
2. **Speaker Embedding Cosine Similarity (SECS):**
   An evaluation metric measuring speaker similarity between generated and original speech.
3. **Inference Latency**

SUBJECTIVE METRICS

1. **Comparative Mean Opinion Score (CMOS):**
   Used for comparing the voice quality of two TTS systems
2. **Comparative Speaker similarity Mean Opinion Score (SMOS):**
   Used for comparing the similarity of waves compared with recording waves

Results and Analysis

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	$\mathbf{0.115 \pm 0.004}$

On LibriSpeech

Model	WER↓	SECS↑
VALL-E	4.3	0.452
P-Flow (proposed)	**2.4**	**0.465**

On VCTK

Dataset	CMOS Î	SMOS Î
LibreSpeech	**0.27** ± 0.10	**0.23** ± 0.13
VCTK	**0.188** ± 0.10	**0.267** ± 0.166

Subjective Metrics:
P-FLOW > VALL-E

Experiments and Ablation study

Model	WER↓	SECS↑
GT (HiFi-GAN)	2.4	0.64
P-Flow (w/o Prompt)	2.9	0.373
P-Flow	**2.6**	**0.544**

P-Flow with and without Prompt
(Importance of Speech promting)

Model	WER↓	SECS↑
P-Flow (Euler method, $N = 10$)	2.6	0.544
P-Flow (Heun's method, $N = 4$)	2.6	**0.552**
P-Flow (Midpoint method, $N = 4$)	2.7	0.540

Different ODE Sampling methods

Model	N	MOS↑	SECS	Inference Latency(s)↓
P-Flow	1	3.55 ± 0.16	0.420	0.028 ± 0.004
	2	3.71 ± 0.12	0.522	0.037 ± 0.004
	5	4.01 ± 0.10	0.549	0.067 ± 0.004
	10	4.08 ± 0.10	0.544	0.115 ± 0.004
	20	4.14 ± 0.10	0.540	0.210 ± 0.005

Euler steps and Accoustic quality through Mean Opinion score(MOS)

Effect of variations in
1. guidance scale γ
2. Euler steps N

Model	γ	N	WER↓	SECS↑	Inference Latency(s)↓
P-Flow (Default)	1	10	2.6	0.544	0.115 ± 0.004
P-Flow	0	10	3.7	0.492	0.115 ± 0.004
P-Flow	2	10	2.6	0.546	0.115 ± 0.004
P-Flow	1	1	2.7	0.420	0.028 ± 0.004
P-Flow	1	2	2.9	0.522	0.037 ± 0.004
P-Flow	1	5	2.6	0.549	0.067 ± 0.004
P-Flow	1	20	2.7	0.540	0.210 ± 0.005

P-Flow Demo

Generated Audio

Ground Truth

3-sec Reference

P-Flow

VALL-E

They moved thereafter cautiously about the hut, groping before and about them to find something to show that Warrenton had fulfilled his mission.

P-Flow Demo

Ground Truth

3-sec Reference

Generated Audio

P-Flow

VALL-E

We have made a couple of Albums

P-Flow in Action

P-Flow in Action

- Paper: [***SCALING NVIDIA'S MULTI-SPEAKER MULTI-LINGUAL TTS SYSTEMS WITH ZERO-SHOT TTS TO INDIC LANGUAGES***]
- P-flow implementation secured 1s rank in MMITS-VC 2024 Challenge for Zero shot TTS track
- MMITS-VC : **Multi-speaker, Multi-lingual Indic TTS** with **VOICE CLONING**
- **Organized as part of** ICASSP's Signal Processing Grand Challenge 2024

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

Speaker Similarity

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

RESULTS

Conclusion

- P-flow paper presents three main components of P-Flow architecture:
  - A conditional flow matching decoder for faster sampling
  - A speech prompted text encoder to better speech prompting
  - A MAS algorithm minimizes distance between speech frames and text representations.
- P-flow avoids probability paths which lead to overshooting as transformations reach the target distribution, hence leads to better convergence during sampling.
- Establishes a challenge to data-hungry LMs in recent trends for need of large data
- Other notable works which use flow matching for TTS:
  - Link: Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale
  - Link: Audiobox: Unified Audio Generation with Natural Language Prompts
  - Link: GENERATIVE PRE-TRAINING FOR SPEECH WITH FLOW MATCHING
  - Link: REFLOW-TTS: A RECTIFIED FLOW MODEL FOR HIGH-FIDELITY TEXT-TO-SPEECH

Future Directions

1. Larger dataset is not required for achieving comparable naturalness and speaker adaptation from P-Flow, trying it out for accented TTS, multi-lingual TTS, where various variety of speech is possible through fewer samples
2. Low resource language speeches could be experimented with p-flow using transfer learning.
3. Zero-shot capabilities of duration predictor remain limited
4. High-quality zero-shot TTS might cause social impact, so steps to detect synthetic audio is required

Learnings

Through the P-Flow paper, we learnt the details of following concepts

1. Current SOTA models and their workflow in the domain of zero-shot TTS
2. Usage of mel-spectograms in Speech as representations
3. Concepts of Flow Matching and their effectiveness
4. Metrics evaluated for the problems of TTS, subjective and objective
5. Need of efficient TTS models and their usecases along with social impact

References

- Kim, J., Kim, S., Kong, J. and Yoon, S., 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems.* [GLOW-TTS]
- Kharitonov, E., Vincent, D., Borsos, Z., Marinier, R., Girgin, S., Pietquin, O., Sharifi, M., Tagliasacchi, M. and Zeghidour, N., 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540.* [SPEAR-TTS]
- Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M. and Le, M., 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747.* [Flow Matching]
- Kim, S., Shih, K.J., Badlani, R., Santos, J.F., Bakhturina, E., Desta, M.T., Valle, R., Yoon, S. and Catanzaro, B., 2023, November. P-Flow: A Fast and Data-Efficient Zero-Shot TTS through Speech Prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*. [P-Flow]
- 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. **[VALL-E]**

References

- Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8599–8608. PMLR, 2021. [Grad-TTS]
- Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019. [LibriTTS]
- Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020. [HIFI-GAN]
- Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR, 2022. [YourTTS]
- Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022. [EnCodec]

THANK YOU SLIDE

THANK
YOU FOR
YOUR TIME

Q&A

References and Tutorials

- Normalizing flows :
- Conditional Flow matching: