End-to-End Text-to-Speech using Latent Duration based on VQ-VAE

Yusuke Yasuda, Xin Wang, Junichi Yamagishi
National institute of informatics, Japan
The graduate university for advanced studies, SOKENDAI, Japan

ICASSP 2021

Background

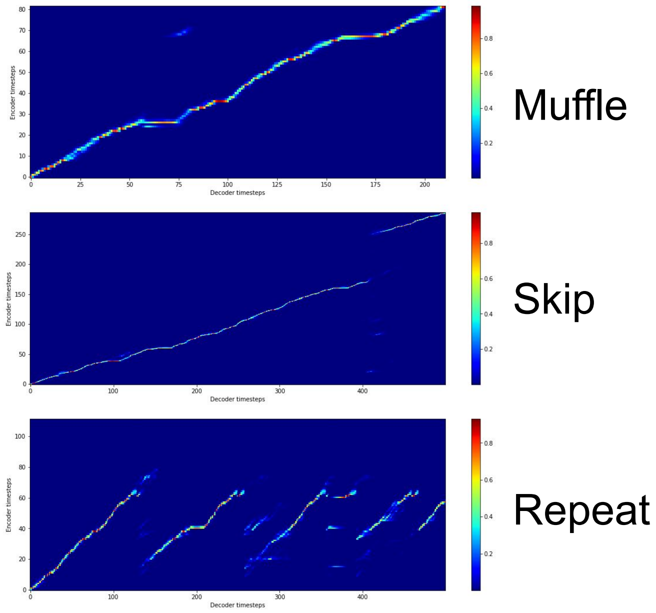
End-to-end TTS is a method of converting text to speech directly in single model.

☆Issues

1) Alignments predicted from soft-attention tend to be unstable.

☆Solutions

1) Construct monotonic alignments from phoneme duration.



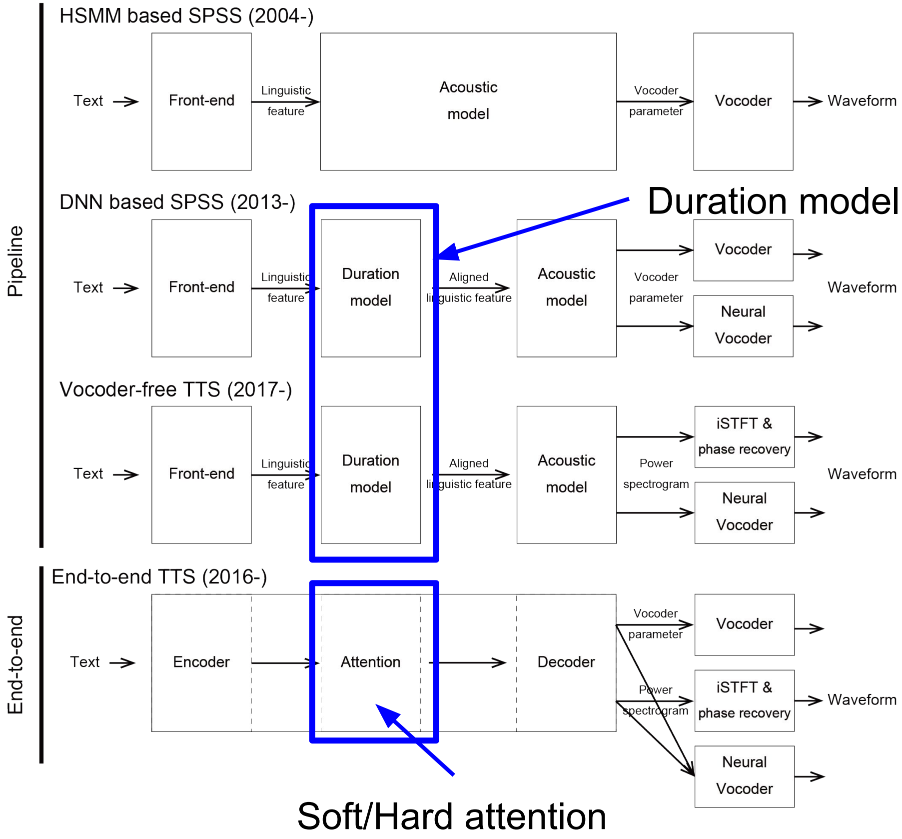
Errors from soft-attention.

Our approach: simplification of duration based TTS

Method	Teacher-Student	Training phases	Aligner	Aligner is external	Duration form	Latent duration
FastSpeech	✓	3	Soft-attention	✓	continuous	
DurIAn		3	HMM?	✓	continuous	
FastSpeech2		3	HMM-GMM	✓	continuous	
AlignTTS		4	SSNT-like MDN	✓	discrete	
JDI-T	✓	1	Soft-attention + CTC		continuous	
Glow-TTS		1	MAS		continuous	
Non-Attentive Tacotron		2	HMM	✓	continuous	?
VQ-VAE		1	CTC		discrete	✓

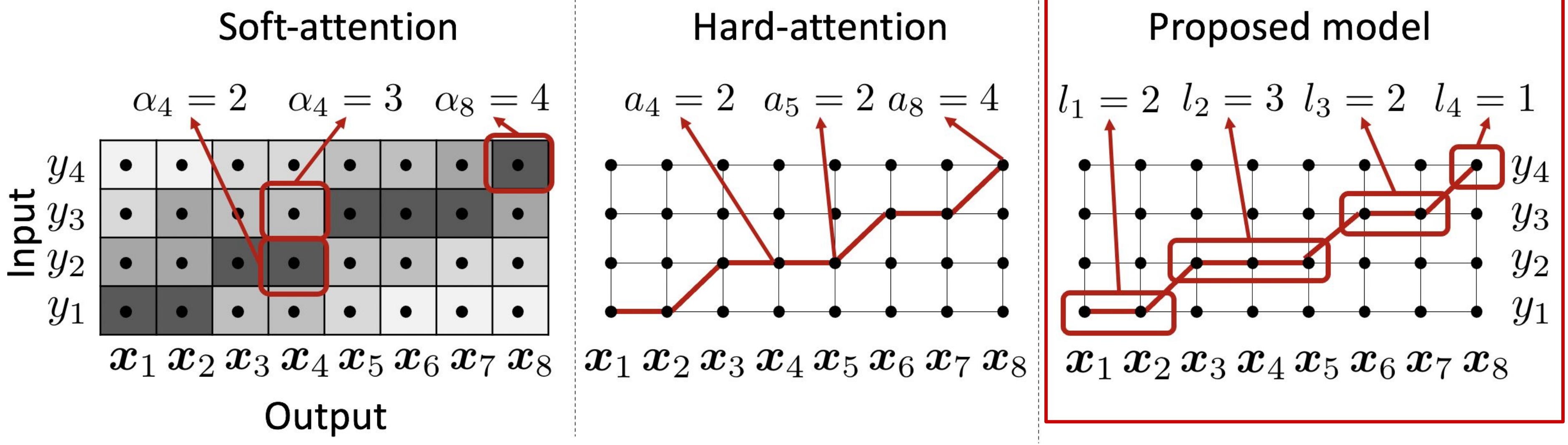
- All modules are jointly trainable
  - Single training phase
- Discrete duration
  - Conform with forced aligner and upsampling
  - No length reguralizer or ceiling of duration
- Duration is modeled as latent variables
  - Simple training criterion based on VAE

Alignment methods in TTS frameworks



- Pipeline TTS
  - Consists of multiple models to convert texts into speech.
  - Each model is dedicated to a single function.
  - Duration model forms alignments between source texts and target speech.
- End-to-end TTS
  - Consists of a single model to convert texts into speech
  - Attention mechanism forms alignments between source texts and target speech.

Design of a latent alignments based on duration

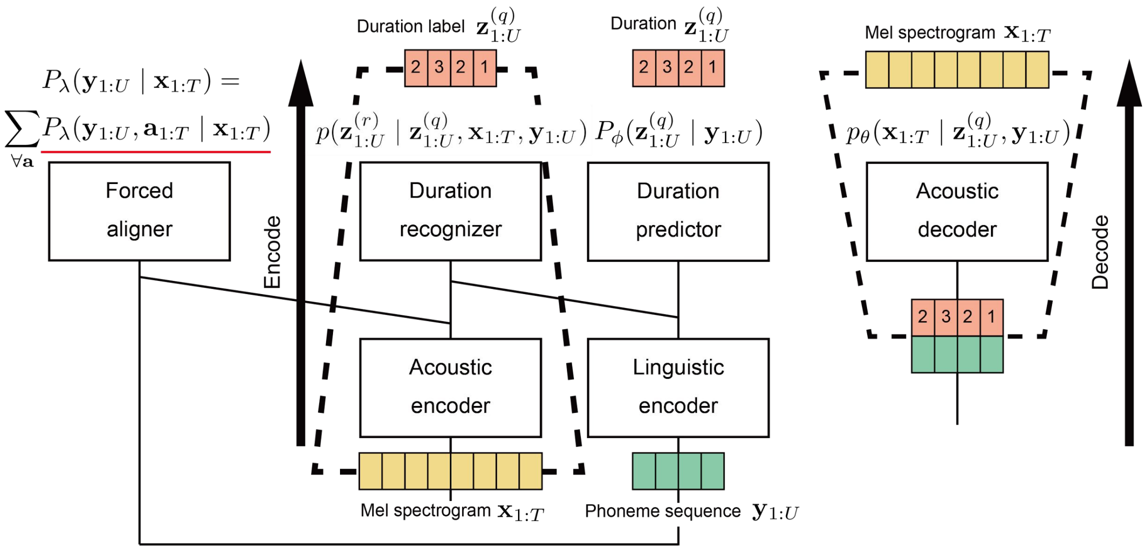


- Using phoneme duration as a latent variable
- Let the phoneme duration length be a discrete variable (the number of frames of acoustic features)

Modeling latent duration with variational autoencoder (VAE)

$$\mathbf{z}^{(q)}_{1:U} = \arg\max_{\mathbf{a}_{1:T}} \underline{P_\lambda(\mathbf{y}_{1:U}, \mathbf{a}_{1:T} \mid \mathbf{x}_{1:T})}$$

Approximate posterior $\sim Q_\psi(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{x}_{1:T}, \mathbf{y}_{1:U})$



- Introduce phoneme duration as discrete latent variables ($\mathbf{z}^{(q)}_{1:U}, \mathbf{z}^{(r)}_{1:U}$)
- Sample phoneme duration from forced aligner
- Define approximate posterior with samples of phoneme duration
- Use duration predictor as prior for VAE
- Use duration recognizer as VAE's encoder
- Use TTS decoder as VAE's decoder

Objective function of conditional VQ-VAE

$$\log p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:U})$$

TTS as a probabilistic model

$$= \log \sum_{\forall \mathbf{z}^{(q)}_{1:U}} \int_{\mathbf{z}^{(r)}_{1:U}} p(\mathbf{x}_{1:T}, \mathbf{z}^{(q)}_{1:U}, \mathbf{z}^{(r)}_{1:U} \mid \mathbf{y}_{1:U}) d\mathbf{z}^{(r)}_{1:U} \tag{7.1}$$

TTS decoder

$$\geq \mathbb{E}_{Q_\lambda(\mathbf{z}^{(q)}_{1:U})} \underbrace{[\log p_\theta(\mathbf{x}_{1:T} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{y}_{1:U})]}_{\text{Decoder}} \tag{7.2}$$

Duration predictor (prior)

$$- \mathrm{KL}[Q_\lambda(\mathbf{z}^{(q)}_{1:U} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:U}) \| \underbrace{P_\phi(\mathbf{z}^{(q)}_{1:U} \mid \mathbf{y}_{1:U})]}_{\text{Prior}} \tag{7.3}$$

Duration recognizer (vector quantization)

$$- \mathbb{E}_{Q_\lambda(\mathbf{z}^{(q)}_{1:U})} \left\{ \mathrm{KL}[Q_\psi(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{x}_{1:T}, \mathbf{y}_{1:U}) \| \underbrace{p(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U})}\ ] \right\}. \tag{7.4}$$

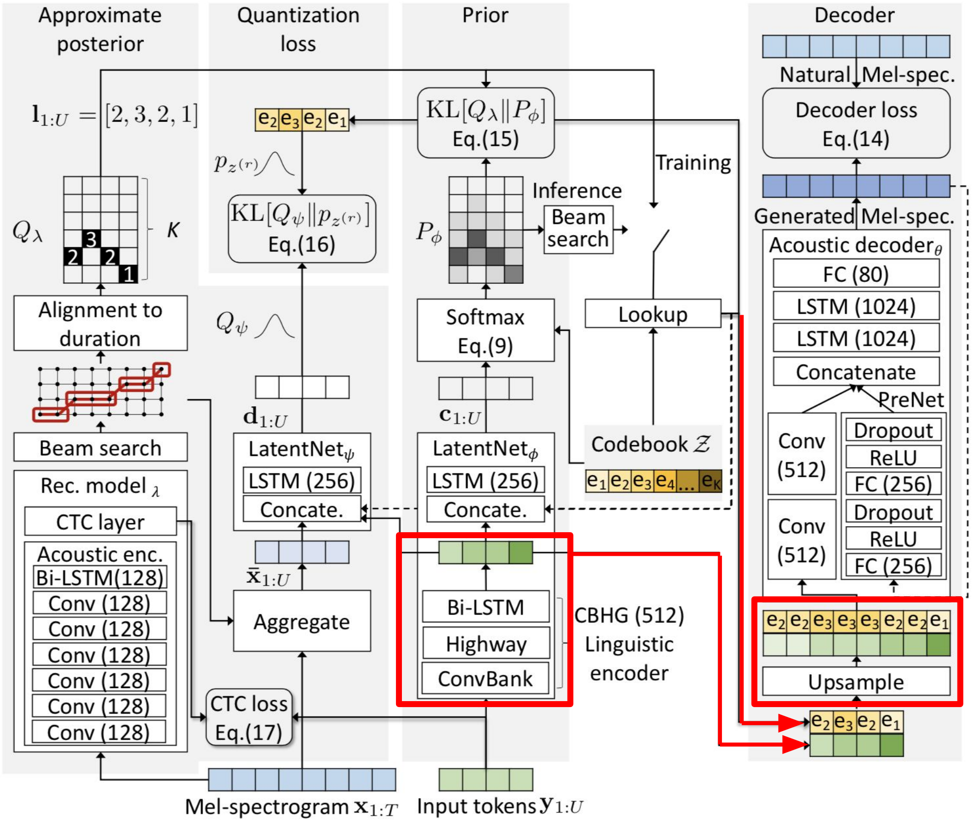
$$+ \gamma \log \sum_{\forall \mathbf{a}_{1:T}} P_\lambda(\mathbf{y}_{1:U}, \mathbf{a}_{1:T} \mid \mathbf{x}_{1:T}), \tag{7.17}$$

Forced aligner

Duration samples from forced aligner (approximate posterior)

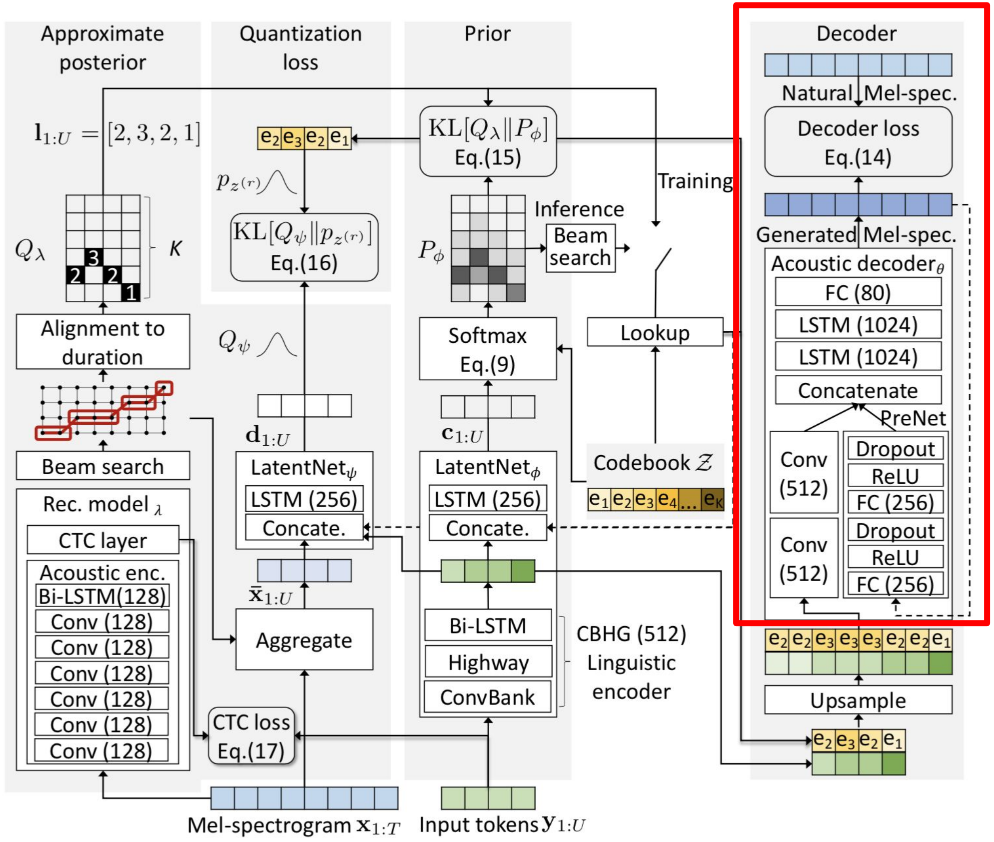
- TTS can be model with the conditional probability (x: speech, y: texts).
- The marginal probability can be approximated with ELBO.
- Modules in TTS can be incorporated to VQ-VAE:
  - Prior: duration predictor
  - VQ: duration recognizer
  - Approximate posterior: duration samples
  - Decoder: TTS decoder
  - Sampler: forced aligner

An architecture of the proposed method (1)



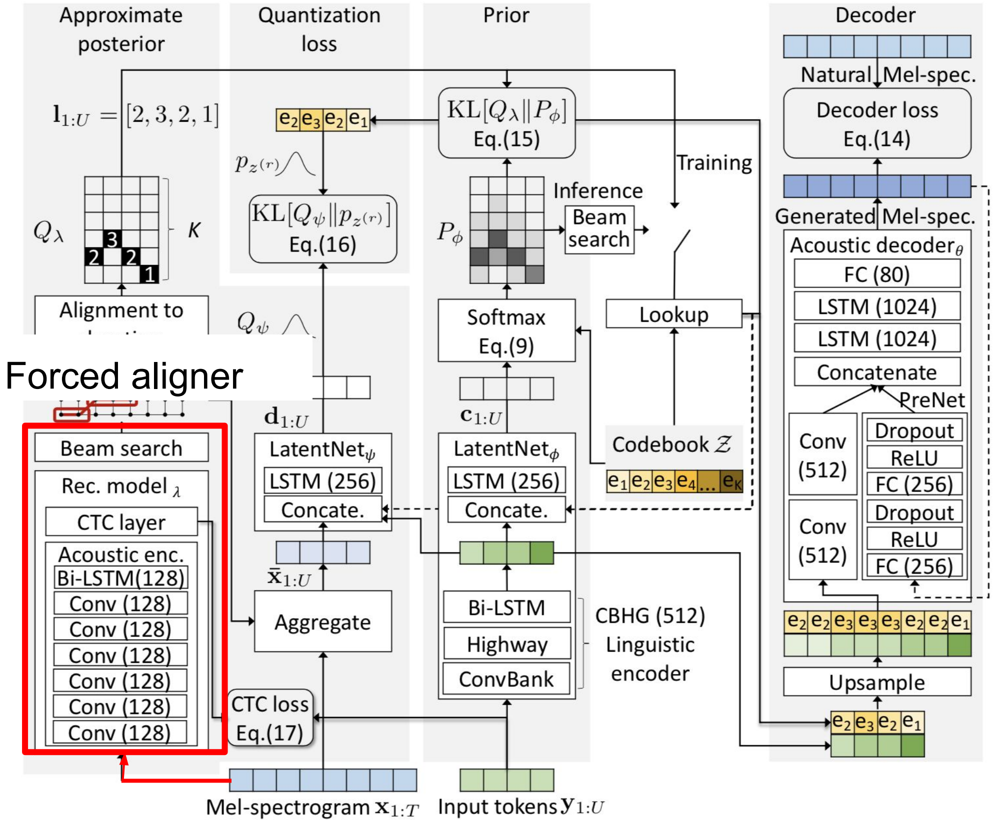
- Common behaviors
  a. Upsample linguistic features based on phoneme duration
  b. Decoder decodes the upsampled linguistic features into acoustic features.

An architecture of the proposed method (2)



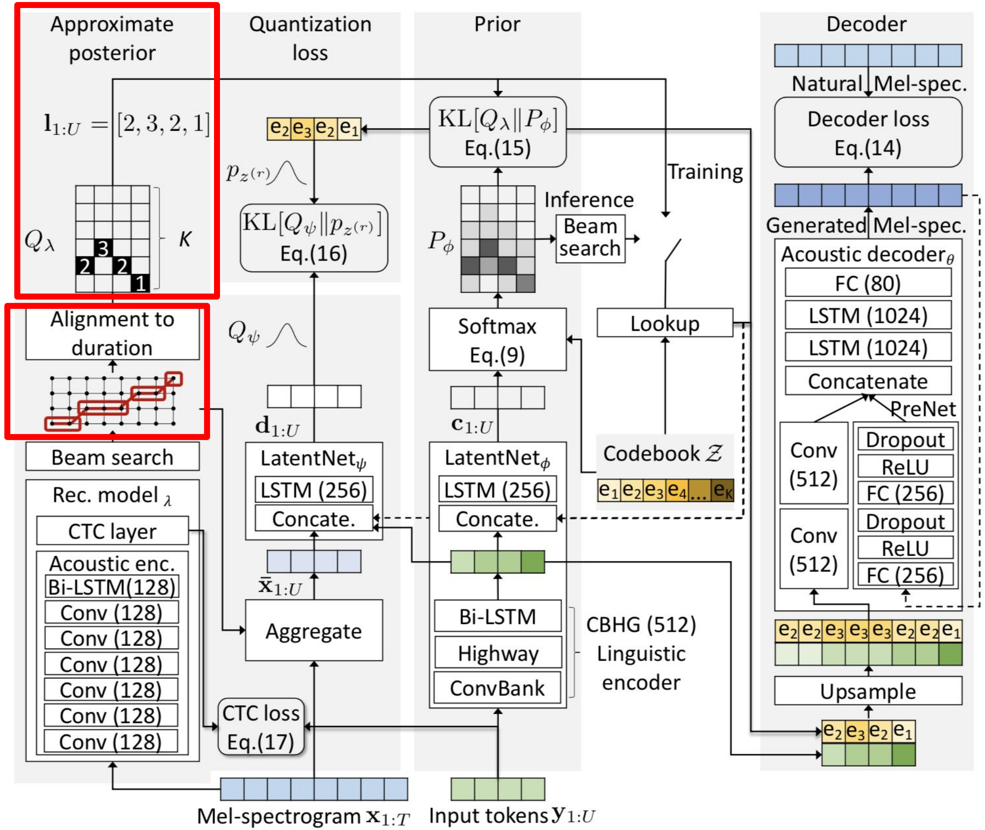
- Common behaviors
  a. Upsample linguistic features based on phoneme duration
  b. Decoder decodes the upsampled linguistic features into acoustic features.

An architecture of the proposed method (3)



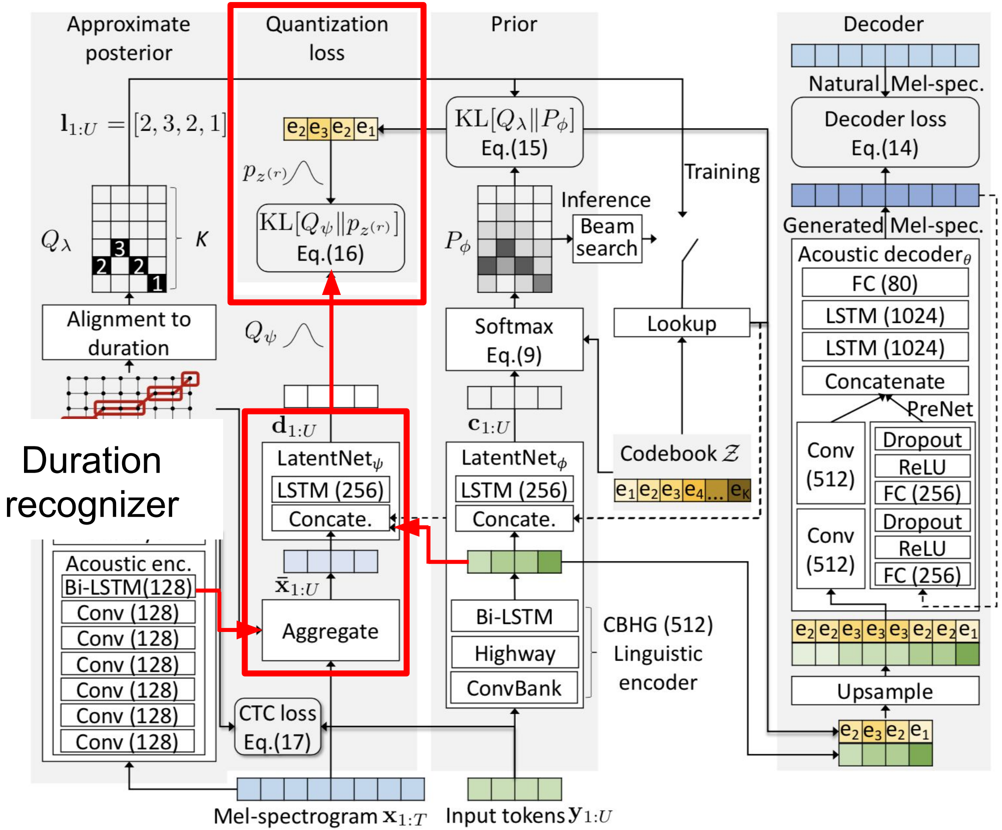
- Training phase
  a. Sample alignments from forced aligener.
  b. Convert alignments into phoneme duration. The duration samples defines approximate posterior.
  c. Duration recognizer predicts distributions by encoding linguistic and acoustic features. Duration codebook is optimized by minimizing KLD between the distribution and codebook.
  d. The samples are used to train duration recognizer and predictor.
  e. Duration predictor predicts distribution about duration from linguistic features. Minimizing
  f. KLD between the distribution and the approximate posterior optimizes the duration predictor.

An architecture of the proposed method (4)



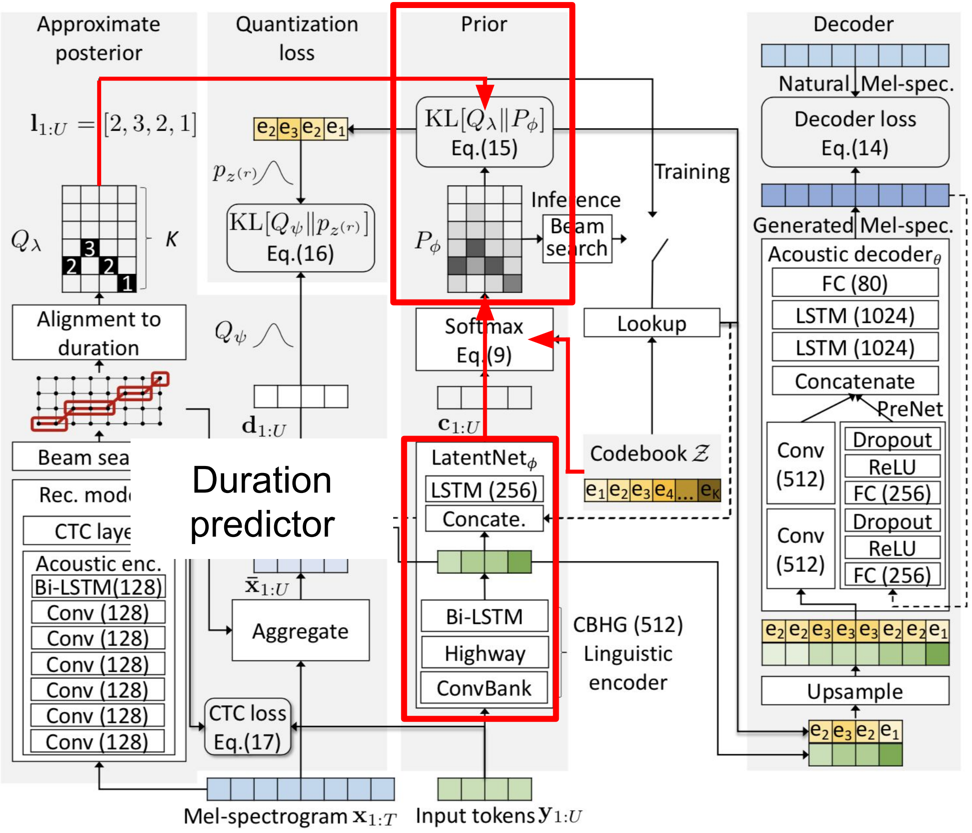
- Training phase
    a. Sample alignments from forced aligener.
    b. Convert alignments into phoneme duration. The duration samples defines approximate posterior.
    c. Duration recognizer predicts distributions by encoding linguistic and acoustic features. Duration codebook is optimized by minimizing KLD between the distribution and codebook.
    d. The samples are used to train duration recognizer and predictor.
    e. Duration predictor predicts distribution about duration from linguistic features. Minimizing
    f. KLD between the distribution and the approximate posterior optimizes the duration predictor.

An architecture of the proposed method (5)



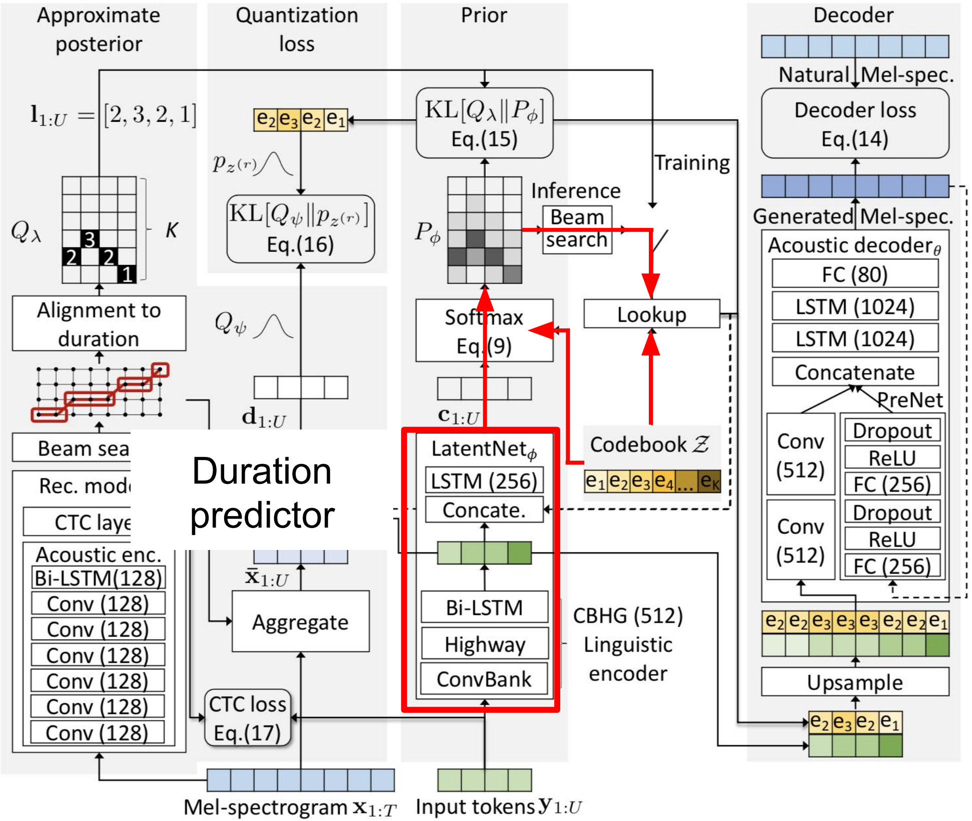
- Training phase
  a. Sample alignments from forced aligener.
  b. Convert alignments into phoneme duration. The duration samples defines approximate posterior.
  c. Duration recognizer predicts distributions by encoding linguistic and acoustic features. Duration codebook is optimized by minimizing KLD between the distribution and codebook.
  d. The samples are used to train duration recognizer and predictor.
  e. Duration predictor predicts distribution about duration from linguistic features. Minimizing
  f. KLD between the distribution and the approximate posterior optimizes the duration predictor.

An architecture of the proposed method (6)



- Training phase
  a. Sample alignments from forced aligener.
  b. Convert alignments into phoneme duration. The duration samples defines approximate posterior.
  c. Duration recognizer predicts distributions by encoding linguistic and acoustic features. Duration codebook is optimized by minimizing KLD between the distribution and codebook.
  d. Duration predictor predicts distribution about duration from linguistic features. Minimizing KLD between the distribution and the approximate posterior optimizes the duration predictor.

An architecture of the proposed method (7)



- Prediction phase
  a. Sample phoneme duration from duration predictor with beam search.

Experiment

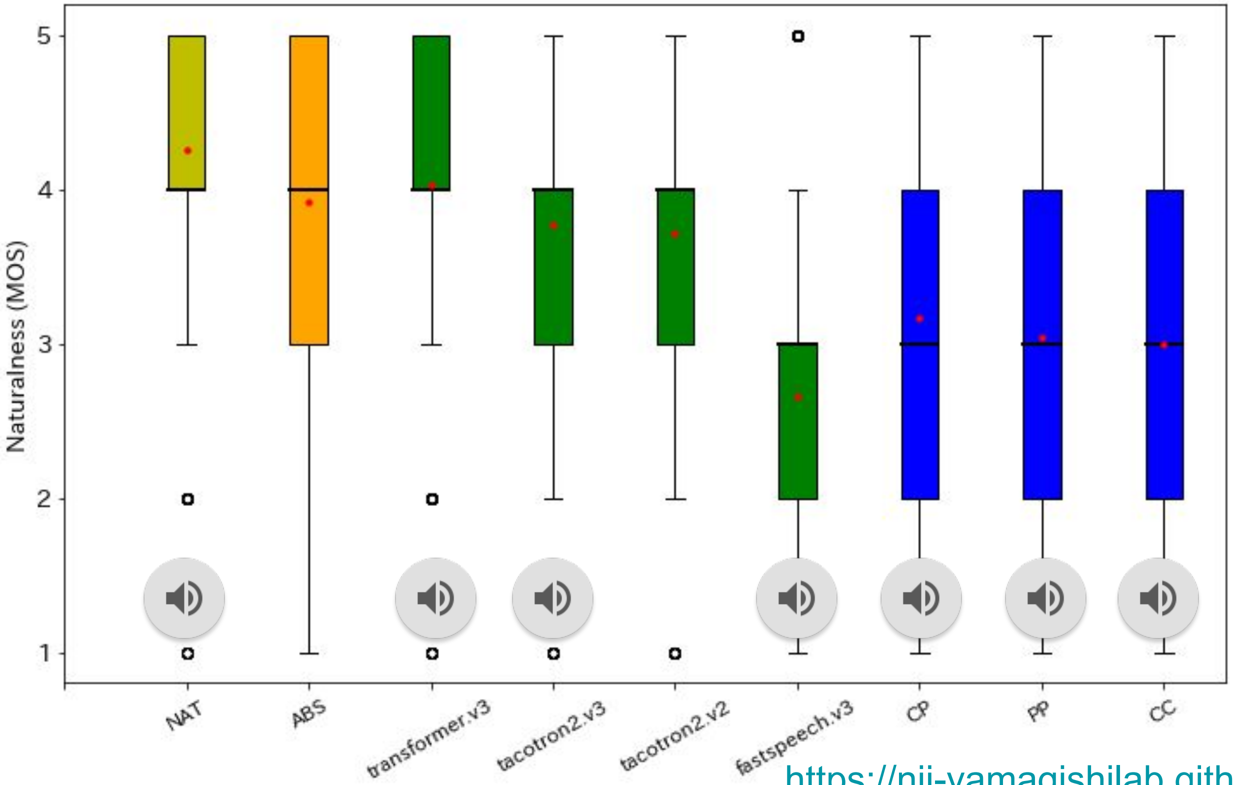
- Corpus: LJSpeech (English, 1 female speaker, 24h)
- Proposed systems
  - CP (Character inputs for TTS, phoneme inputs for forced aligner)
  - PP (Phoneme inputs)
  - CC (Character inputs)
- Baselines:
  - Transformer TTS
  - Tacotron2 (v2, v3)
  - FastSpeech
  - ABS
  - Natural

Major end-to-end TTS methods

A major TTS method using duration model

- Evaluation
  - Listening test about naturalness (5-grade MOS)
  - 200 listeners

Experimental results



- Naturalness:
  - Natural speech
  - > End-to-end TTS
    - Transformer
    - Tacotron v3, v2
  - > Proposed systems
    - CP, PP, CC
  - > TTS using duration model
    - FastSpeech
- Proposed systems
  - CP (character for TTS, phoneme for CTC) was the best

https://nii-yamagishilab.github.io/sample-tts-latent-duration/

Pros & Cons of the proposed method

- Pros
  - Single training phase.
  - Simple training criterion.

Method	Teacher-Student	Training phases	Aligner	Aligner is external	Duration form	Latent duration
FastSpeech	✓	3	Soft-attention	✓	continuous	
DurIAn		3	HMM?	✓	continuous	
FastSpeech2		3	HMM-GMM	✓	continuous	
AlignTTS		4	SSNT-like MDN	✓	discrete	
JDI-T	✓	1	Soft-attention + CTC		continuous	
Glow-TTS		1	MAS		continuous	
Non-Attentive Tacotron		2	HMM	✓	continuous	?
VQ-VAE		1	CTC		discrete	✓

- Cons
  - Sensitive to design of linguistic feature labels
    - Absence of pauses
    - Symbols which is not straightforwardly related to duration
  - Duration sampled from forced aligner (CTC) is not good enough.
    - CTC assumes conditional independence across time steps.
    - It may not be suitable for segmentation.