VAENAR-TTS: Variational Auto-Encoder based Non-AutoRegressive Text-to-Speech Synthesis

Hui Lu[1,2], *Zhiyong Wu*[1,3], *Xixin Wu*[4], *Xu Li*[1], *Shiyin Kang*[5], *Xunying Liu*[1], *Helen Meng*[1,2,3]

[1]Dept. of Systems Engineering & Engineering Management, Chinese University of Hong Kong
[2]Centre for Perceptual and Interactive Intelligence, CUHK
[3]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[4]Department of Engineering, University of Cambridge, UK
[5]Huya Inc., Guangzhou, China

`{luhui, zywu, wuxx, xuli, xyliu, hmmeng}@se.cuhk.edu.hk, kangshiyin@huya.com`

Content

01 Related Work

02 Motivations

03 Model Formalization

04 Model Architecture

05 Experiments

06 Conclusion

Related Work

Overview

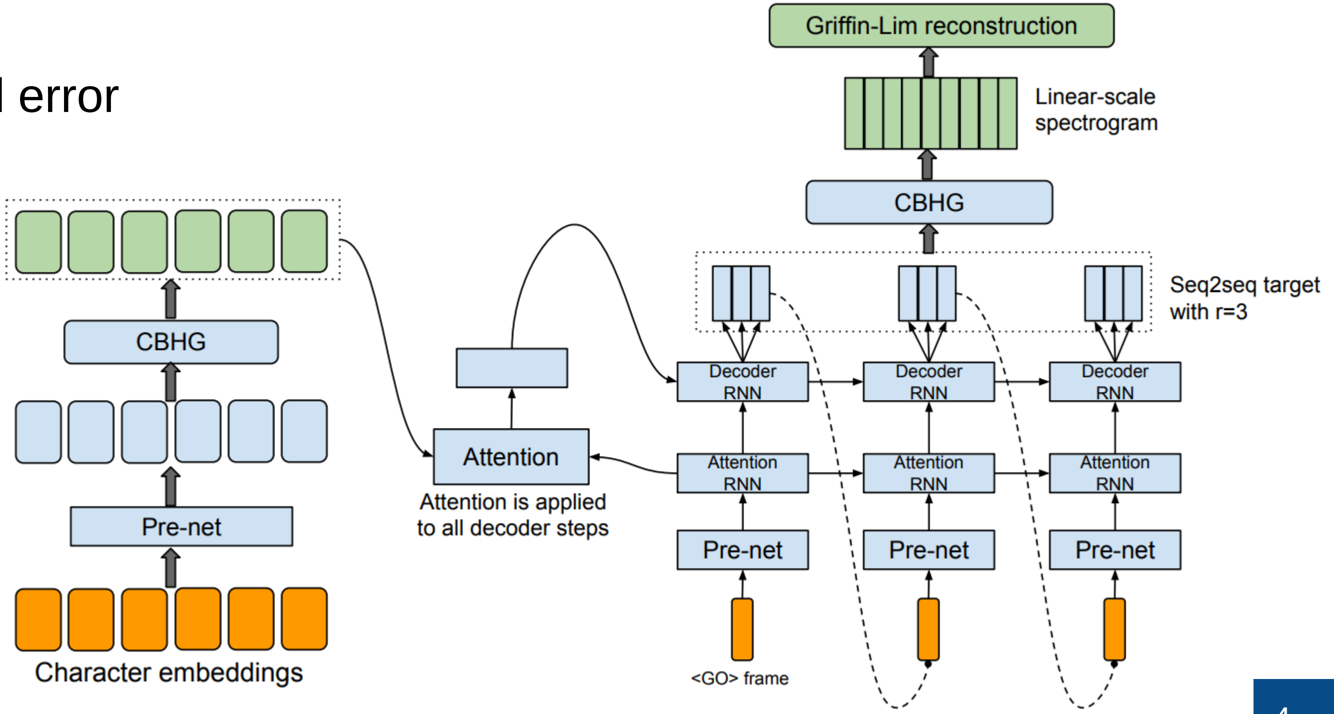
- Autoregressive TTS models
  - ✓ Tacotron, Tacotron2, Transformer-TTS, DeepVoice-3
- Non-autoregressive TTS models
  - ✓ Glow-TTS, BVAE-TTS, VARA-TTS, Fastspeech, Fastspeech-2, Flow-TTS, ParaNet

[1] Y. Wang, et al. "Tacotron: Towards end-to-end speech synthesis," Interspeech, 2017.
[2] J. Shen, et al. "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," ICASSP, 2018.
[3] N. Li, et al. "Neural speech synthesis with transformer network," AAAI, 2019.
[4] W. Ping, et al. "Deep voice 3: 2000-speaker neural text-to-speech," ICLR, 2018.
[7] Y. Lee, et al. "Bidirectional variational inference for non-autoregressive text-to-speech," ICLR, 2021.
[8] Y. Ren, et al. "Fastspeech: Fast, robust and controllable text to speech," NeurIPS, 2019.
[9] Y. Ren, et al. "Fastspeech 2: Fast and high-quality end-to-end text to speech," CoRR, abs/2006.04558, 2020.
[10] J. Kim, et al. "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," NeurIPS, 2020.
[11] K. Peng, et al. "Non-autoregressive neural text-to-speech," ICML, 2020.
[12] C. Miao, et al. "Flow-TTS: A non-autoregressive network for text to speech based on flow," ICASSP, 2020.

Related Work

AR TTS models

- Autoregressive TTS models
  - ✓ Sequence-to-sequence model with attention mechanism
  - ✓ The decoding process is autoregressive
    - ➢ No need of explicit duration modeling
    - ➢ Time-consuming
    - ➢ May be vulnerable to accumulated error



Related Work

From AR to NAR

- From AR to NAR, a core issue is
  - ✓ How to align the phoneme/character-level linguistic feature into the frame-level
  - ✓ Duration model is required in NAR based TTS models
    - Phoneme-level duration
    - Utterance-level duration

Related Work

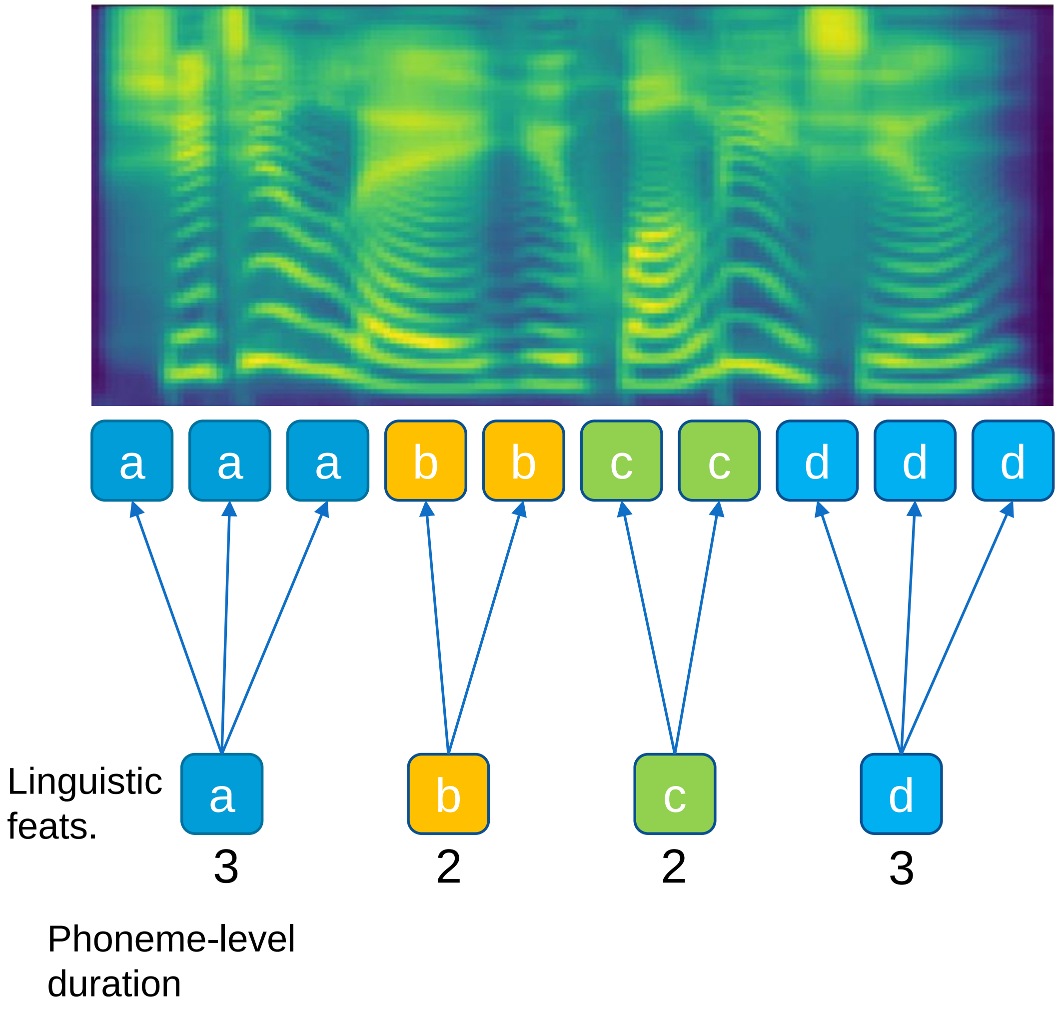
NAR TTS models

- Non-autoregressive TTS models
  - ✓ Phoneme-level duration-based models
    - ➢ Expand the linguistic feature into the frame-level according to durations
    - ➢ Map the frame-level linguistic feature to the spectrogram
    - ➢ Models: FastSpeech, FastSpeech-2, Glow-TTS, BVAE-TTS
  - ✓ Utterance-level duration-based models
    - ➢ Create a spectrogram placeholder with the utterance-level duration
    - ➢ Align the linguistic feature onto the placeholder
    - ➢ Map the aligned linguistic feature to the spectrogram
    - ➢ Models: Flow-TTS, ParaNet, VARA-TTS

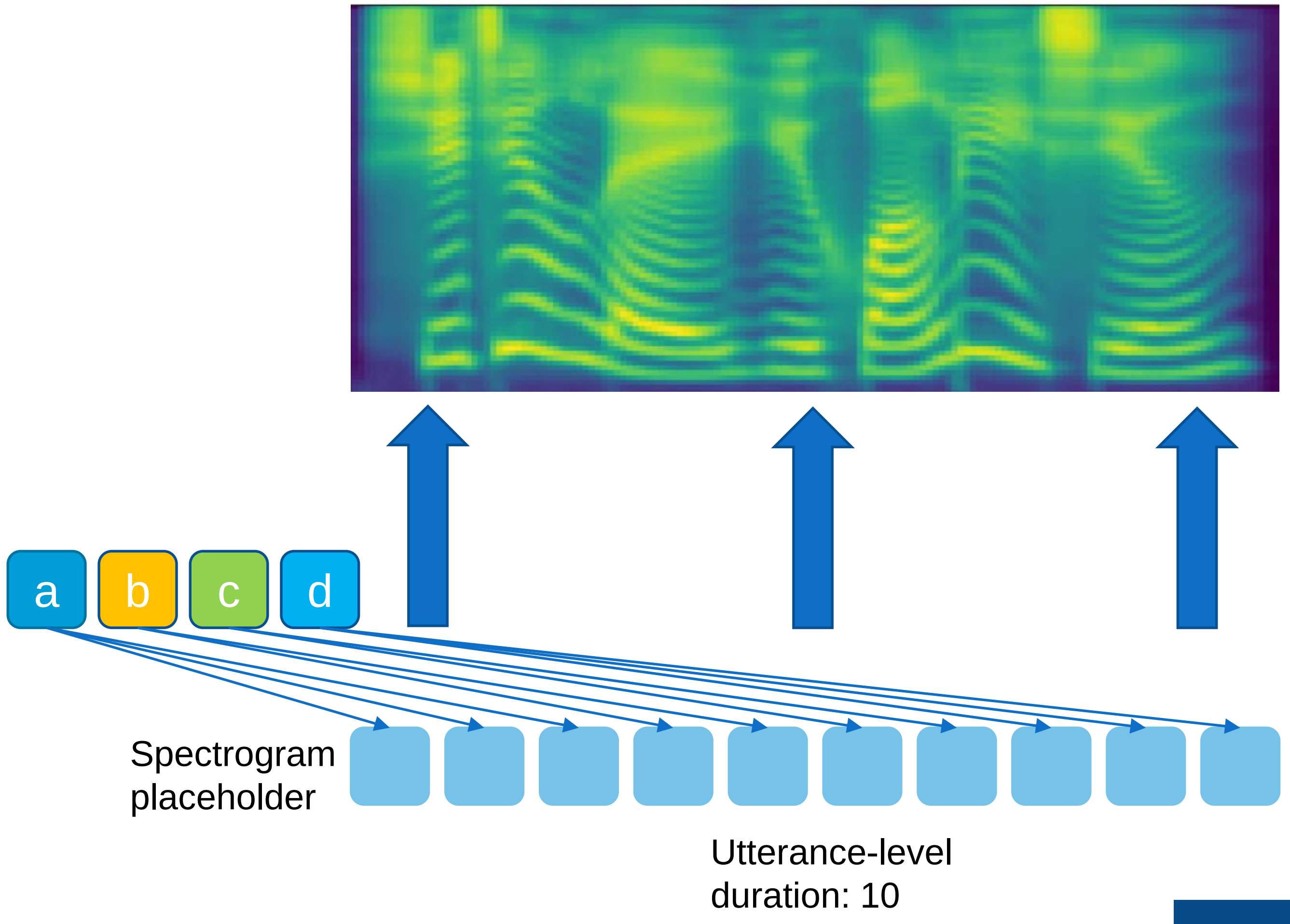
Related Work

Phoneme-level vs. Utterance-level duration

Phoneme-level duration-based models



Utterance-level duration-based models



Related Work

Phoneme-level duration based NAR TTS models

- Non-autoregressive TTS models
  - ✓ Phoneme-level duration-based models
    - FastSpeech, BVAE-TTS
      - Distill durations from an AR-TTS teacher
    - FastsSpeech-2
      - Obtain durations via the HMM force-alignment tool (e.g. MFA)
    - Glow-TTS
      - Extract durations from the dynamic-programming obtained alignments during training
      - Predict durations directly during inference phase
    - Issues
      - Obtaining durations is cumbersome
      - Hard alignment may hurt the naturalness of the synthesized speech

Related Work

Utterance-level duration based NAR TTS models

- Non-autoregressive TTS models
  - ✓ Utterance-level duration-based models
    - Flow-TTS
      - Sample a noise sequence with the utterance-level duration and transform it into the spectrogram using Flow
      - Align the linguistic feature into the frame-level with the positional attention
    - ParaNet
      - Initialize the spectrogram placeholder with the positional encoding
      - Learn the attention alignments from an AR-TTS teacher
    - VARA-TTS
      - Initialize the spectrogram placeholder with the positional encoding
      - Refine the alignment with a layer-by-layer manner

Related Work

Utterance-level duration based NAR TTS models

- Non-autoregressive TTS models
  - ✓ Utterance-level duration-based models
    - ➢ Advantages
      - Utterance-level duration is inherently available
    - ➢ Issues
      - Aligning the linguistic feature onto the spectrogram placeholder is difficult
      - Positional encoding is too simple to express the temporal linguistic information in the spectrogram

Motivations

VAENAR-TTS

- VAENAR-TTS
  - ✓ A novel NAR based approach for TTS based on VAE
  - ✓ Offers greater simplicity and is more straightforwardly end-to-end
    - ➢ Requires only text-spectrogram pair
    - ➢ Avoids the complexities of forced alignment or knowledge distillation processes

Motivations

VAENAR-TTS

- Features of VAENAR-TTS
  - ✓ No need of phoneme-level durations
  - ✓ Using utterance-level duration
  - ✓ Using VAE to learn a more informative spectrogram placeholder Z
  - ✓ The alignment between linguistic feature and the spectrogram placeholder Z is attention-based soft-alignment
  - ✓ VAENAR-TTS is mainly inspired by FlowSeq, a NAR machine-translation model

[13] X. Ma, et al. "FlowSeq: Non-autoregressive conditional sequence generation with generative flow," EMNLP, 2019.

Model Formalization

- Linguistic feature sequence: $X = [x_1, x_2, \ldots, xM]$
- Spectrogram: $Y = [y_1, y_2, \ldots, yN]$
- TTS models: $P(Y|X)$
- AR-TTS version factorization:

$$P(Y|X) = \prod_{i=1}^{N} P(y_i|y_{-i}, X)$$

- Let's introduce a latent variable Z and make it **NAR**:

$$P(Y|X,Z) = \prod_{i=1}^{N} P(y_i|Z, X)$$

- Is this possible?

Model Formalization

$$P(Y|X,Z) = \prod_{i=1}^{N} P(y_i|Z,X)$$

- This is possible when Z represents the **phoneme-level durations**!
- Using phoneme-level durations introduce other issues (e.g. much extra effort, hard aligment)

- Let the model learn Z by itself!

$$P(Y|X) = \int_Z P(Y|X,Z)P(Z|X)dZ$$

- To approximate this formulation: **VAE**

Model Formalization

- ELBO: Evidence lower bound

$$\log P(Y|X) = \log \int_Z P(Y|X,Z)P(Z|X)dZ$$

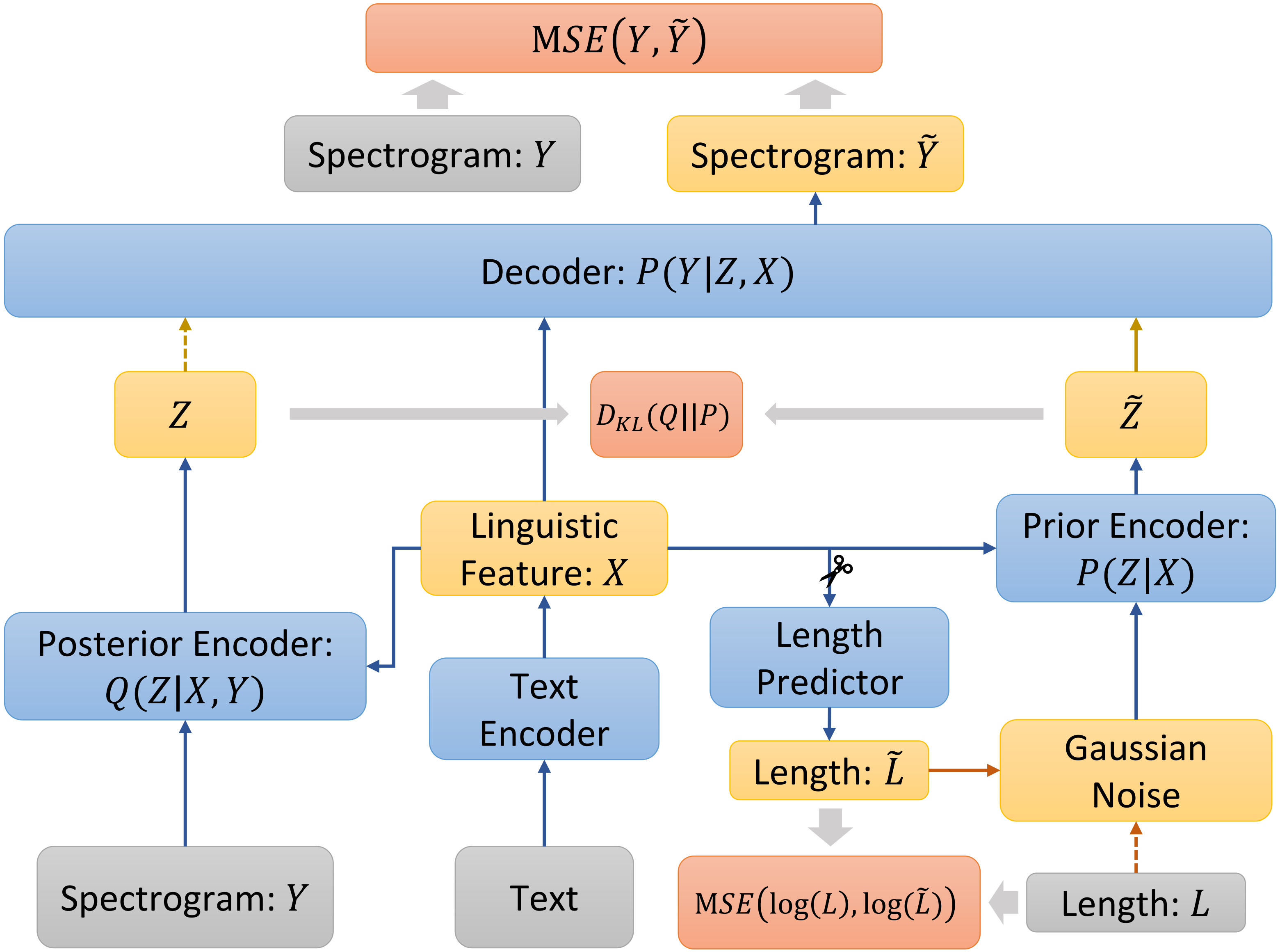
$$= \log \int_Z P(Y|X,Z)Q(Z|X,Y)\frac{P(Z|X)}{Q(Z|X,Y)}dZ$$

$$\geq \int_Z Q(Z|X,Y)\log\left[P(Y|X,Z)\frac{P(Z|X)}{Q(Z|X,Y)}\right]dZ$$

$$= \int_Z Q(Z|X,Y)\log[P(Y|X,Z)]dZ - \int_Z Q(Z|X,Y)\log\left[\frac{Q(Z|X,Y)}{P(Z|X)}\right]dZ$$

$$= E_{Q(Z|X,Y)}[\log P(Y|X,Z)] - D_{KL}(Q(Z|X,Y)||P(Z|X))$$

Model Architecture



- Text Encoder
- Posterior Encoder:
  $Q(Z|X,Y)$
- Prior Encoder:
  $P(Z|X)$
- Decoder:
  $P(Y|Z,X)$
- Length Predictor

Model Architecture

- **Text Encoder**: Similar as that in the Transformer-TTS
  - ✓ Aims to encode the raw character sequence into the context-aware linguistic feature X
- **Posterior Encoder**: $Q(Z|X,Y)$: Transformer decoder structure
  - ✓ Models the posterior distribution of Z given the spectrogram Y and linguistic feature X
  - ✓ More informative about the alignment since it is conditioned on the ground-truth spectrogram
- **Prior Encoder**: $P(Z|X)$: Glow structure: 1x1 Convolution, ActNorm, Affine Coupling
  - ✓ Models the prior distribution of Z conditioned on X
  - ✓ Pushed towards the posterior by the KL-divergence loss during the training phase
- **Decoder**: $P(Y|Z,X)$: Transformer decoder structure
  - ✓ Aligns the linguistic feature X onto the latent variable Z
  - ✓ Reconstructs the spectrogram

[14] D. P. Kingma and P. Dhariwal, “Glow: generative flow with invertible 1 1 convolutions,” NeurIPS, 2018.

Model Architecture

- **Length Predictor**: 1 fully connected layer
  - ✓ Built to infer the utterance-level duration from the linguistic feature X
- Conditioning on the linguistic feature
  - ✓ Accomplished through the attention mechanism
  - ✓ The linguistic feature is used as the key and value being queried
  - ✓ Self-attention blocks and decoder attention blocks from Transformer adopted
- **Loss Function**

$$L = \mathbf{MSE}\left(Y, \tilde{Y}\right) + \alpha \mathbf{D}_{\mathbf{KL}}\left(Q(Z|X,Y)\middle|\middle|P(Z|X)\right) + \beta \mathbf{MSE}\left(\log(L), \log\left(\tilde{L}\right)\right)$$

[15] A. Vaswani, et al. “Attention is all you need,” NeurIPS, 2017.

Model Architecture

Interpretation

- What does Z represent?
  - ✓ Z serves as the spectrogram placeholder to be aligned with the linguistic feature
  - ✓ Z encodes the alignment information between the linguistic feature and the spectrogram

Model Architecture

Advantages of the proposed model

- Compared to other NAR-TTS models
  - ✓ Requires **no phoneme-level durations**, more straightforwardly end-to-end
  - ✓ Attention based **soft-alignment** between the linguistic feature and the spectrogram enables more natural synthesized speech
  - ✓ The spectrogram placeholder Z is **alignment and linguistic aware**, which can be more easily aligned with the linguistic feature

Model Architecture

Alignment learning

- To learn better alignment
  - Transformer based components are used
    - Transformer decoder, self-attention block
  - **Annealing reduction factor** strategy of the spectrogram
    - Larger reduction factor, faster alignment convergence
    - Smaller reduction factor enables fine-grained posterior, spectrogram learning
  - **Scaled positional encoding**
    - $PE(pos, 2i) = \sin\left(\frac{pos * \boldsymbol{s}}{10000^{\frac{2i}{d_{model}}}}\right)$
    - No significant effects, but can help **stabilize** the loss when changes the reduction factor
    - Can be used to control the speaking rate
  - **Causality mask** on the frame-level feature side **self-attention**
    - Help reduce repetition errors

Experiments

Experimental setup

- Dataset: LJSpeech
  - ✓ 13,100 English utterances, female speaker
  - ✓ Two 131-utterance subsets randomly sampled out as the validation and test set
  - ✓ Remaining as the training set
- Experimental setup
  - ✓ Weights for KL-divergence and the utterance-level duration loss are set to 1.0e-5 and 1.0 respectively
  - ✓ r is initially set to 5 and is decreased by 1 every 200 training epochs until it reaches 2, after which r remains as 2 for the rest of the training epochs
  - ✓ Train the model for 2000 epochs and the final model checkpoint is used for evaluation
  - ✓ During the training phase, the initial noise for the prior encoder is sampled from the normal distribution, while for inference it is set to all zeros

Experiments

Synthesis quality and speed experiments

- MOS
  - ✓ 10 randomly selected sentences
  - ✓ presented with 95% confidence intervals
  - ✓ VAENAR-TTS achieves best naturalness, **comparable or better than Tacotron2**
- RTF
  - ✓ Conducted on a single RTX2080Ti GPU with batch size of 1
  - ✓ Averaged over 10 runs on the whose test set
  - ✓ Comparable with other NAR-TTS models, about **18× faster than Tacotron2**

Table 1: *Comparison results of different TTS models*

Model	**MOS**	**RTF**(Sec)
Ground-Truth	4.56 ± 0.09	-
Hifi-GAN-Resyn	4.47 ± 0.10	-
Tacotron2	4.03 ± 0.12	1.35×10^{-1}
FastSpeech2	3.83 ± 0.14	$\mathbf{4.21 \times 10^{-3}}$
Glow-TTS	3.62 ± 0.13	9.39×10^{-3}
BVAE-TTS	3.16 ± 0.13	$\mathbf{4.21 \times 10^{-3}}$
VAENAR-TTS	$\mathbf{4.15 \pm 0.12}$	7.45×10^{-3}
RF5	3.43 ± 0.14	6.99×10^{-3}
RF4	3.83 ± 0.13	7.30×10^{-3}
RF3	3.84 ± 0.14	7.43×10^{-3}

Experiments

Alignment learning experiments

- Comparing RF5, RF4, RF3
  - ✓ Significant improvement of speech naturalness when *r* decreased from 5 to 4
  - ✓ MOS gap between RF4 and RF3 is small
  - ✓ RTFs do not vary too much
- RF4, RF3
  - ✓ Achieves better MOS than Glow-TTS and BVAE-TTS, and comparable with FastSpeech2
- With the **annealing reduction factor** strategy, and the final *r* as 2, VAENAR-TTS achieves much better quality

Table 1: *Comparison results of different TTS models*

Model	**MOS**	**RTF**(Sec)
Ground-Truth	4.56 ± 0.09	-
Hifi-GAN-Resyn	4.47 ± 0.10	-
Tacotron2	4.03 ± 0.12	1.35×10^{-1}
FastSpeech2	3.83 ± 0.14	$\mathbf{4.21 \times 10^{-3}}$
Glow-TTS	3.62 ± 0.13	9.39×10^{-3}
BVAE-TTS	3.16 ± 0.13	$\mathbf{4.21 \times 10^{-3}}$
VAENAR-TTS	$\mathbf{4.15 \pm 0.12}$	7.45×10^{-3}
RF5	3.43 ± 0.14	6.99×10^{-3}
RF4	3.83 ± 0.13	7.30×10^{-3}
RF3	3.84 ± 0.14	7.43×10^{-3}

Experiments

Alignment learning experiments

- Attention alignments with larger *r* converge much faster
- Using causality mask helps reduce the repetition errors

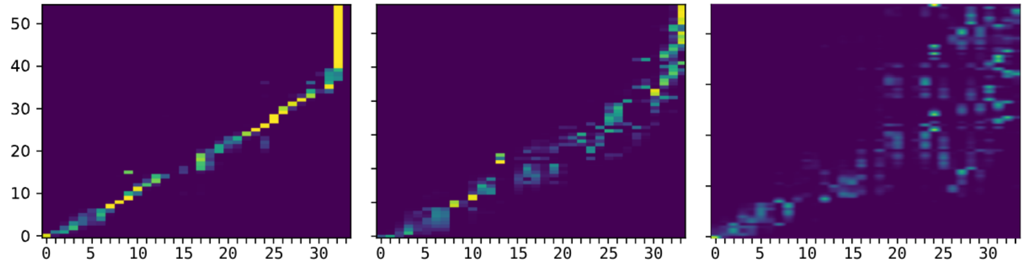


Figure 2: *Attention alignments of RF5 model (left), RF4 model (middle) and RF3 model (right) after 56 training epochs*

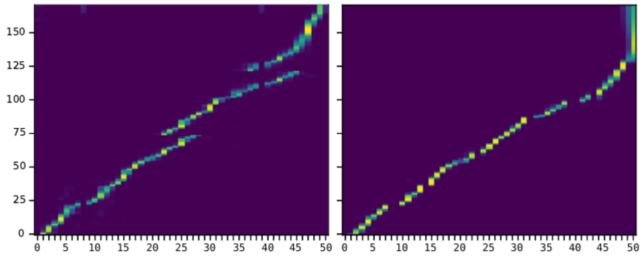
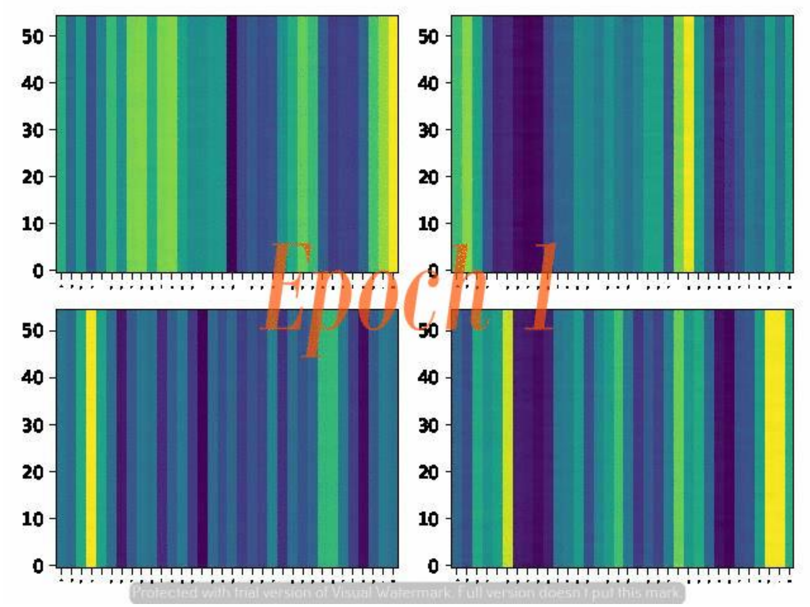


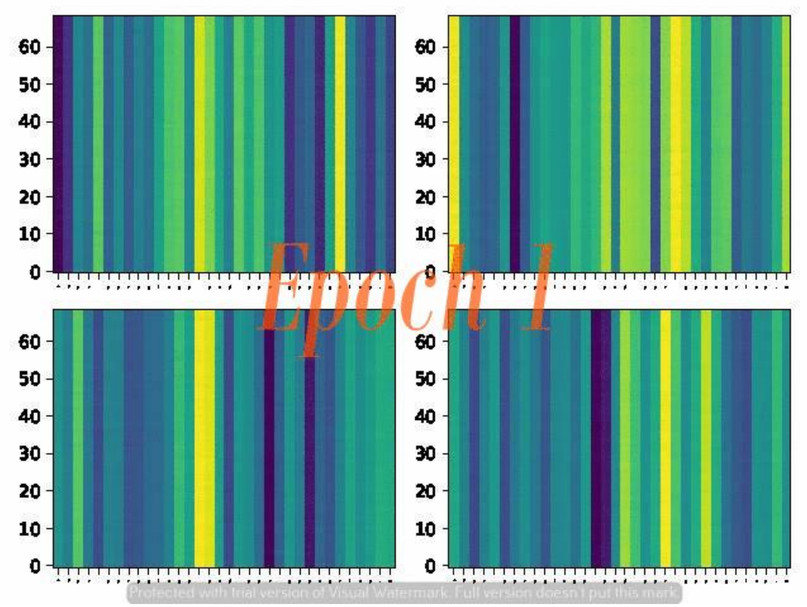
Figure 3: *Decoding attention alignments of models without (left) versus with (right) causality mask in acoustic side self-attention, where the vertical and horizontal axis denotes the decoder and encoder step, respectively.*

Experiments

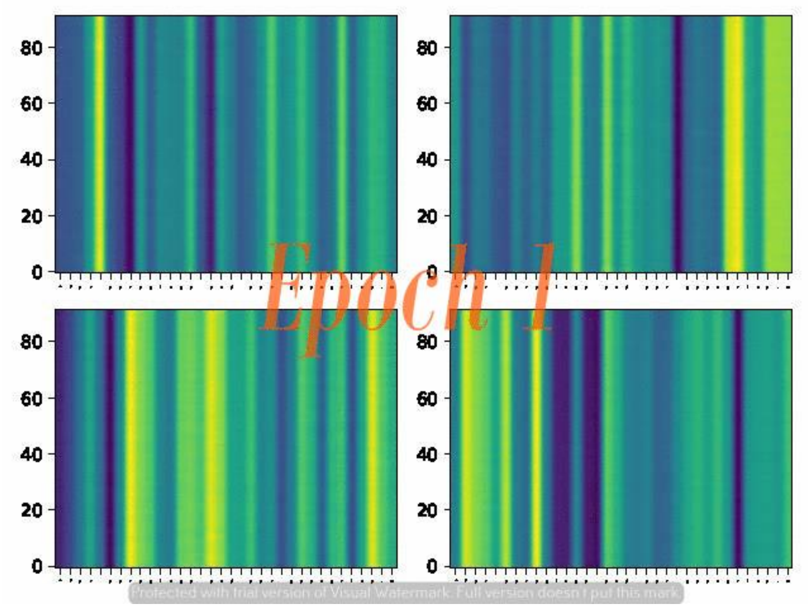
Alignment learning experiments



RF5



RF4



RF3

Experiments

Code, paper, pretrained models and demo

- https://github.com/thuhcsi/VAENAR-TTS

Other results

- Mandarin TTS (DataBaker BZNSYP opensource corpus)
- Emotional TTS

Neutral Happy Fear Disgust Angry Sad Surprised

- Cantonese TTS
  - ✓ 並處罰款人民幣五千元
  - ✓ 他認為舖內衛生符合標準
  - ✓ 他們不排除尋求法律仲裁

Conclusions

- VAENAR-TTS is a more end-to-end NAR-TTS model
  - ✓ No need of phoneme-level durations
- The synthesis quality achieves SOTA while the synthesis speed is fast
- Condition inputs (e.g. emotion labels, speaker ids) can be easily added

References

- [1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017, F. Lacerda, Ed. ISCA, 2017, pp. 4006–4010.
- [2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 4779–4783.
- [3] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 6706–6713.
- [4] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," Proc. ICLR, pp. 214–217, 2018.
- [7] Y. Lee, J. Shin, and K. Jung, "Bidirectional variational inference for non-autoregressive text-to-speech," in International Conference on Learning Representations, 2021.
- [8] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech: Fast, robust and controllable text to speech," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alch´e-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 3165–3174.
- [9] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," CoRR, vol. abs/2006.04558, 2020.
- [10] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.
- [11] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 7586–7598.
- [12] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020. IEEE, 2020, pp. 7209–7213.
- [13] X. Ma, C. Zhou, X. Li, G. Neubig, and E. Hovy, "Flowseq: Non-autoregressive conditional sequence generation with generative flow," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, November 2019.
- D. P. Kingma and P. Dhariwal, "Glow: generative flow with invertible 1×1 convolutions," in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 10 236–10 245.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

THUHCSI
清华大学人机语音交互实验室

Thank You!
Q&A