

**Summary of
introduction
Text-to-speech
and NVIDIA NeMo**

Created by TN AI Ton



Reference source

Main references and sorting topics by

https://github.com/NVIDIA/NeMo/blob/main/tutorials/tts/NeMo_TTS_Primer.ipynb

Additional references

<https://docs.nvidia.com/nemo-framework/index.html>

<https://www.nvidia.com/en-us/glossary/text-to-speech/>

<https://medium.com/towards-data-science/text-to-speech-with-tacotron-2-and-fastspeech-using-espnet-3a711131e0fa>

<https://paperswithcode.com/method/wavernn>

https://pytorch.org/hub/nvidia_deeplearningexamples_fastpitch/

What is Text-to-Speech?

Text-to-Speech (TTS) หรือ ระบบสังเคราะห์เสียงพูด คือเทคโนโลยีที่แปลง

ข้อความเป็นเสียงพูดโดยอัตโนมัติ

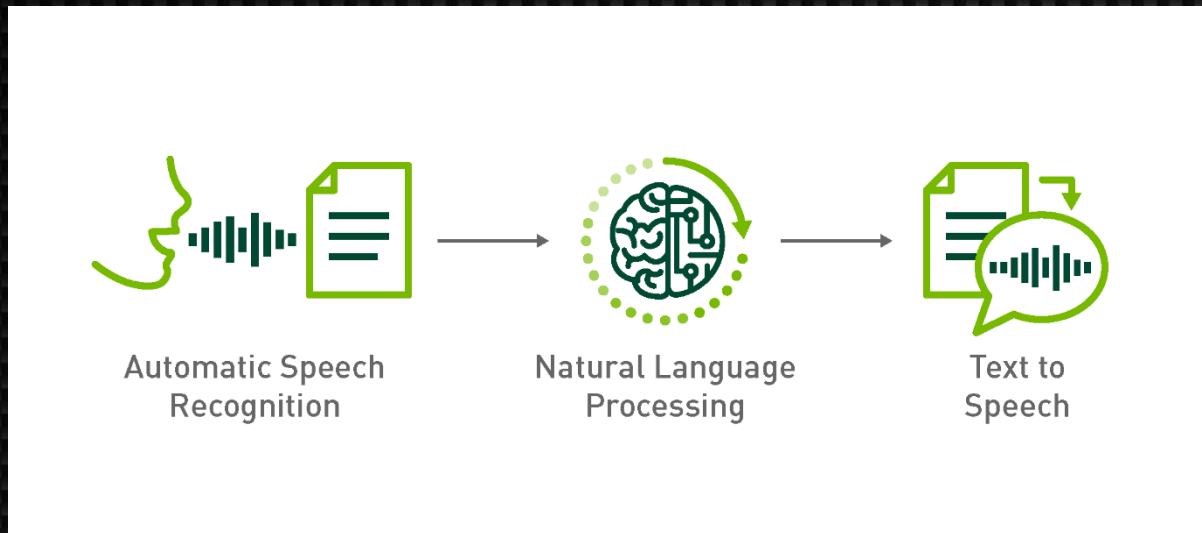


Why TTS?

1. **TTS** กำลังพัฒนาไปสู่ AI สนทนา โดยร่วมกับ **ASR** และ **NLP**

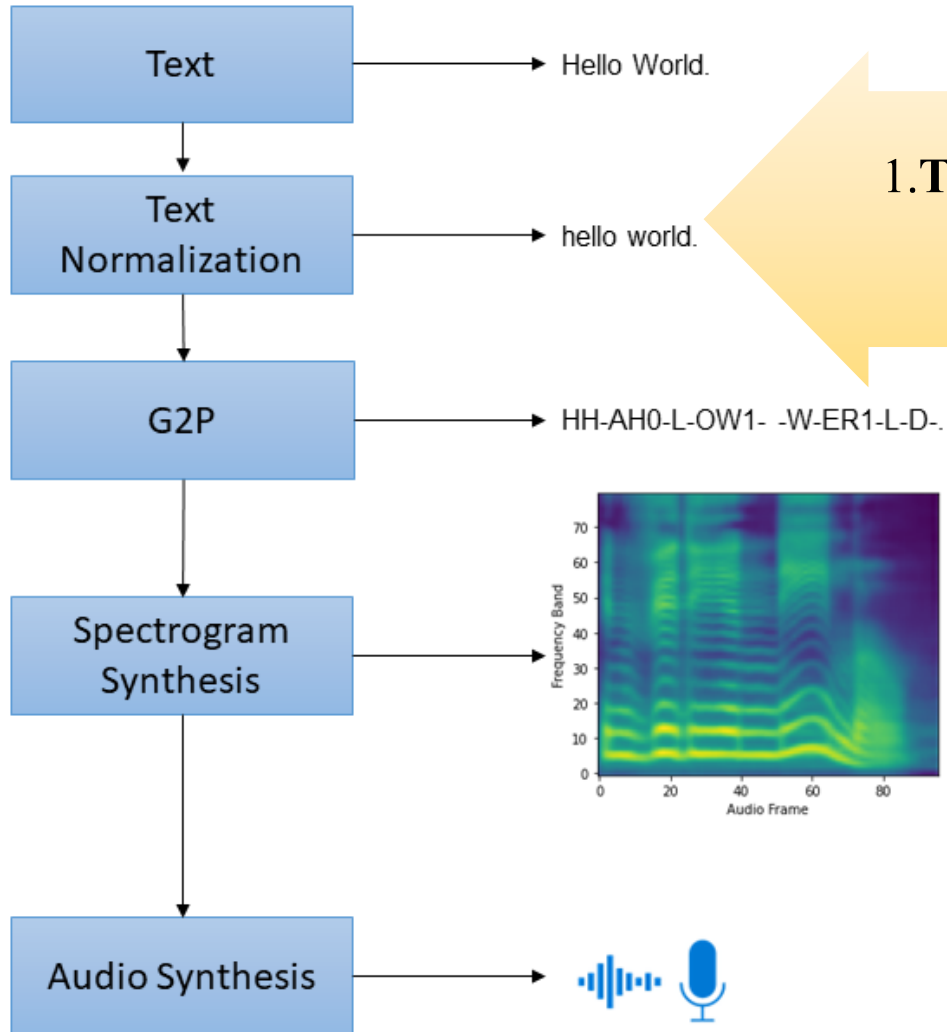
2. นักการตลาดทางโทรศัพท์ เริ่มใช้ **TTS** แทนพนักงานมนุษย์ โดยใช้หุ่นยนต์สนทนาที่สมจริงมากขึ้นเรื่อยๆ

และ อื่นๆ



The TTS Pipeline

กระบวนการทำงานของ TTS มี 4 ขั้นตอนหลัก

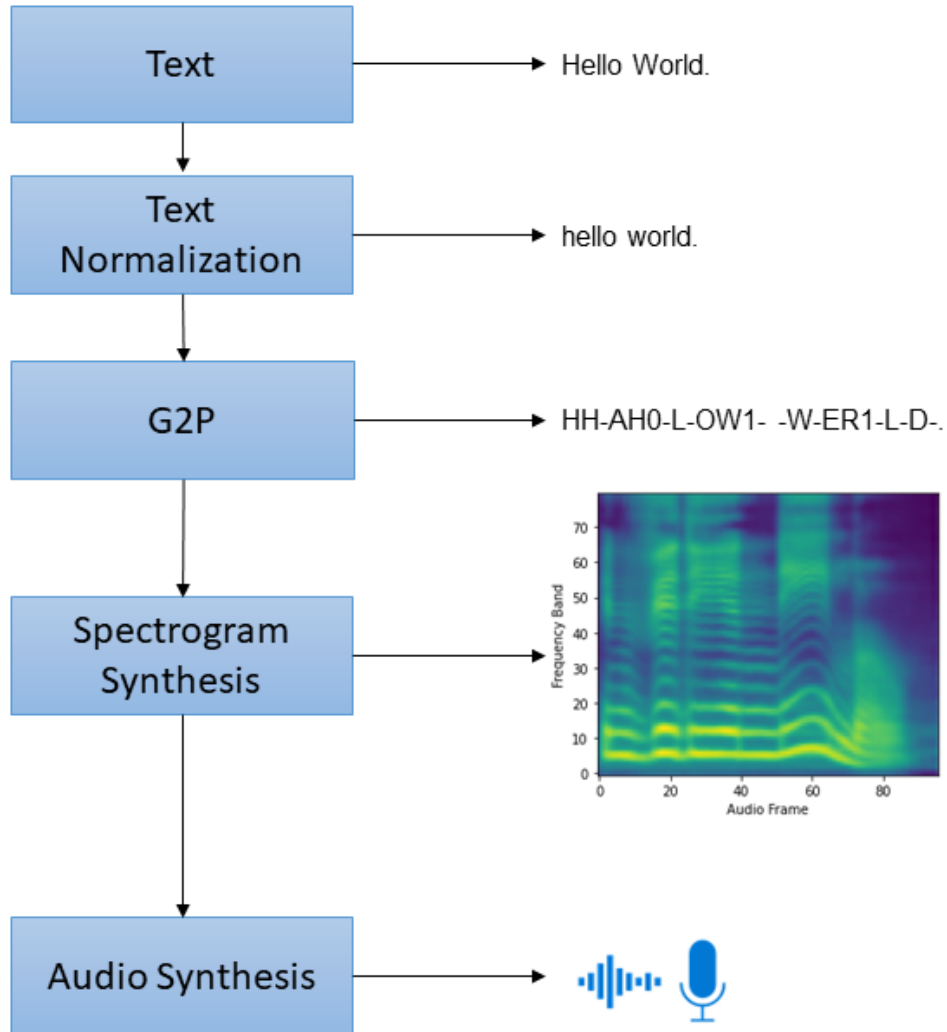


1. **Text Normalization** : แปลงข้อความดิบให้อยู่ในรูปที่อ่านได้
(เช่น "Mr." → "Mister")

Normalization Type	Input	Output
Abbreviations	Mr.	mister
Acronyms	TTS	text to speech
Numbers	42	forty two
Decimals	1.2	one point two
Roman Numerals	VII	seventh
Cardinal Directions	N E S W	north east south west
URL	www.github.com	w w w dot github dot com

The TTS Pipeline

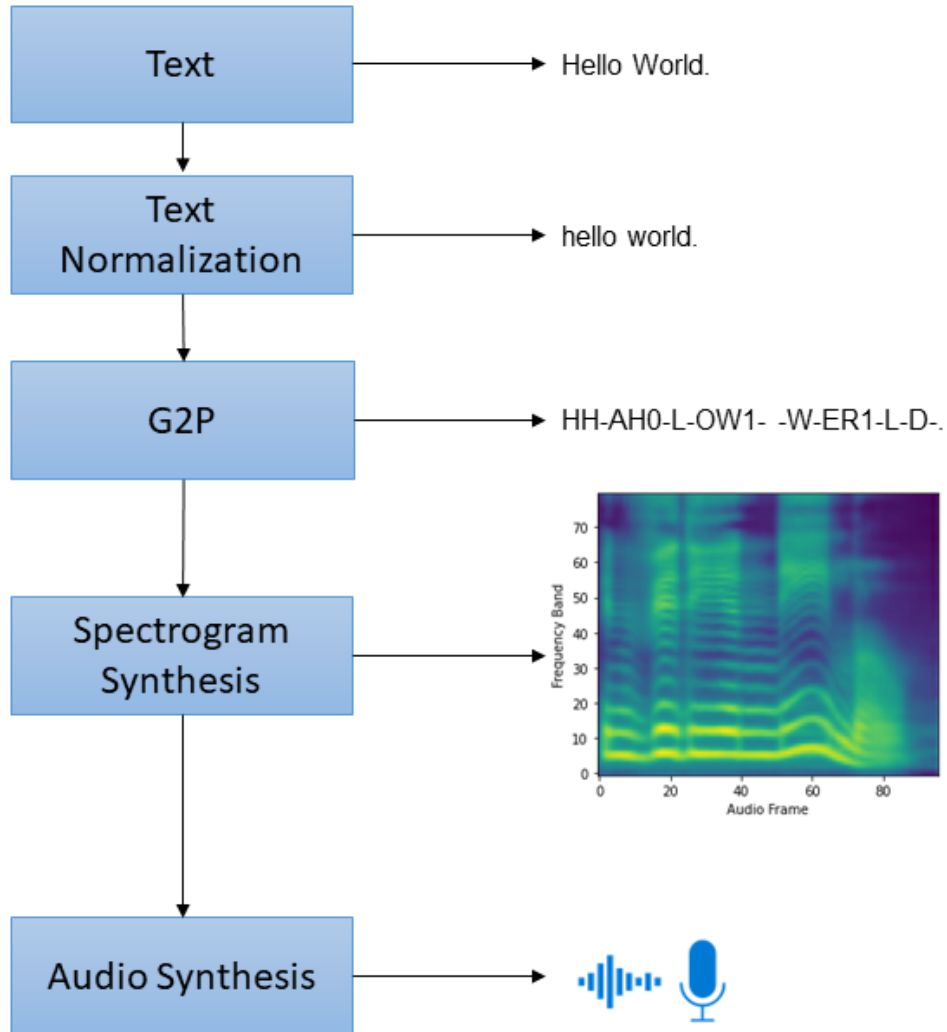
กระบวนการทำงานของ TTS มี 4 ขั้นตอนหลัก



2. Grapheme to Phoneme Conversion (G2P) : แปลงตัวอักษรเป็นหน่วยเสียง (เช่น "Hello" → "HH-AH0-L-OW1")

The TTS Pipeline

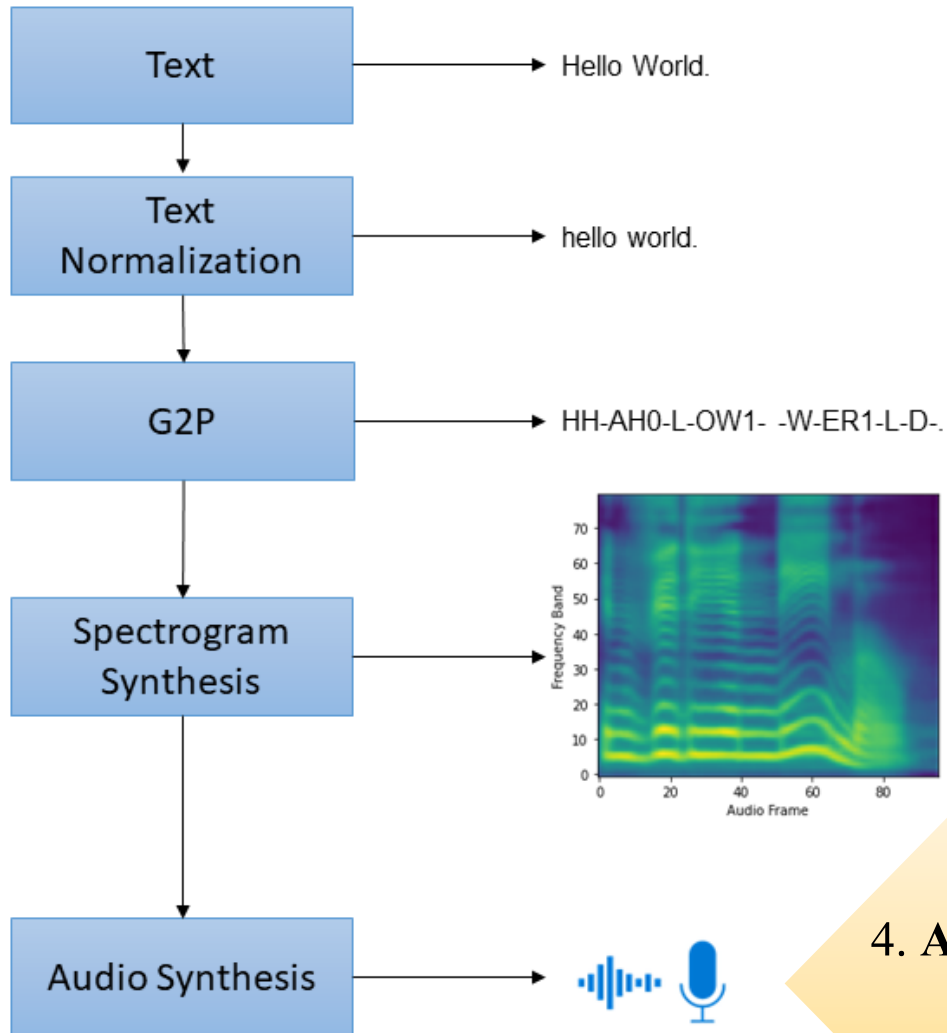
กระบวนการทำงานของ TTS มี 4 ขั้นตอนหลัก



3. Spectrogram Synthesis : แปลงหน่วยเสียงเป็น สเปกโตรแกรม

The TTS Pipeline

กระบวนการทำงานของ TTS มี 4 ขั้นตอนหลัก

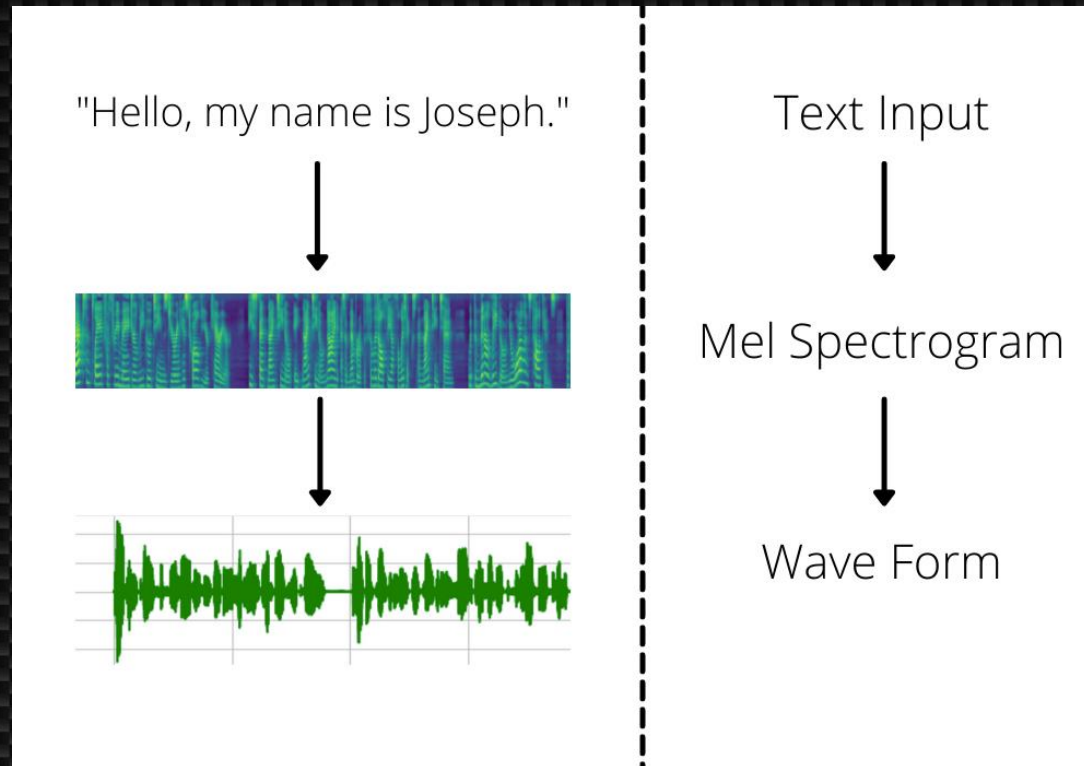


4. **Audio Synthesis** (Spectrogram Inversion) : แปลงสเปกโตรแกรม เป็นเสียงพูด โดยใช้ Vocoder

Spectrogram Synthesis

Spectrogram Synthesis กระบวนการแปลงข้อความ (Text) หรือโฟนีม (Phonemes) ให้กลายเป็น

Spectrogram ซึ่งเป็น ตัวแทนของเสียงในรูปของความถี่เทียบกับเวลา



Why use a spectrogram?

1. ประสิทธิภาพคำนวณเร็วขึ้น ลดขนาดข้อมูลได้ ~5 เท่าเมื่อเทียบกับเสียงดิบ (ใช้ Fast Fourier Transform)
2. ลดความยาวของข้อมูล ใช้ได้ดีกับโมเดล Deep Learning โดยเฉพาะ RNN/LSTM ที่มีปัญหากับซีแวนซ์ที่ยาว
3. รองรับ CNN/Transformer ใช้กับโมเดลได้มีประสิทธิภาพกว่าเสียงดิบ

Types of TTS models

ประเภทของโมเดล TTS มี 2 ประเภทหลัก



Auto-Regressive Models

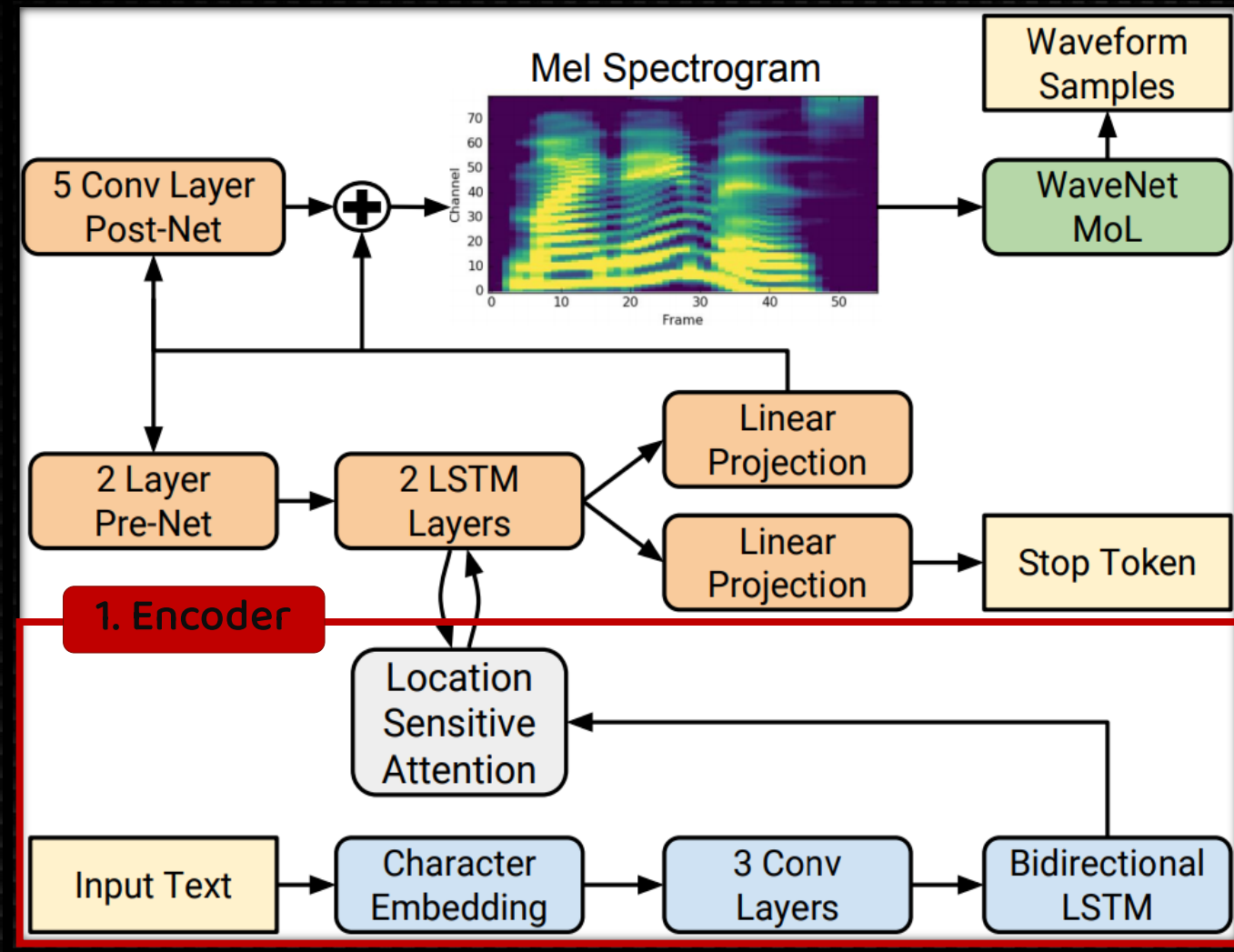
ทำนายสเปกโตรแกรมทีละขั้น ใช้ Attention
หรือ Duration Prediction

Parallel Models

ทำนายแบบขนาน ใช้ Duration
Prediction

Auto-Regressive Models

Tacotron2 Model Architecture



1.Encoder

[แปลงข้อความเป็นตัวแทนเสียง]

Input Text : ป้อนข้อความเข้าโมเดล

Character Embedding : แปลงตัวอักษรเป็น
เวกเตอร์

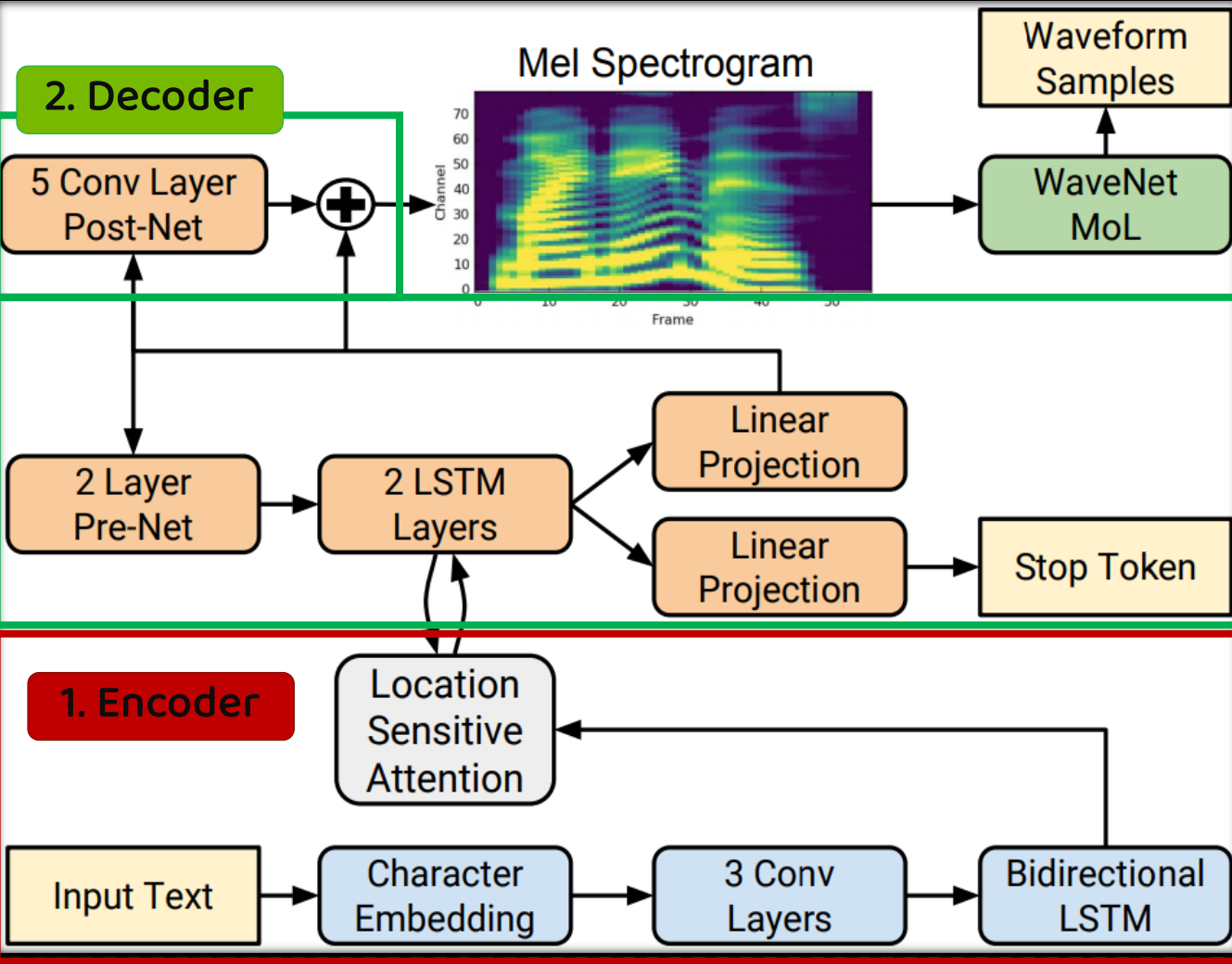
3 Conv Layers + BiLSTM : วิเคราะห์ลำดับ
ข้อความและพิจารณาการออกเสียง

Location Sensitive Attention : ช่วยเลือกข้อมูลที่เหมาะสมจาก Encoder

Auto-Regressive Models

Tacotron2 Model Architecture

2. Decoder



2. Decoder

[สร้าง Mel Spectrogram]

2 Layer Pre-Net: เตรียมข้อมูลก่อนป้อนเข้า

Decoder

2 LSTM Layers: ทำนาย Mel Spectrogram ทีละเฟรม

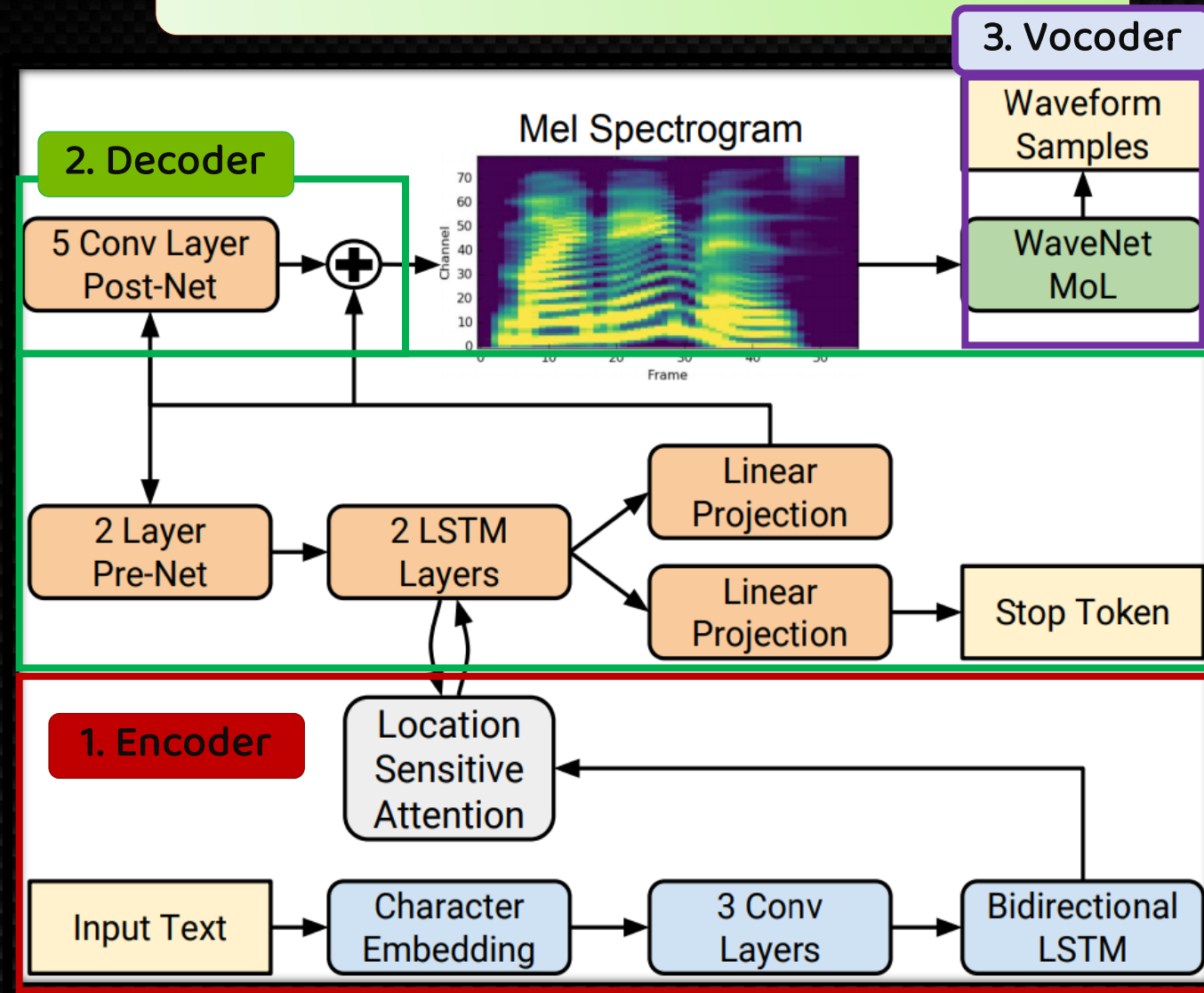
Linear Projection: แปลงผลลัพธ์เป็น

Stop Token: ทำนายว่าควรหยุดสร้างเสียงหรือไม่

5 Conv Layer Post-Net: ปรับปรุงคุณภาพ Spectrogram

Auto-Regressive Models

Tacotron2 Model Architecture



3. Vocoder

[แปลง Spectrogram เป็นเสียงพูด]

WaveNet MoL แปลง **Mel Spectrogram** เป็นคลื่นเสียง (Waveform Samples)

Summary

1. Encoder-Decoder + Attention

เพื่อสร้าง Mel Spectrogram

2 ใช้ Pre-Net, Stop Token, และ

Post-Net เพื่อปรับคุณภาพเสียง

3. WaveNet MoL เป็น Vocoder แปลง

Spectrogram เป็นเสียงที่เป็นธรรมชาติ

Types of TTS models

ประเภทของโมเดล TTS มี 2 ประเภทหลัก



Auto-Regressive Models

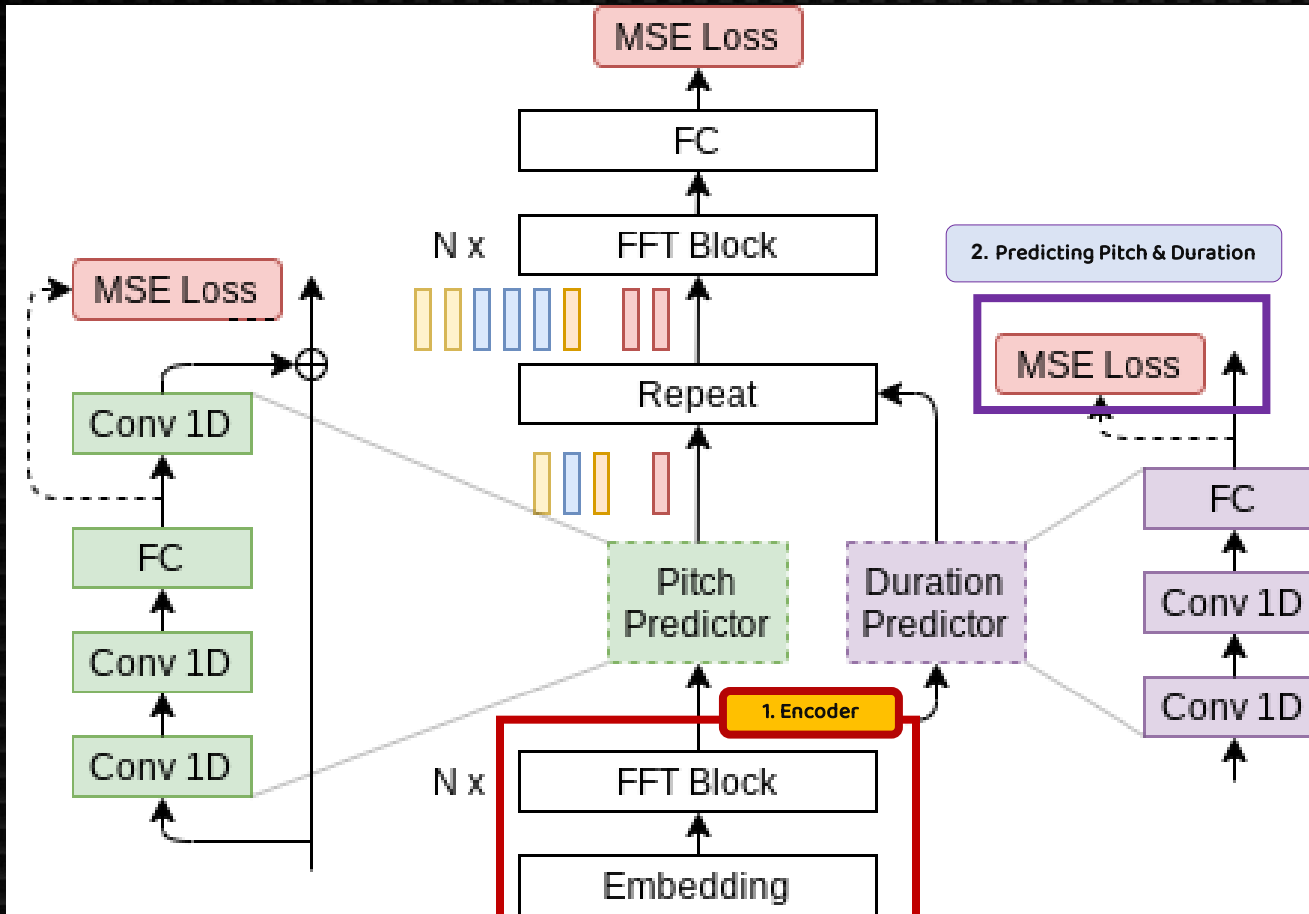
ทำนายสเปกโตรแกรมทีละขั้น ใช้ Attention
หรือ Duration Prediction

Parallel Models

ทำนายแบบขนาน ใช้ Duration
Prediction

Parallel Models

FastPitch Model Architecture



1.Encoder

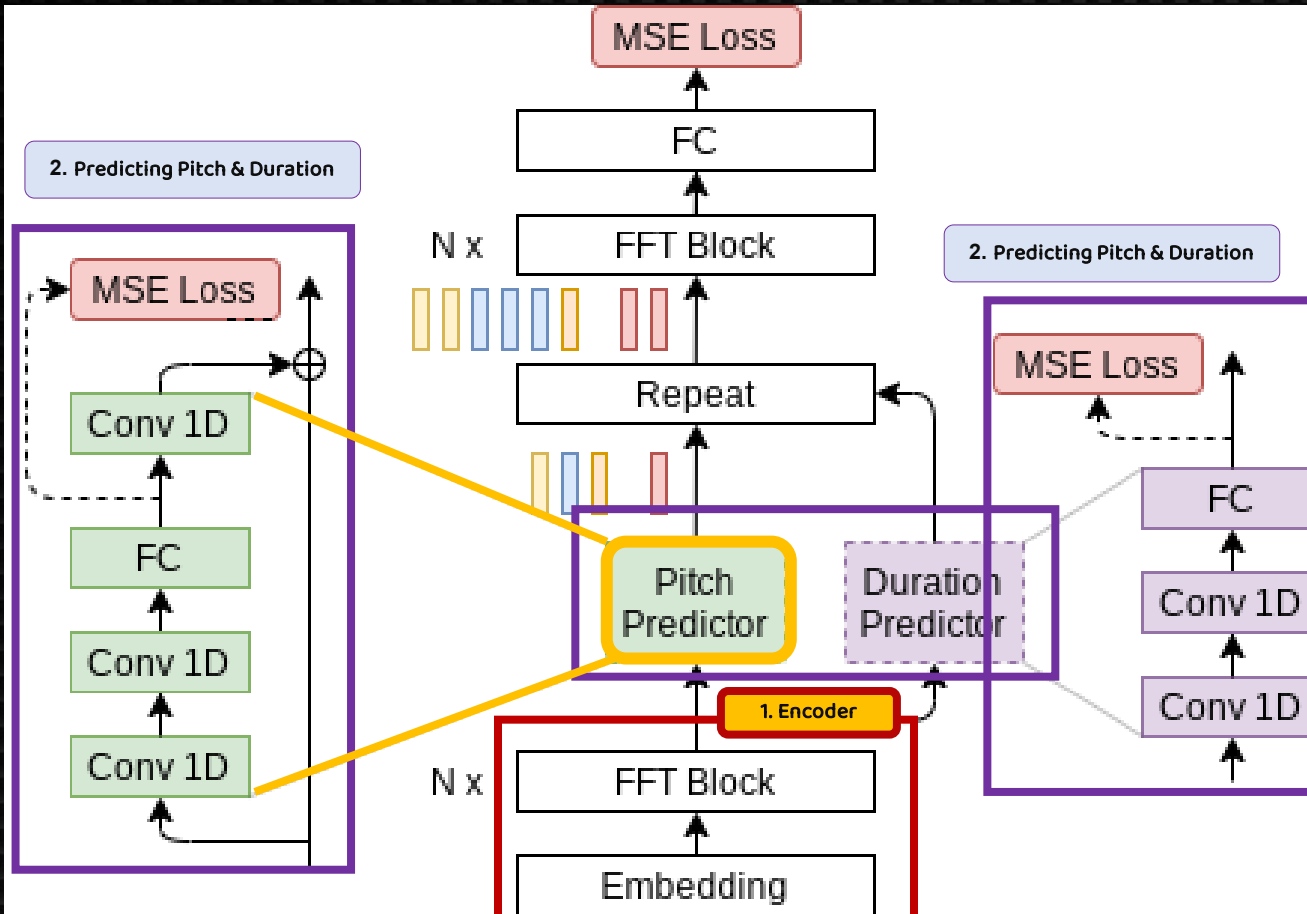
[แปลงข้อความเป็นตัวแทนเสียง]

Embedding : แปลงอักขระเป็นเวกเตอร์

FFT Block (Feed-Forward Transformer - FFT) : วิเคราะห์โครงสร้างเสียง

Parallel Models

FastPitch Model Architecture



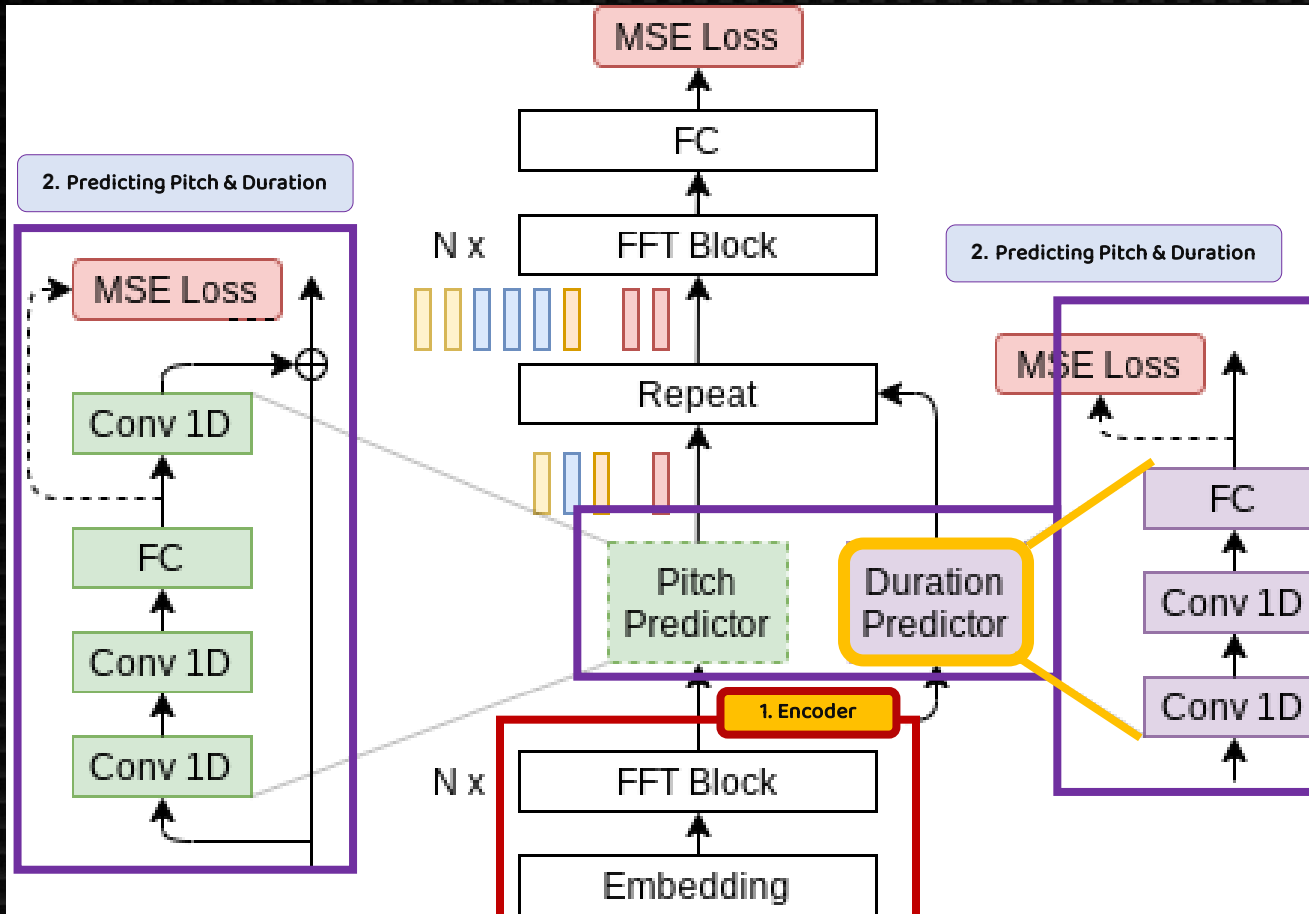
2. Predicting Pitch & Duration [ทำนายโทนเสียงและระยะเวลา]

2.1 Pitch Predictor: ทำนายโทนเสียงของแต่ละอักขระใช้

ใช้ **Conv1D + FC Layers** → คำนวณ **MSE Loss**
เพื่อปรับค่าการทำนายให้แม่นยำ

Parallel Models

FastPitch Model Architecture



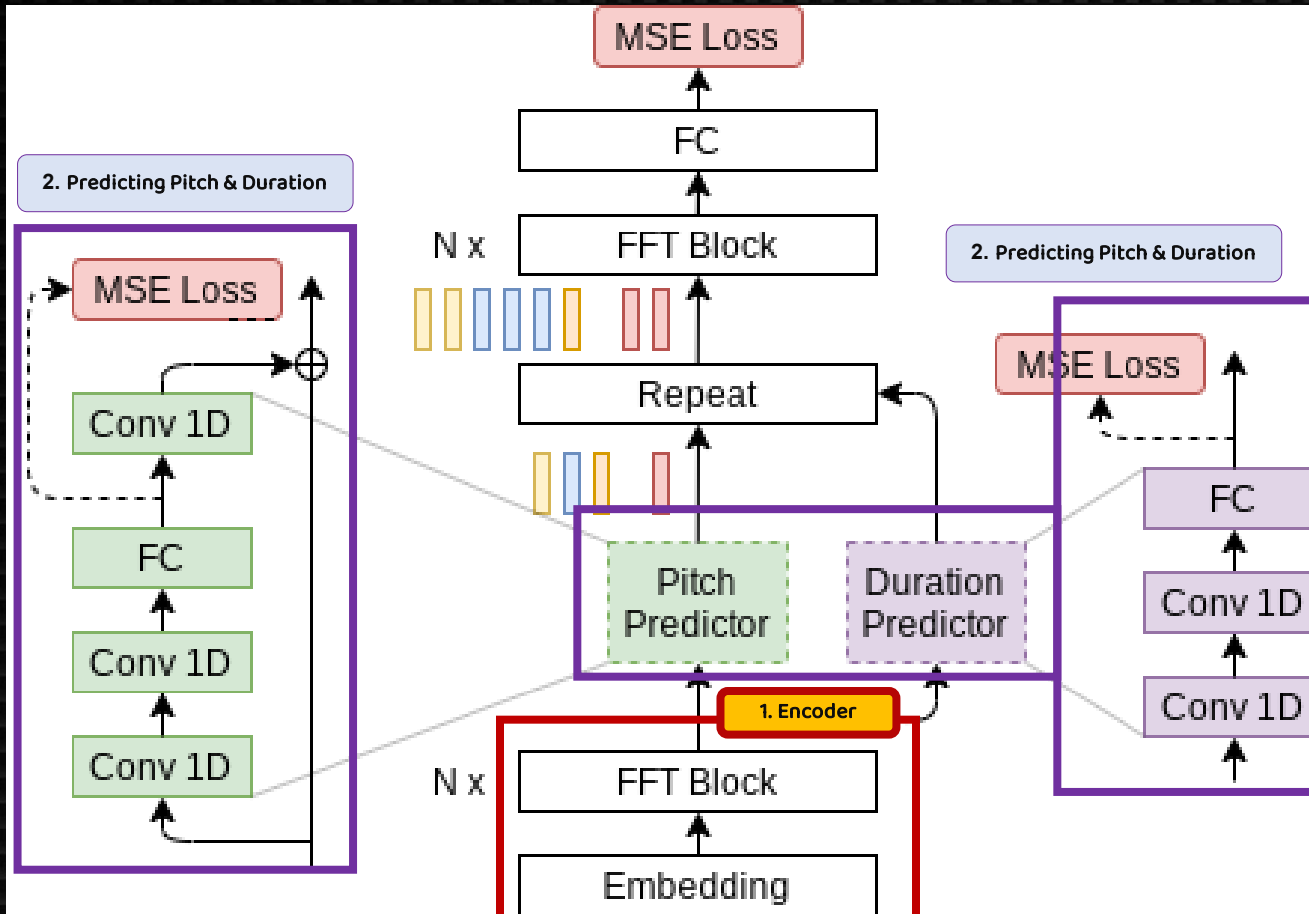
2. Predicting Pitch & Duration [ทำนายโทนเสียงและระยะเวลา]

2.2 **Duration Predictor** : ทำนายระยะเวลาของแต่ละอักขระใช้

ใช้ **Conv1D + FC Layers** → คำนวณ **MSE Loss**
เพื่อปรับค่าการทำนายให้แม่นยำ

Parallel Models

FastPitch Model Architecture



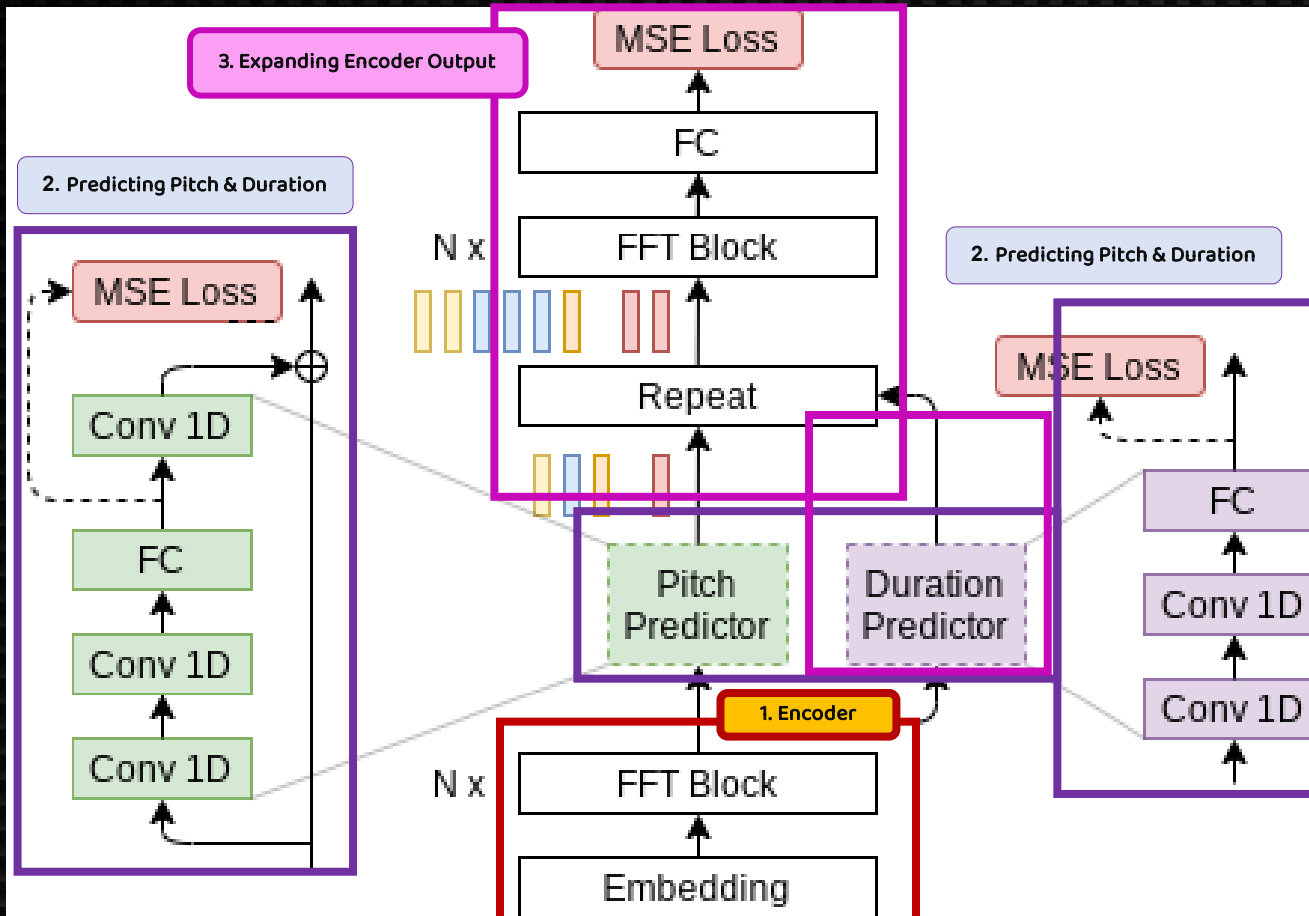
2. Predicting Pitch & Duration

[ทำนายโทนเสียงและระยะเวลา]

ระหว่าง Training Time ใช้ค่าจริงของ Pitch และ Duration เพื่อสอนโมเดล (คล้าย Teacher Forcing)

Parallel Models

FastPitch Model Architecture



3. Expanding Encoder Output

[ขยายผลลัพธ์ของ Encoder ให้ตรงกับ Spectrogram]

Duration Predictor (Conv1D + FC) : ทำนายจำนวนครั้งที่ต้องทำซ้ำ แล้ว คำนวณ MSE Loss

Repeat Layer : ทำซ้ำ Encoder Output ตาม Duration ที่คาดการณ์

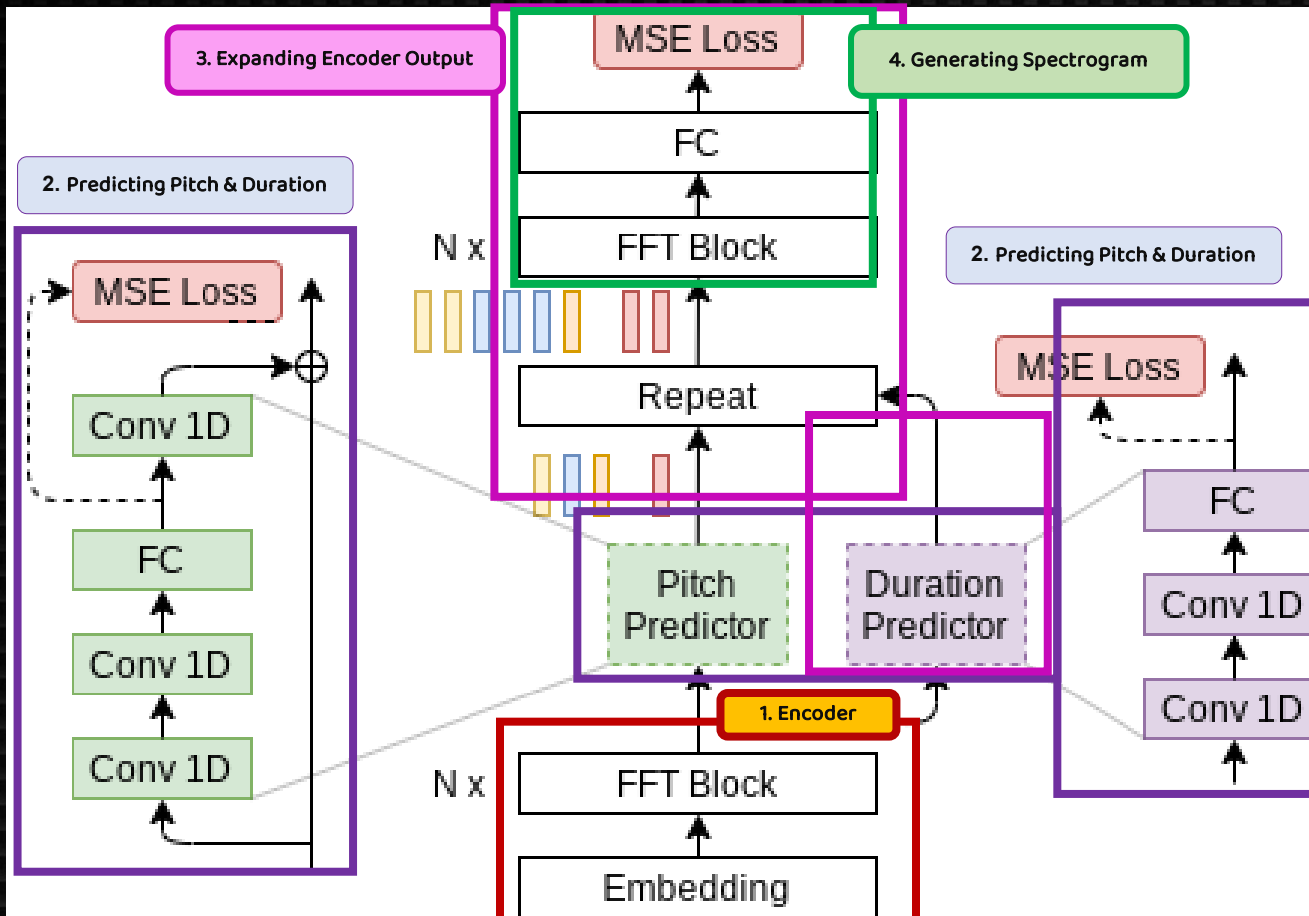
FFT Block : ประมวลผลข้อมูลที่ถูกขยาย

FC (Fully Connected Layer) : แปลงข้อมูลเป็นคุณลักษณะเสียง

คำนวณ MSE Loss : ปรับผลลัพธ์ให้แม่นยำ

Parallel Models

FastPitch Model Architecture



4. Generating Spectrogram [สร้าง Mel Spectrogram]

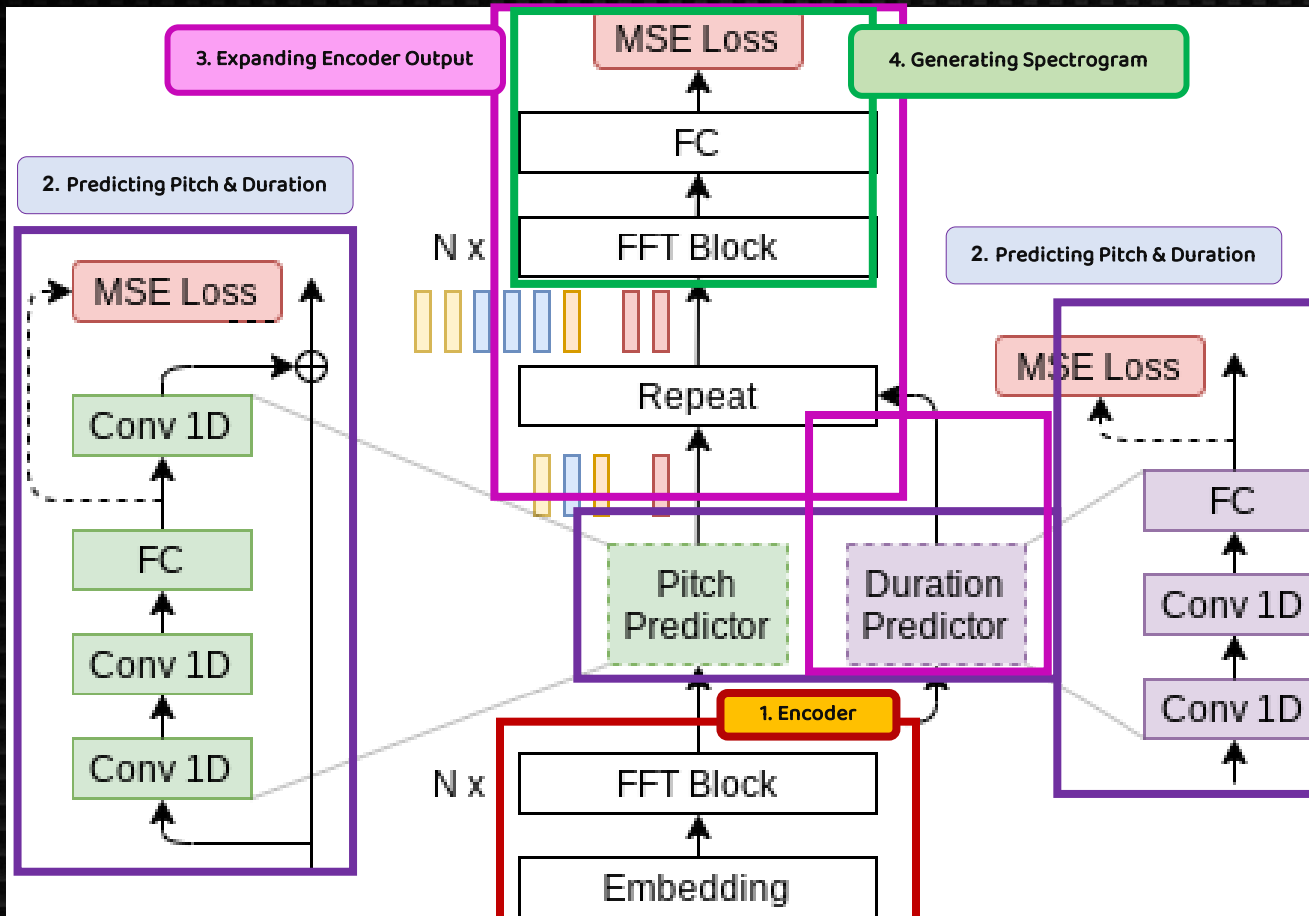
FFT Block (FFTr): แปลง Encoder Output เป็น Spectrogram

FC (Fully Connected Layer): แปลงผลลัพธ์ให้เป็น Mel Spectrogram

MSE Loss: ปรับผลลัพธ์ให้แม่นยำขึ้น

Parallel Models

FastPitch Model Architecture



Summary

- 1. Encoding :** FFT Block วิเคราะห์ตัวอักษรเป็นตัวแทนเสียง
- 2. Predicting Pitch & Duration :** Conv1D + FC ทำนายโทนเสียงและระยะเวลา
- 3. Expanding Encoder Output :** Repeat Layer ทำซ้ำข้อมูลตาม Duration, FFT Block ประมวลผล
- 4. Generating Spectrogram** FFT Block + FC แปลงข้อมูลเป็น Mel Spectrogram


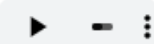
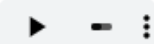

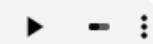
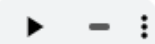

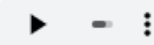
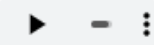

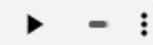
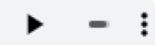




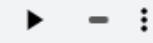
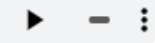

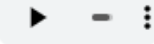
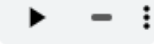
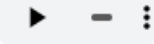
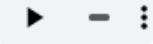
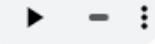
Compare Auto-Regressive Models with Parallel Models

คุณสมบัติ	Auto-Regressive Models	Parallel Models ⚡
หลักการทำงาน	ทำนายสเปกโตรแกรม ทีละเฟรม ต่อเนื่องกัน	ทำนายสเปกโตรแกรม ทุกเฟรมพร้อมกัน
ความเร็วในการประมวลผล	ช้า เพราะต้องรอเฟรมก่อนหน้า	เร็วมาก เพราะคำนวณทุกเฟรมพร้อมกัน
Inference Speed	ช้า เพราะต้องรอแต่ละเฟรมก่อนสร้างเฟรม ถัดไป	เร็วขึ้น 100x เนื่องจากใช้ Duration Prediction
โครงสร้างหลัก	ใช้ RNN/LSTM + Attention ในการเรียนรู้ ลำดับเสียง	ใช้ CNN/Transformer + Duration Prediction
การฝึก (Training)	ฝึกยากกว่า เสี่ยงต่อ Vanishing/Exploding Gradients	ฝึกง่ายกว่า เพราะคำนวณแบบขนาน
คุณภาพเสียง	ดีน้ําไหลกว่า เพราะใช้ Attention	ควบคุมจังหวะเสียงได้ดี ด้วย Duration Prediction
ตัวอย่างโมเดล	Tacotron 2 (ใช้ Attention + WaveNet)	FastPitch (ใช้ Duration Prediction)

Audio Synthesis (Spectrogram Inversion)

- * เรียกอีกชื่อว่า **Spectrogram Inversion**
- * เป็นกระบวนการ แปลง **Spectrogram** ให้เป็นเสียงพูด
- * ใช้ **Vocoder** ในการสร้างคลื่นเสียงจาก **Spectrogram**

Sound samples from the DSP15 paper

Paper Reference (and original signal)	G&L 1 iter.	G&L 10 iter.	SPSI	SPSI+G&L 1	SPSI+G&L 10
Male Speaker 					
Music #1 					
Music #2 					
Female Speaker 					

How Spectrogram Inversion Works ?

1. รับ **Spectrogram** เป็นอินพุต
2. เติมข้อมูลเฟสเพื่อสร้างคลื่นเสียง (**Waveform Reconstruction**)
3. **Vocoder** แปลง **Spectrogram** เป็น คลื่นเสียง
4. สร้างไฟล์เสียงที่สามารถเล่นได้

What is Vocoder ?

Vocoder (Voice Encoder-Decoder) คือ โมเดลที่ใช้แปลง Spectrogram ให้เป็นคลื่นเสียง

- * ใช้ใน ขั้นตอนสุดท้ายของระบบ **TTS**

- * เติมข้อมูลเฟส (**Phase Information**) ที่ขาดหายไปจาก **Spectrogram**

- * ช่วยให้เสียงพูดที่สร้างขึ้น ฟังดูเป็นธรรมชาติขึ้น

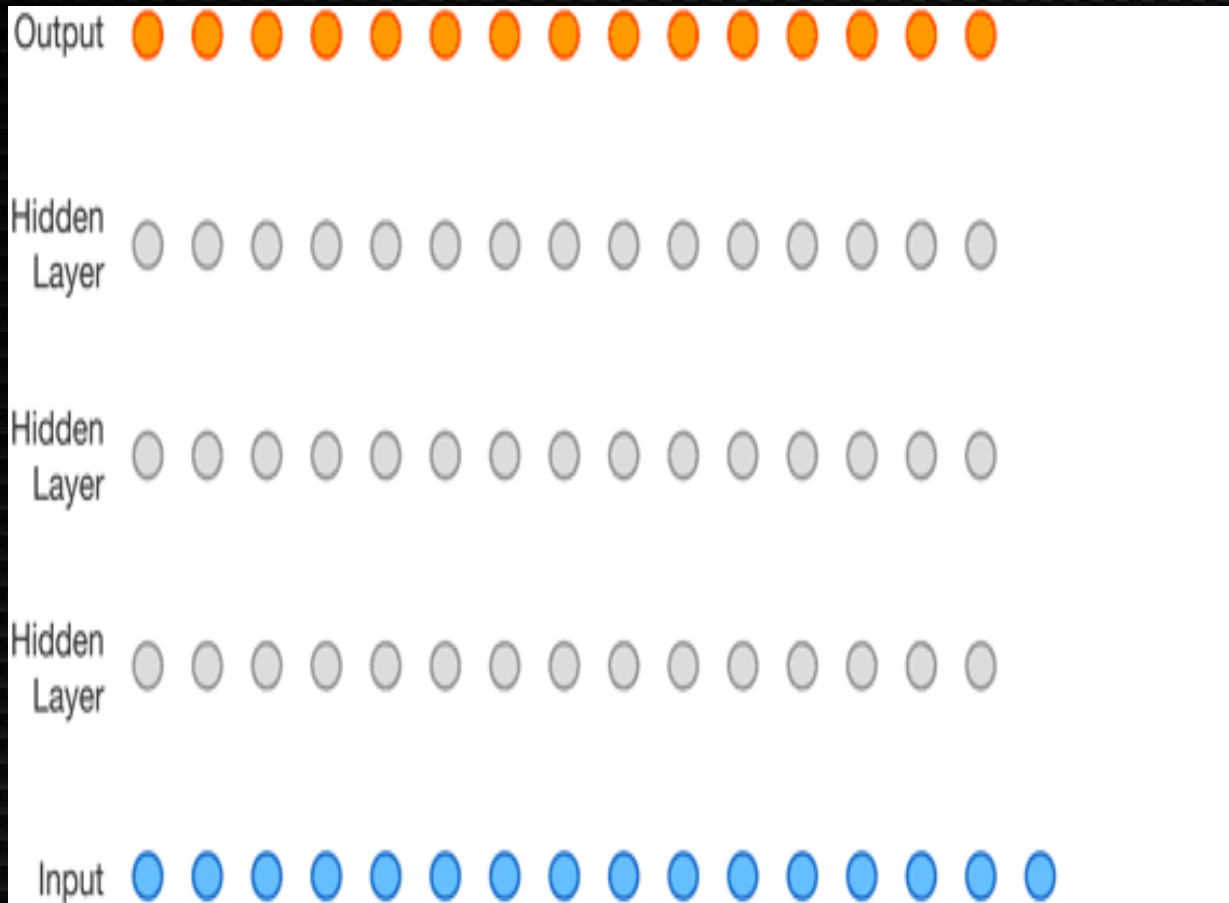
WaveNet

HiFi-GAN

WaveRNN

vocode

WaveNet



- * WaveNet คือ vocoder แบบ auto-regressive
- * ใช้ dilated causal CNN ทำนายเสียงที่ละตัวจากข้อมูลก่อนหน้า
- * คุณภาพเสียงสูงแต่ ช้ามาก
(RTF = 100 \rightarrow 1 วินาทีเสียงใช้ 100 วินาทีสร้าง)

ข้อดี

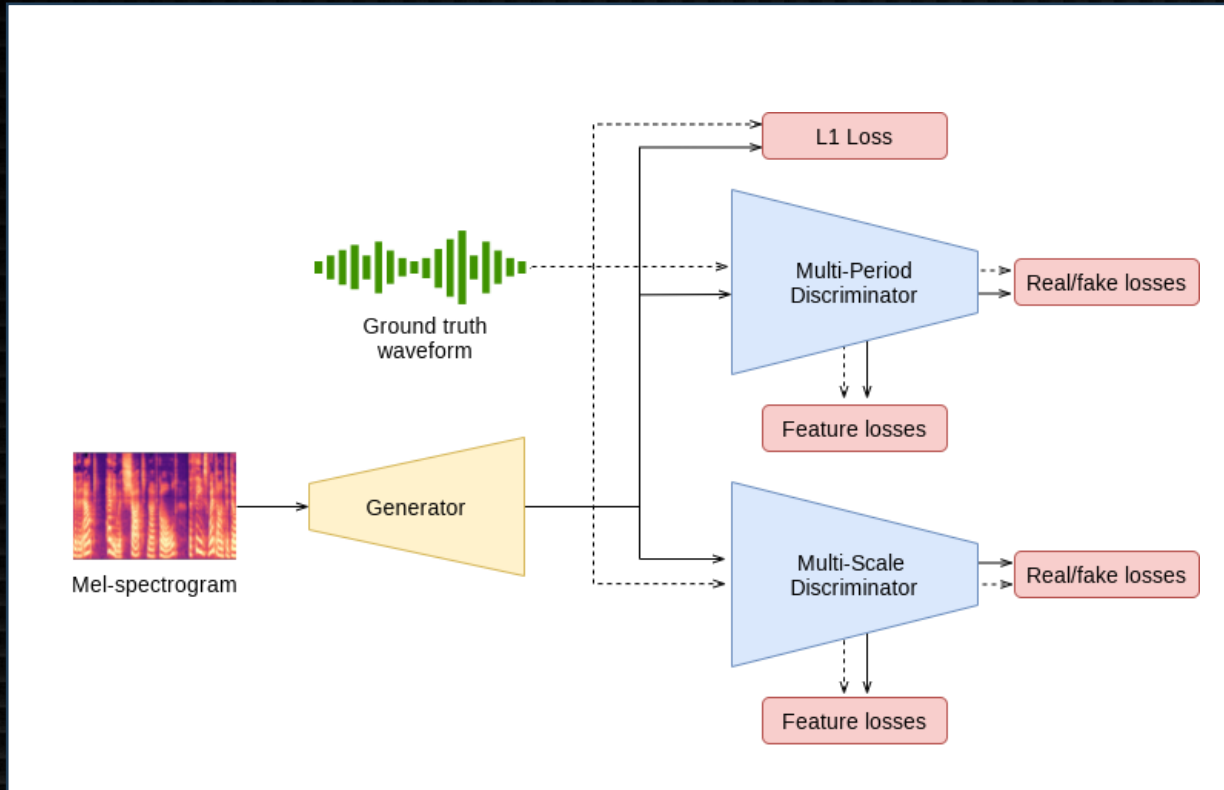
- WaveNet ให้เสียงสมจริง

ข้อจำกัด

- ช้ามาก ไม่เหมาะกับ real-time TTS

vocode

HiFi-GAN



* ที่เร็วกว่า WaveNet 10,000 เท่า

* GAN-based training แทน Auto- Regressive → เร็วขึ้น (RTF = 0.01)

* Scale & Period Discriminators เพื่อตรวจจับเสียงปลอม

ข้อดี

- เร็ว เหมาะสำหรับ real-time TTS
- ใช้ Feature Matching Loss เพื่อให้เสียงสมจริง

ข้อจำกัด

- ต้องใช้พลังการคำนวณสูงกว่าบางโมเดล

vocode

WaveRNN

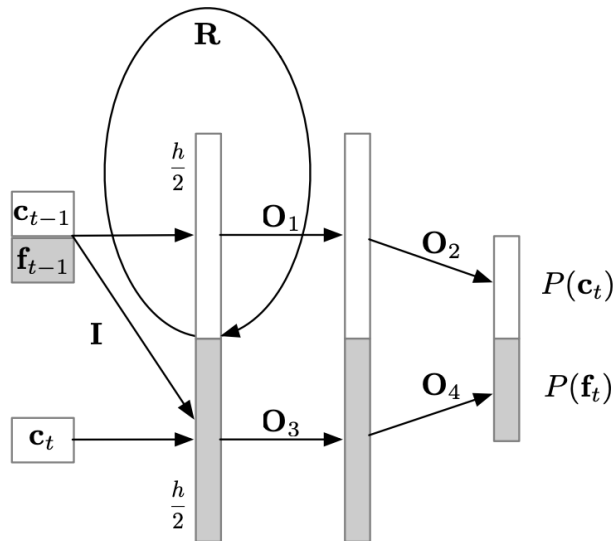


Figure 1. The architecture of the WaveRNN with the dual softmax layer. \mathbf{c} represents the coarse (high 8-bits) of the sample and \mathbf{f} represents the fine (low 8-bits) of the sample. The multiplication by \mathbf{R} happens for both the coarse and fine bits simultaneously, then output of the gates is evaluated for the coarse bits only and \mathbf{c}_t is sampled. Once \mathbf{c}_t has been sampled from $P(\mathbf{c}_t)$, the gates are evaluated for the fine bits and \mathbf{f}_t is sampled.

- * WaveRNN คือ vocoder ที่ใช้ RNN
- * ออกแบบให้ เล็กและใช้พลังงานต่ำ เหมาะกับ on-device

TTS

- * ใช้ Softmax แบบแยกส่วน, Subscaling, และ Sparse Training เพื่อลดภาระการคำนวณ

ข้อดี

- ประหยัดพลังงาน
- Inference เร็วกว่า WaveNet

ข้อจำกัด

- ยังช้ากว่า HiFi-GAN

Compare WaveNet, HiFi-GAN, and WaveRNN

คุณสมบัติ	WaveNet (Auto-Regressive)	HiFi-GAN (Non-Auto-Regressive)	WaveRNN (RNN-Based)
โครงสร้างโมเดล	Dilated Causal CNN	GAN-Based (Generator + Discriminators)	Recurrent Neural Network (RNN)
Auto-Regressive?	ใช่ (ช้าแต่แม่นยำ)	ไม่ใช่ (Parallel, เร็วกว่า)	ใช่ แต่ปรับปรุงให้เร็วขึ้น
Inference Speed	ช้ามาก (RTF ≈ 100)	เร็วมาก (RTF ≈ 0.01 , 10,000x เร็วกว่า WaveNet)	เร็วกว่า WaveNet แต่ช้ากว่า HiFi-GAN
คุณภาพเสียง	แม่นยำและสมจริง	เร็ว และ สมจริงที่สุด	ดีแต่ไม่เท่า GAN-based models
ขนาดโมเดล	ใหญ่ (ต้องใช้ GPU)	ขนาดกลาง (เหมาะกับ real-time TTS)	ขนาดเล็ก (เหมาะสำหรับ on-device TTS)
การใช้งานที่เหมาะสม	คุณภาพสูงแต่ต้องการพลังประมวลผลมาก	Real-Time TTS, ใช้งานจริงได้ดี	เหมาะกับอุปกรณ์พลังงานต่ำ (Edge/On-device)

Model Evaluation

- * ไม่มีตัวชี้วัดเชิงตัวเลขที่ชัดเจน สำหรับ TTS
- * คุณภาพมักถูกวัดโดย **ความคิดเห็นของมนุษย์ (Human Perception)** ผ่านแบบสำรวจ

วิธีการประเมินที่นิยม

Mean Opinion Score (MOS) : ให้ผู้ฟังให้คะแนนคุณภาพเสียงจาก 1 ถึง 5

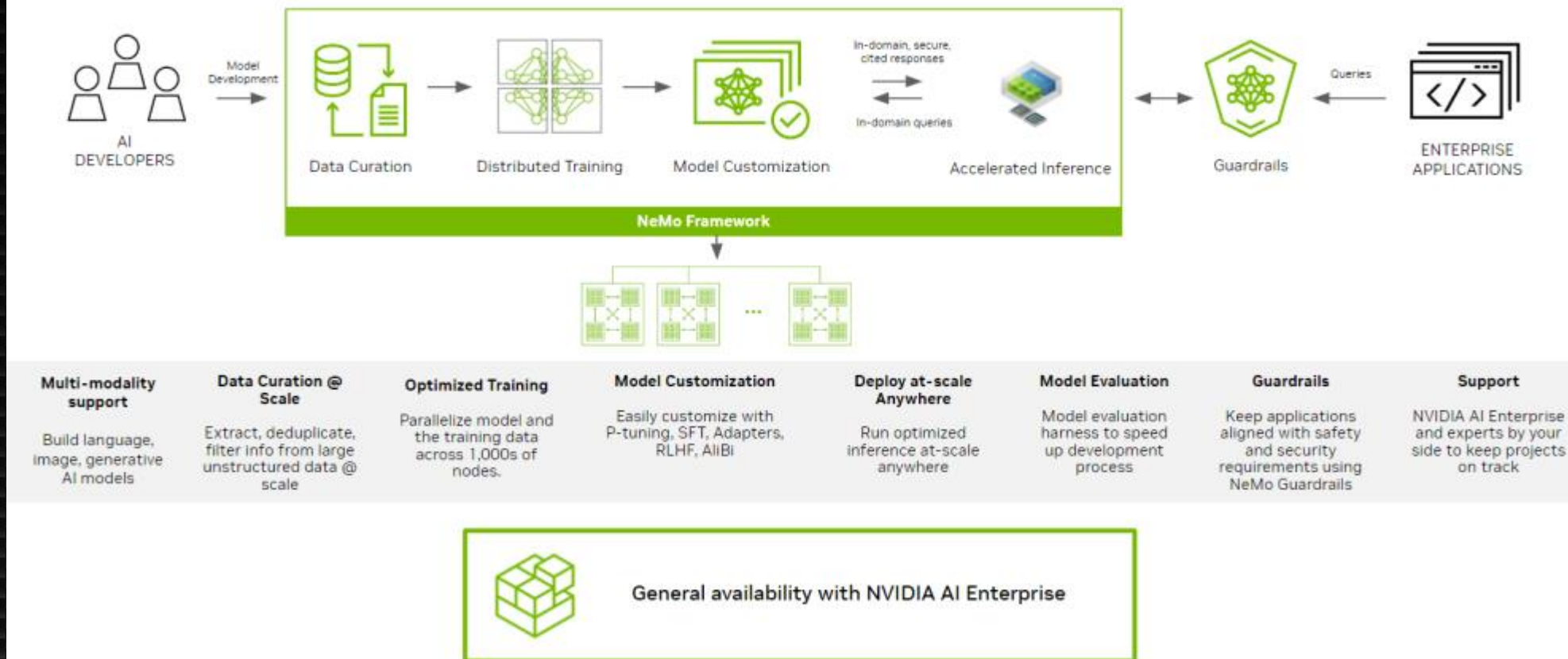
MUSHRA : ให้ผู้ใช้ฟัง เสียงต้นฉบับ (Ground Truth) แล้วให้คะแนน TTS

ที่สร้างขึ้น เปรียบเทียบกับต้นฉบับ

NVIDIA NeMo

NeMo Framework

End-to-end, cloud-native framework to build, customize, and deploy generative AI models



THANK
YOU