Wav2vec 2.0을 활용한 음성위조탐지 향상

중앙대학교
일반대학원 통계학과
ET LAB 강태인

중앙대학교

CONTENTS

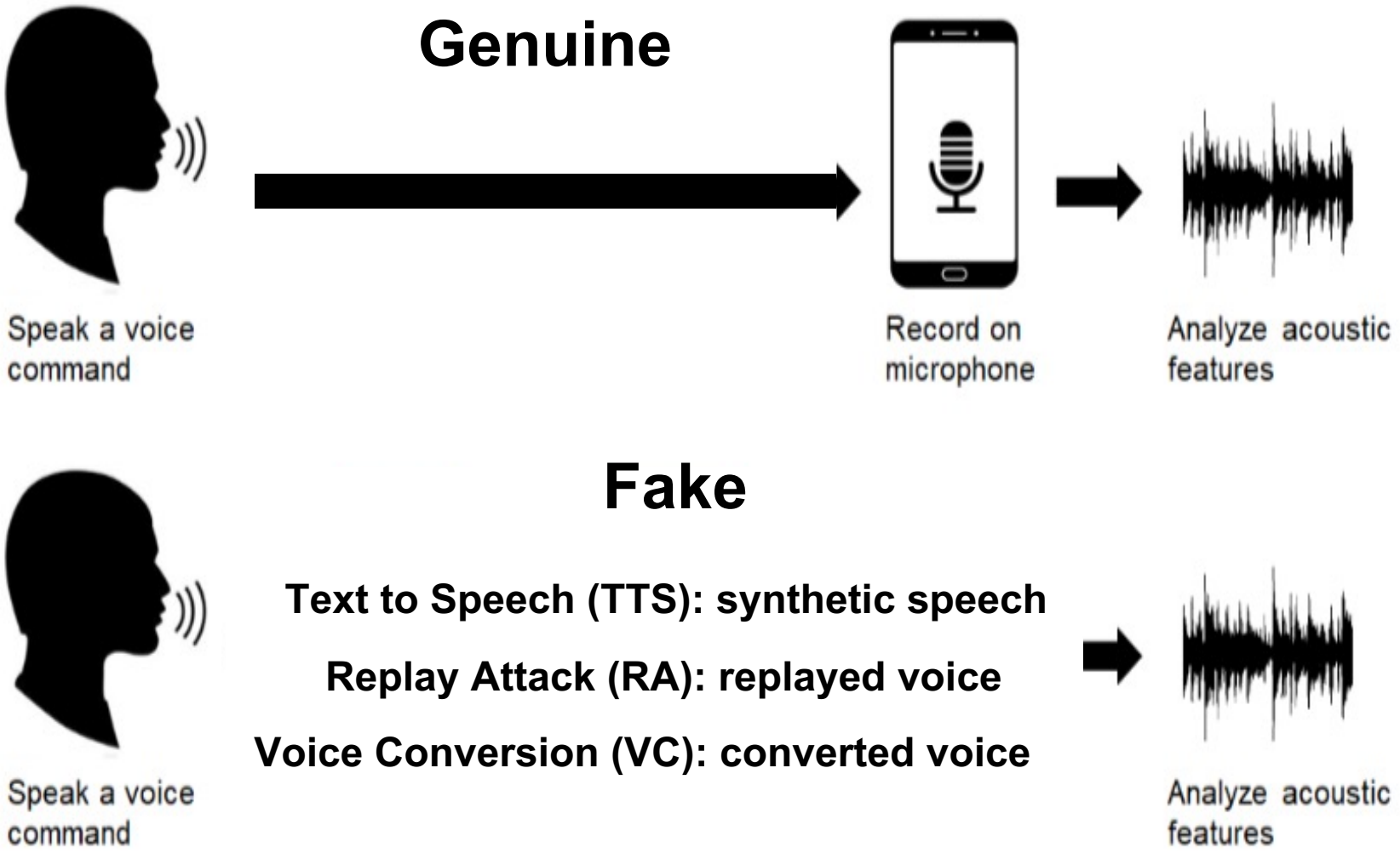
#01

Fake Audio Detection?

Fake Audio (Spoofing Audio)

- 실제 음성을 다양한 방법 등으로 위조하여 만든 음성을 실제 음성과 구별.

- 위조 방법에는 TTS, RA, VC 등이 있다.



출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

#01

Fake Audio Detection?

Why we need fake audio detection? (Spoofing detection)

- 녹음된 목소리를 이용해 금융 거래나 보안 인증에서 사용.

- 유명인의 목소리를 녹음한 후 가짜 음성으로 합성 후 특정 발언 조작.

- 지인 목소리의 보이스피싱.

The New York Times

Burger King 'O.K. Google' Ad Doesn't Seem O.K. With Google

Give this article

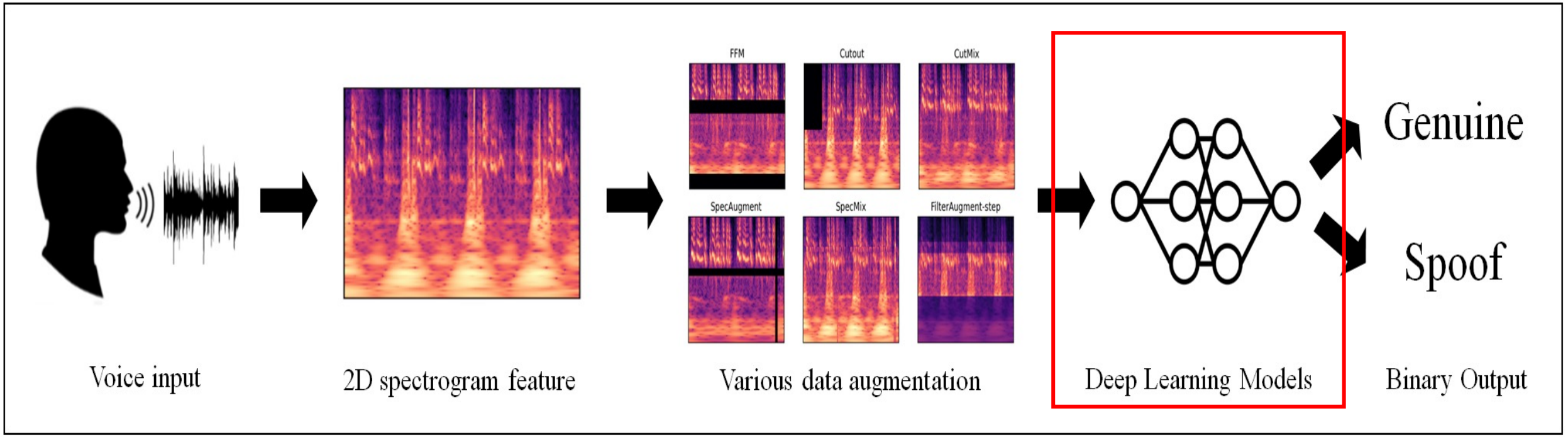
A video from the fast-food chain shows the the latest example of marketers entering the living room, with the advertisement intended to prompt voice-activated smart speakers from Google into describing its burgers.

출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

#01

Fake Audio Detection?

Traditional
Fake Audio Detection Model
(Spoofing Detection Model)

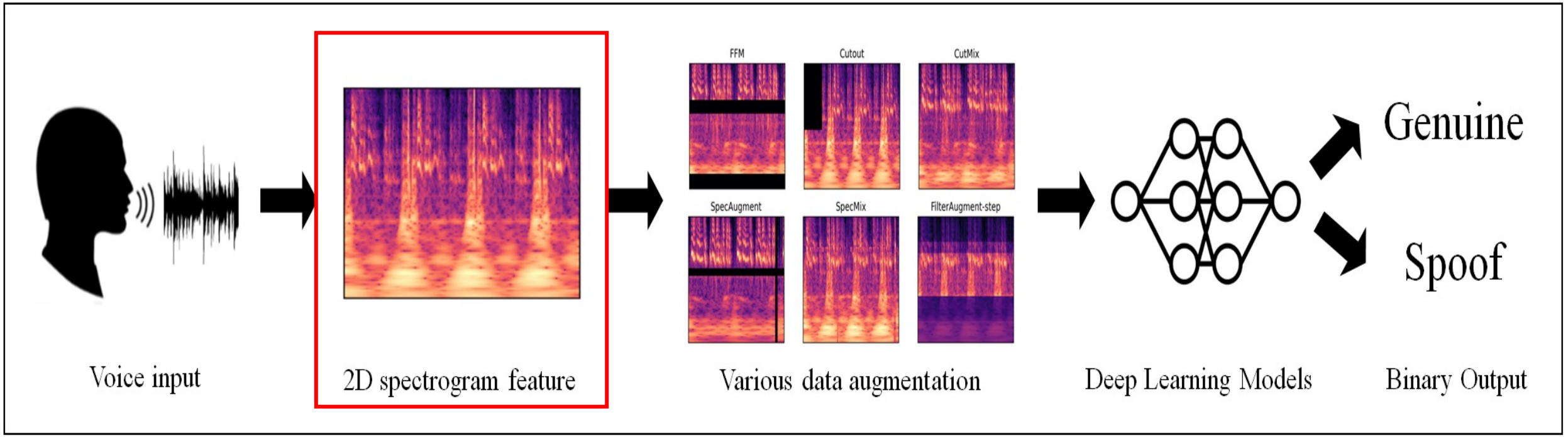


출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

#02

Feature Extraction

Traditional
2D-like Feature Extraction



출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

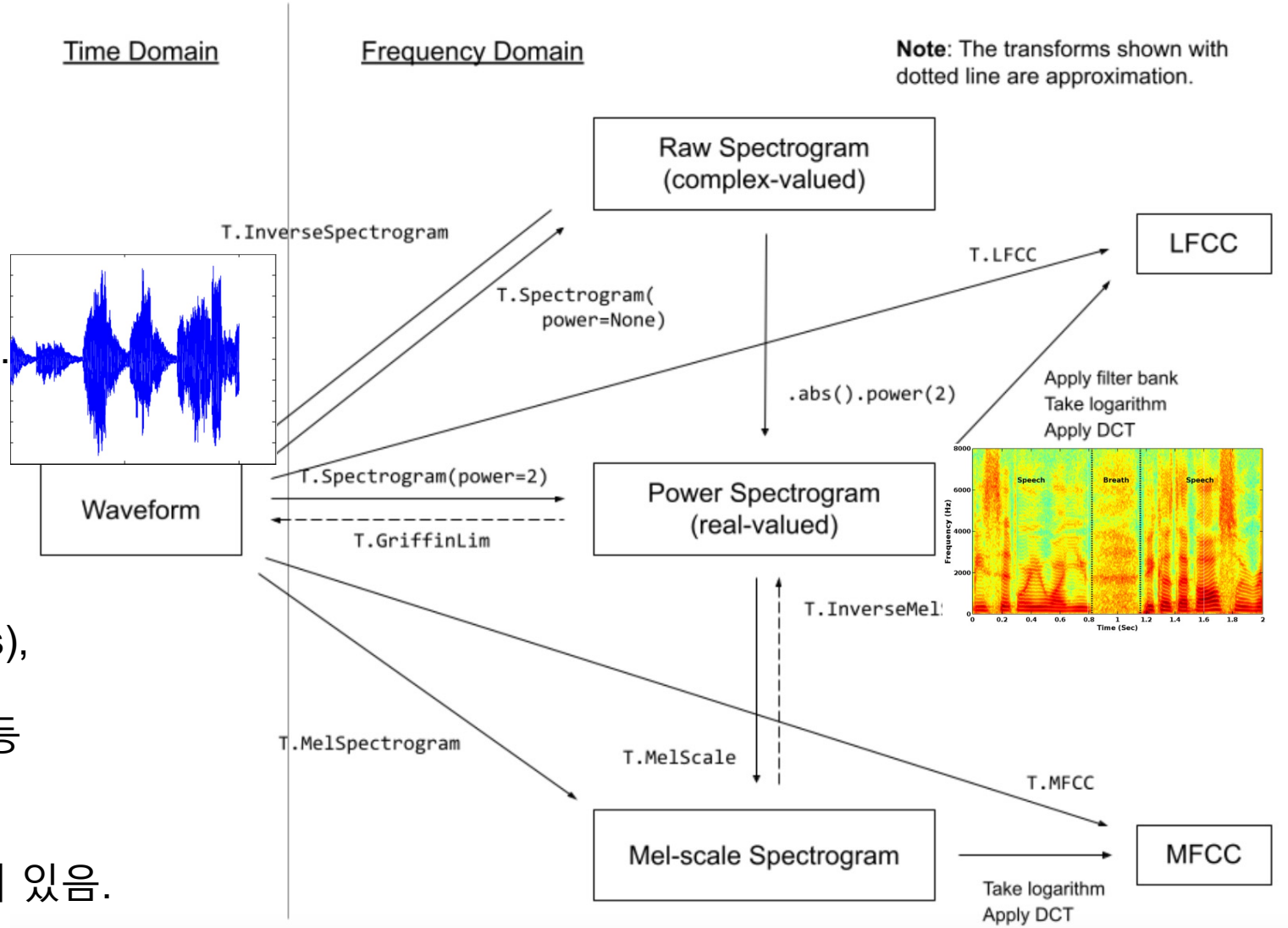
Feature Extraction

2D-like Handcrafted Feature Extraction

- FFT(Fast Fourier Transform)를 기반으로 여러 handcrafted feature를 생성할 수 있다.

Ex) Spectrogram,
Mel-Spectrogram,
LFCC
(Linear-frequency cepstrum coefficients),
MFCC
(Mel-frequency cepstrum coefficients) 등

- 이 외에도 CQT(Constant Q Transform), CWT(Continuous Wavelet Transform) 등이 있음.



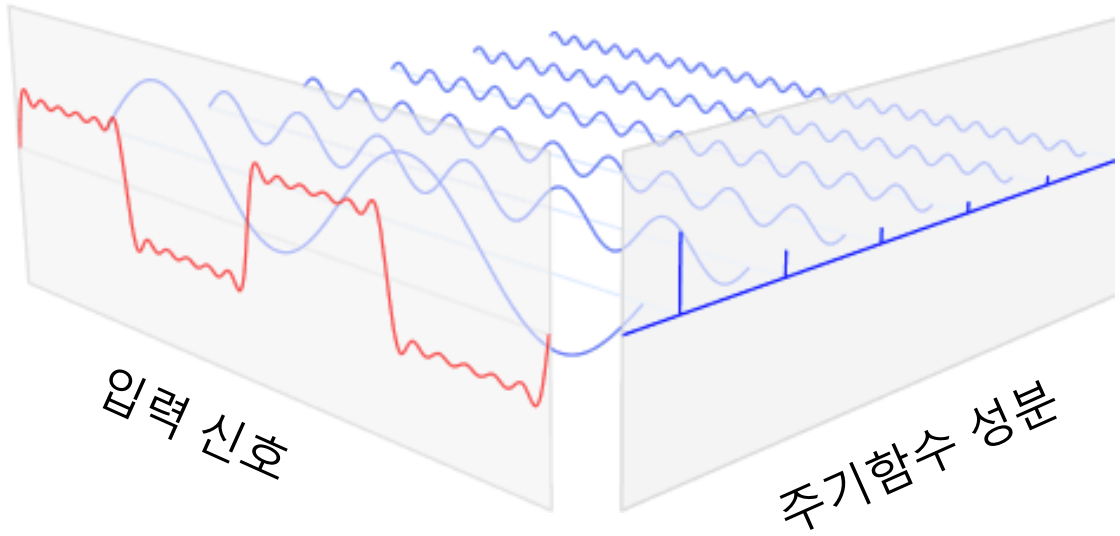
출처 : https://pytorch.org/audio/stable/transforms.html https://www.researchgate.net/figure/The-audio-signal-waveform_fig2_266648972, https://www.researchgate.net/figure/Spectrogram-of-a-speech-signal-with-breath-sound-marked-as-Breath-whose-bounds-are_fig1_319081627

#02

Feature Extraction

Fast Fourier Transform

- 신호 데이터는 다양한 주기의 주기함수의 합으로 표현 가능하기 때문에 입력 신호를 다양한 주기의 주기함수의 세기로 분해하여 변환.



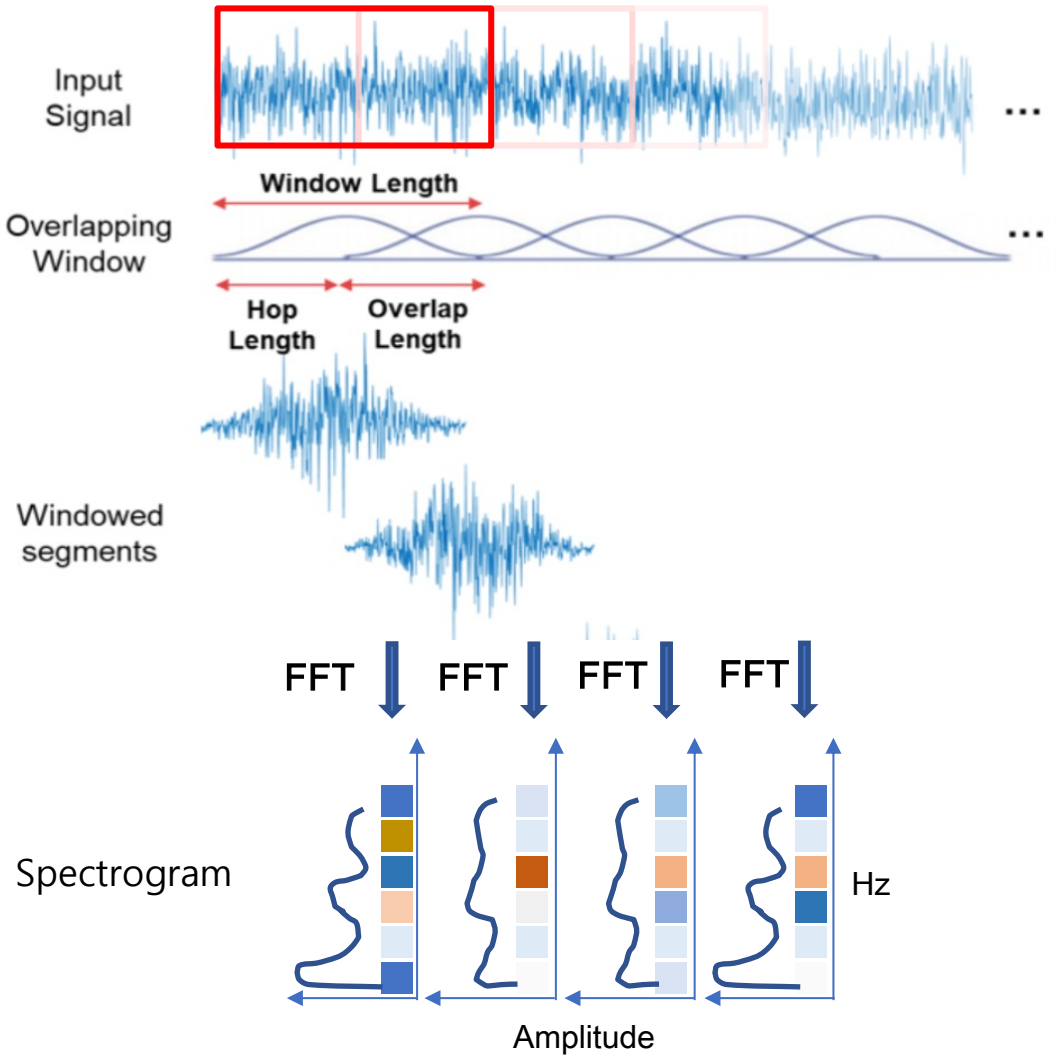
출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

Feature Extraction

Short Time Fourier Transform

- 일정한 time step마다 FFT를 수행하게 되면 시간 정보를 반영 가능.

- Time step마다의 frequency 정보를 얻을 수 있다.



출처 : 중앙대학교 통계학과 ET LAB 한소율 박사 발표자료

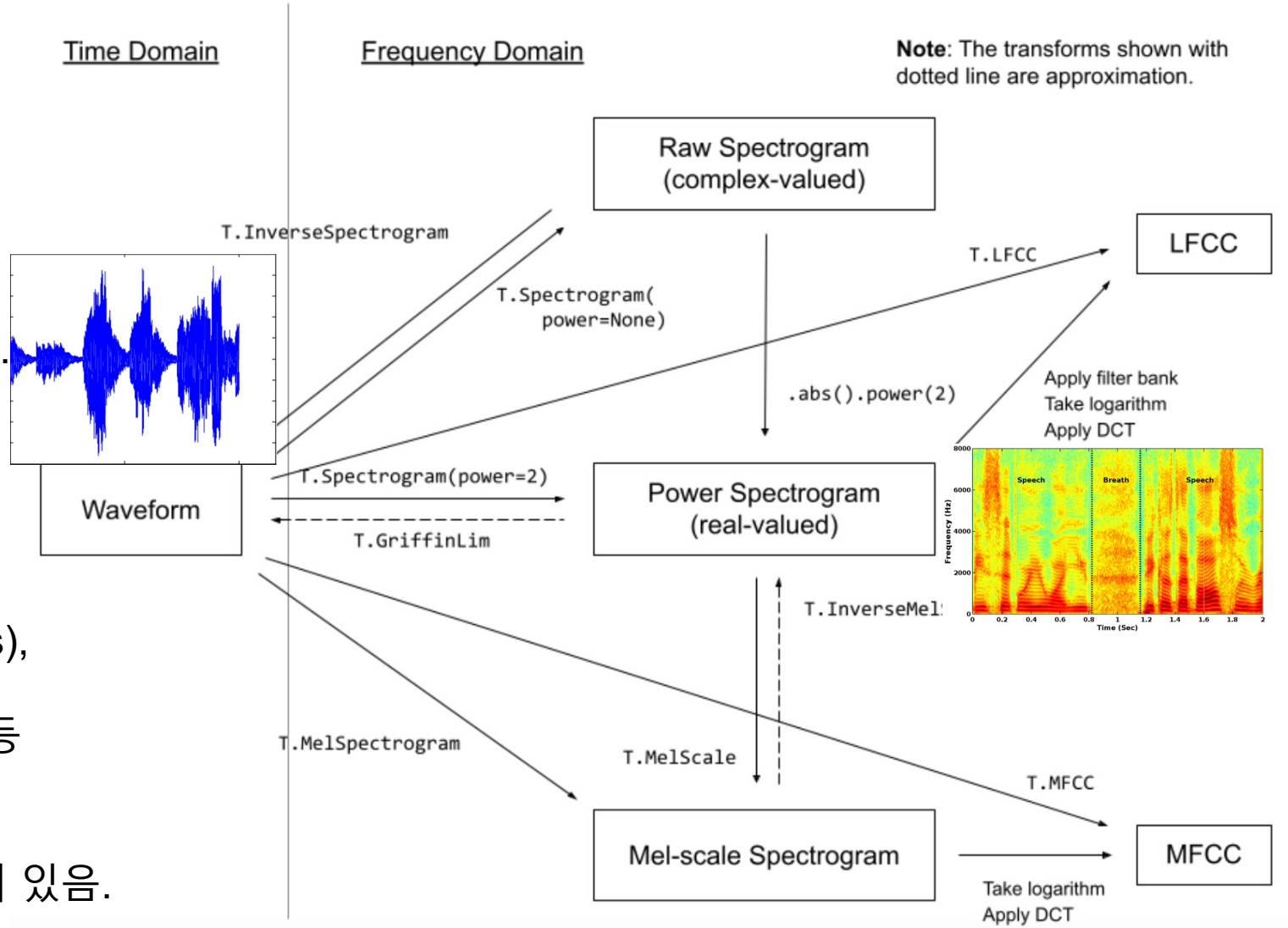
Feature Extraction

2D-like Handcrafted Feature Extraction

- FFT(Fast Fourier Transform)를 기반으로 여러 handcrafted feature를 생성할 수 있다.

Ex) Spectrogram,
Mel-Spectrogram,
LFCC
(Linear-frequency cepstrum coefficients),
MFCC
(Mel-frequency cepstrum coefficients) 등

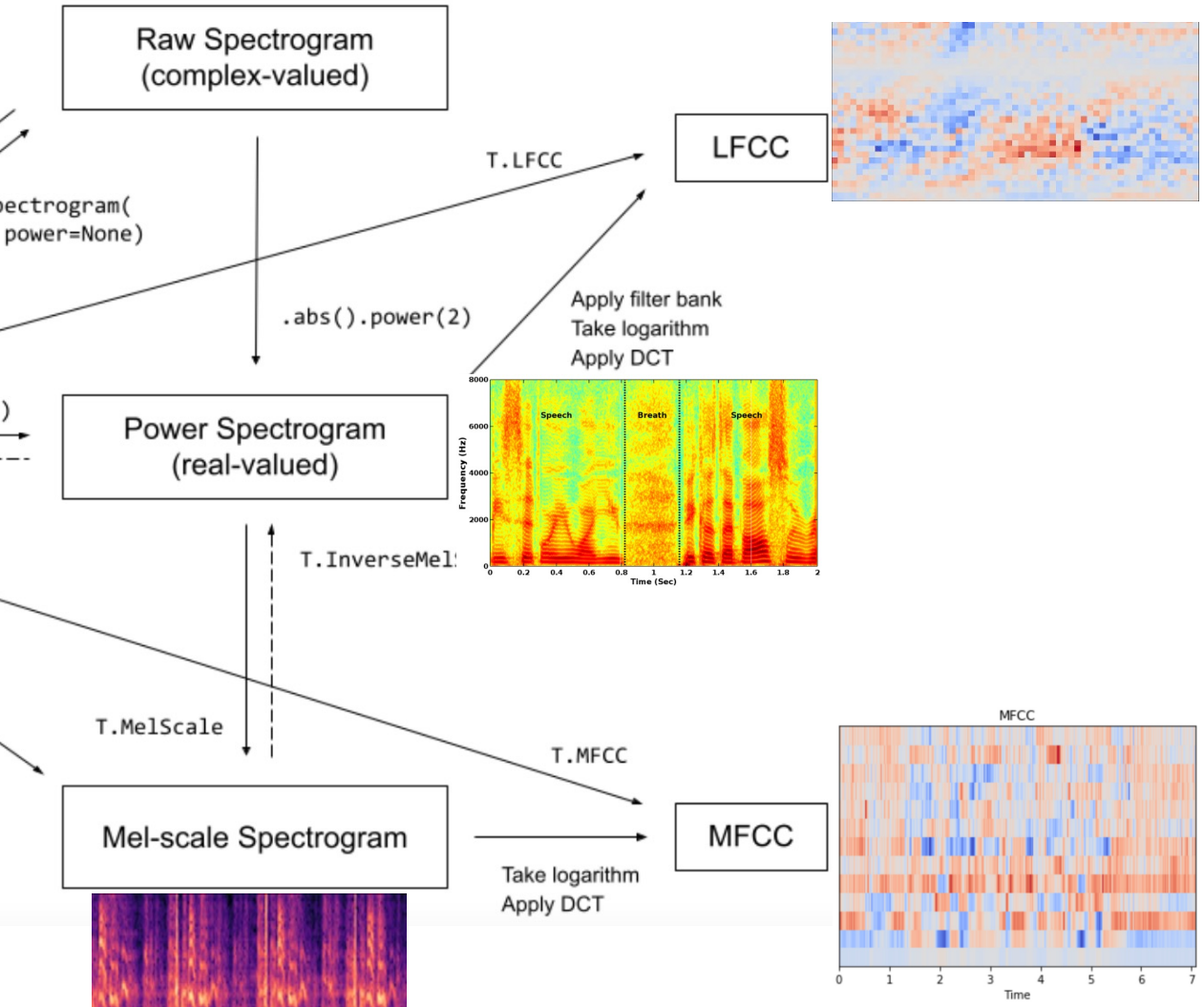
- 이 외에도 CQT(Constant Q Transform), CWT(Continuous Wavelet Transform) 등이 있음.



출처 : https://pytorch.org/audio/stable/transforms.html https://www.researchgate.net/figure/The-audio-signal-waveform_fig2_266648972, https://www.researchgate.net/figure/Spectrogram-of-a-speech-signal-with-breath-sound-marked-as-Breath-whose-bounds-are_fig1_319081627

#02

Feature Extraction



Advancement of the model

- MFCC, LFCC → Machine Learning Model
  GMM(Gausian Mixture Model)…

- STFT, Mel-Spec, CQT → 1D CNN, 2D CNN
  LCNN, Non-OFD…

출처 : https://www.mdpi.com/1424-8220/22/19/7443. https://blog.naver.com/PostView.nhn?blogId=gaechuni&logNo=221651532336

Feature Extraction From Raw Waveform

- SincNet에서 Band-pass filter의 일종인 Sinc function을 사용한 SincNet filter를 제시.

- SincNet filter는 data로부터 low and high cufoff frequencies를 학습.

- 2021년 부터 이를 이용한 Graph Neural Networks(GNNs) 기반 모델(RawGAT-ST, AASIST 등)이 높은 성능을 기록.

- Fully end-to-end Model이 시작.

→ 고도화된, 학습하는 feature가 성능향상의 요인인가 ?

Feature Extraction – wav2vec 2.0

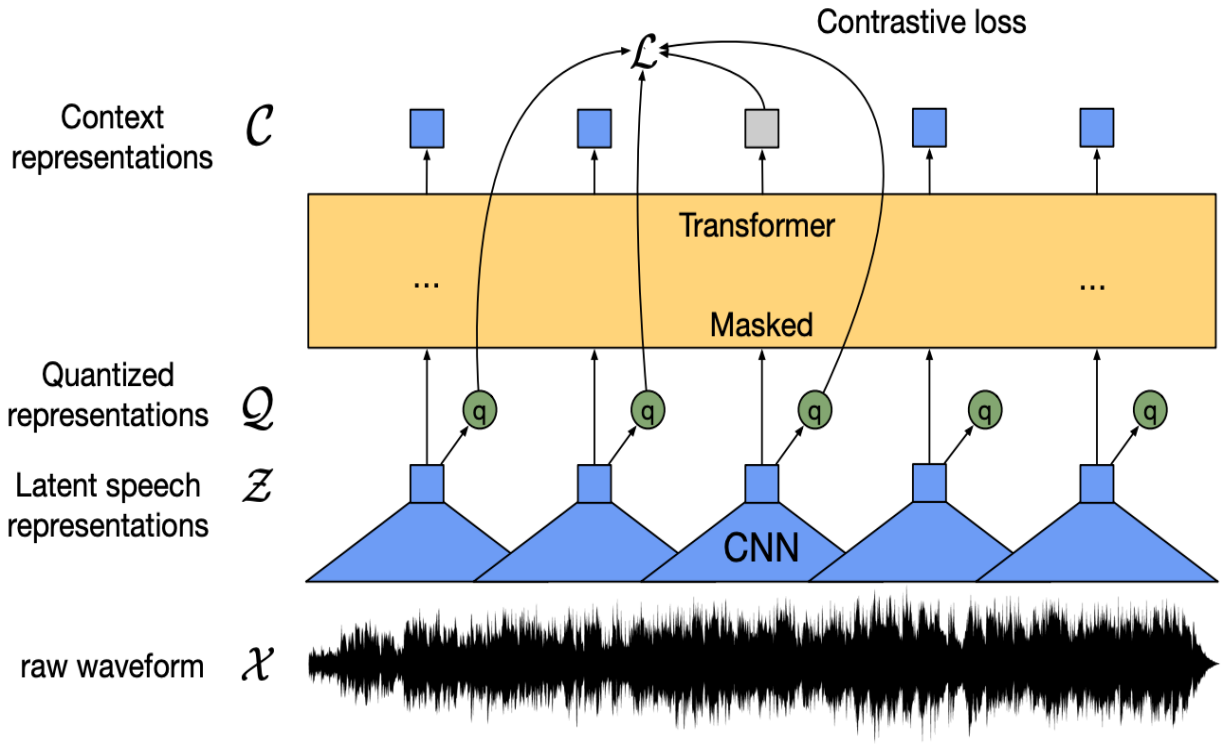
Wav2vec 2.0

- 자기 지도 학습 모델로서, 음성 표현 학습에 집중.

- 여러 Downstream task에서 뛰어난 성능을 보여줌.

- 여러 버전의 사전학습 모델을 쉽게 접근 가능.

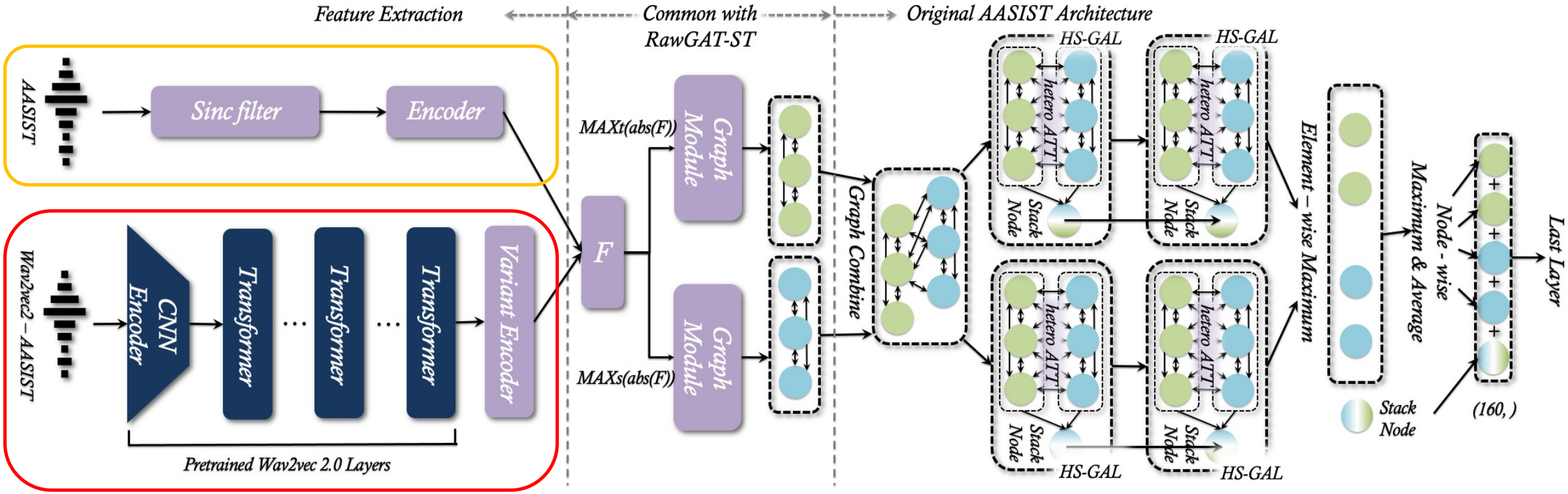
→ 음성표현을 많이 학습했고, 여러 downstream task에서도 좋았으니,,,,
Spoofing Detection에서도 어쩌면,,?



출처 : https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

#03

Proposed Feature Extraction



Wav2vec 2.0 As Feature Extractor

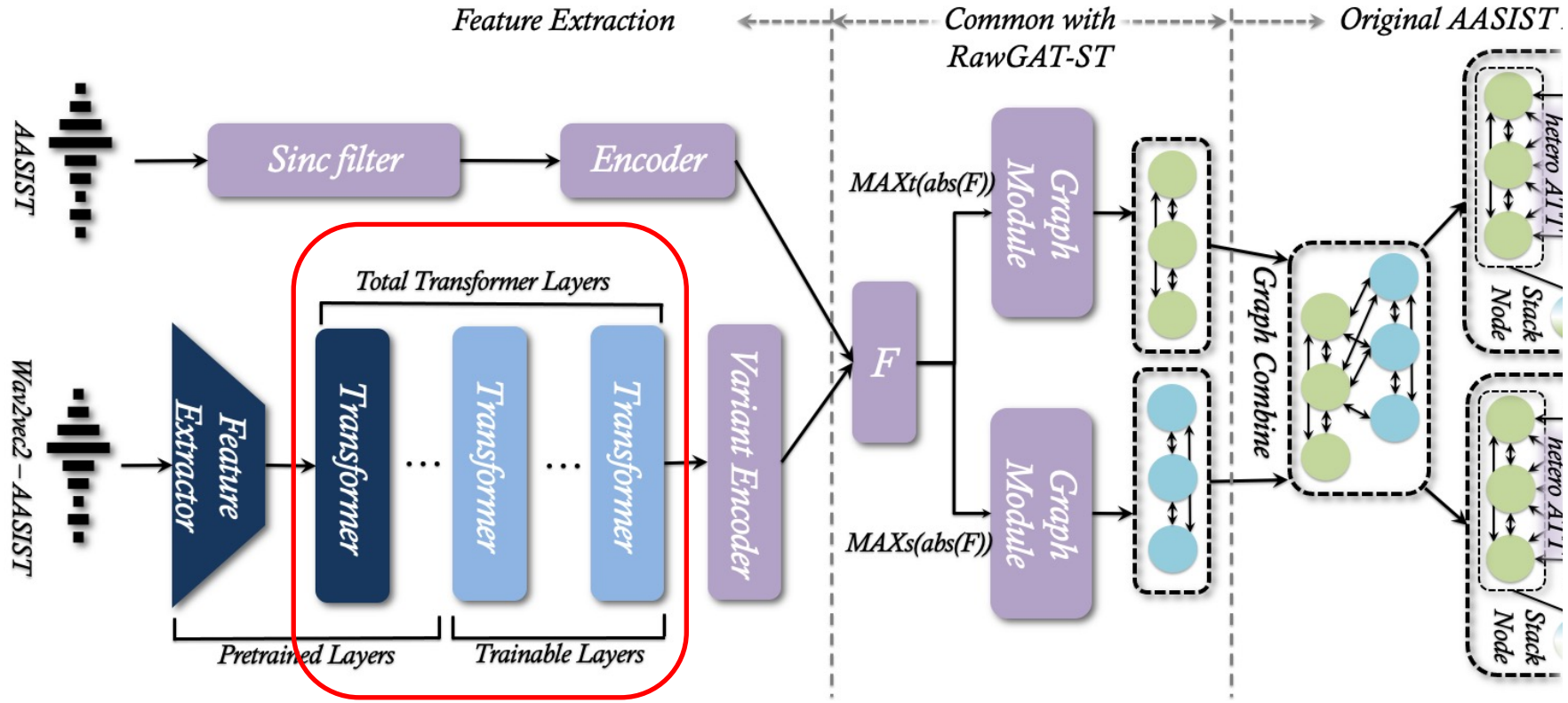
- Parameter를 모두 freeze시키고 AASIST를 붙여보니 성능이 매우 안좋음.

→ 데이터에 adaptation 시키면?

→ '마지막' Transformer layer의 output이 정말 spoofing에 유리한 representation일까?

#03

Proposed Feature Extraction



Wav2vec 2.0 As Feature Extractor

<Hyperparameter setting>

- #TTL(Total Transformer layers) : 총 몇 개의 transformer layer를 사용 할 것인가.

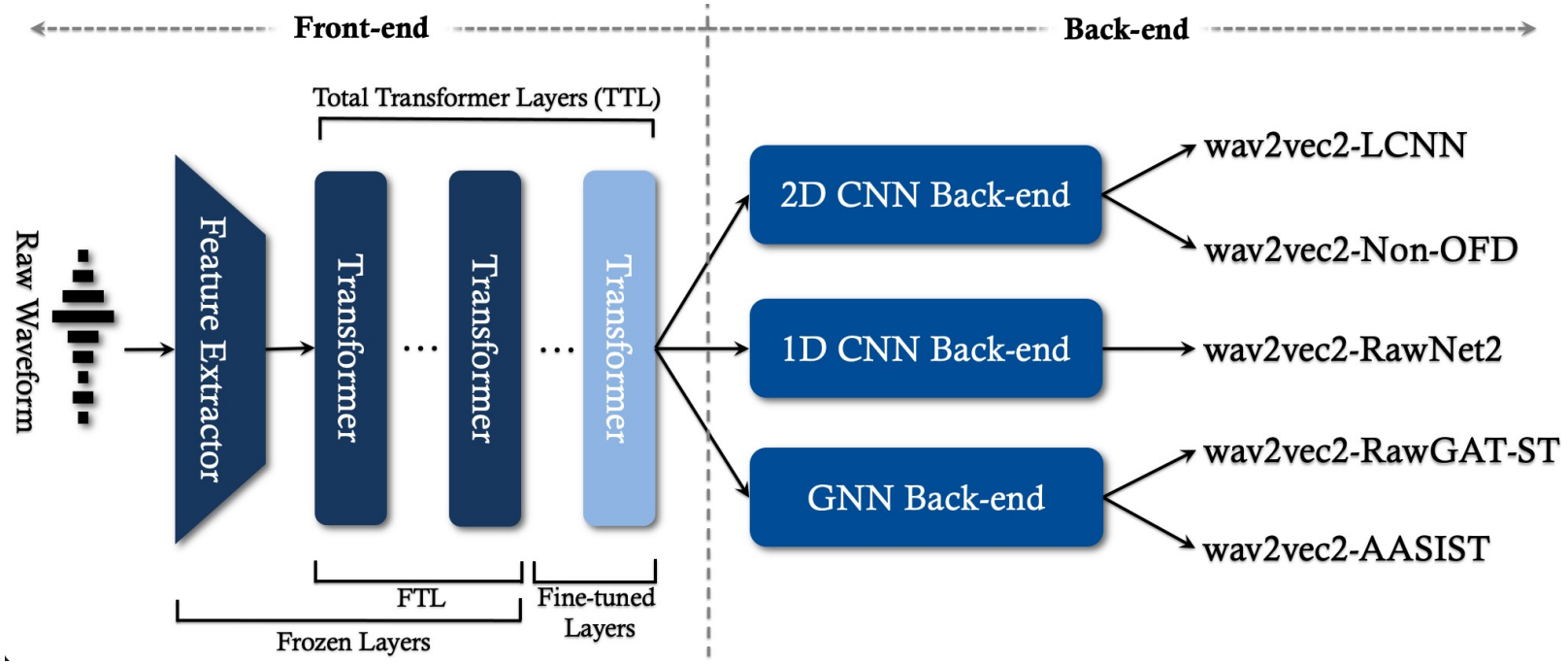
- #FTL(Frozen Transformer layers) : 몇 개의 transformer layer까지 freeze시킬 것인가.

Experiments & Results

Experimental Design

<Experimental design>

→ 최적의 #TTL, #FTL 조합은?
(Spoofing detection에 맞는 parameter space가 있지 않을까?)



→ 사전학습 버전에 따른 성능 차이가 있을까?

→ 기존 feature와 비교하면?

→ 다른 spoofing detection model들과 비교하면?

Experiments & Results

→ 최적의 #TTL, #FTL 조합은?

- ASVSpoof2019 LA Dataset 사용.

- EER(Equal Error Rate) metric사용.

- Default model로 AASIST를 사용.
  (AASIST의 성능은 **EER 0.83%**)

- Default version으로 XLS-R(1B) 를 사용.

- 48개의 layer에서 3개 단위로 모든 조합을 실험.

#TTL	#FTL								
	0	**3**	**6**	**9**	12	15	18	33	48
3	8.96	38.23	-	-	-	-	-	-	-
6	7.71	6.54	42.82	-	-	-	-	-	-
9	1.38	0.88	0.98	17.84	-	-	-	-	-
12	**0.77**	**0.41**	**0.57**	**0.22**	17.85	-	-	-	-
15	0.87	1.04	1.35	**0.80**	1.43	41.00	-	-	-
18	**0.63**	**0.40**	**0.61**	2.14	2.39	6.68	26.01	-	-
33	2.41	2.01	3.94	3.57	4.26	4.36	4.54	37.51	-
48	2.37	5.75	0.65	2.83	7.52	5.13	9.72	4.53	41.09

Table 1. EER(%) results on XLS-R(1B) and AASIST combination system.

<Result>

- 최적의 #TTL space = {12, 15, 18}

- 최적의 #FTL space = {0, 3, 6, 9, 12, 15}

Experiments & Results

→ 사전학습 버전에 따른 차이?

- 실험한 버전은 XLSR-53, XLS-R(0.3B), XLS-R(1B), XLS-R(2B)이다.

- Default model로 AASIST를 사용.

- 앞선 실험의 결과로 parameter space를 지정.

version	#TTL	#FTL	min t-DCF	EER(%)
XLSR-53	12	3	0.0083	0.26
XLS-R(0.3B)	15	6	0.0093	0.29
XLS-R(1B)	12	9	**0.0063**	**0.22**
XLS-R(2B)	18	3	0.0098	0.30

Table 2. Results on various wav2vec 2.0 front-ends.

- min t-DCF(Detection Cost Function), EER을 metric으로 사용.

<Result>

- XLS-R(1B)뿐만 아니라 optimal한 parameter space가 다른 사전학습 버전에서도 최고 성능에 크게 영향을 주지 않는다.

Experiments & Results

→ 기존 feature와의 차이

- 실험한 model은 2D CNN 기반 LCNN, Non-OFD,
  1D CNN 기반 RawNet2,
  GNN 기반 RawGAT-ST, AASIST이다.

- 비교 feature는 STFT, CQT, SincNet filter 등이 있다.

- Default 버전으로 XLS-R(1B)를 사용.

Model	Front-end	min t-DCF	EER(%)
LCNN [2]	STFT	0.1028	4.53
	XLS-R(1B) (12/9)	**0.0320**	**1.02**
Non-OFD [13]	CQT	-	1.35
	XLS-R(1B) (15/9)	0.0111	**0.41**
RawNet2 [3]	SincNet filter	0.1301	5.64
	XLS-R(1B) (12/6)	**0.0032**	**0.12**
RawGAT-ST [5]	SincNet filter	0.0335	1.06
	XLS-R(1B) (18/12)	**0.0048**	**0.24**
AASIST [6]	SincNet filter	0.0275	0.83
	XLS-R(1B) (12/9)	**0.0063**	**0.22**

Table 3. Results on various spoofing detection back-ends.

<Result>

- 모든 Network에서 wav2vec 2.0 front-end가 높은 성능
  향상을 시켰다. 특히, wav2vec 2.0 + RawNet2 는 State-Of-The-Art의 성능.

- Optimal hyperparameter space가 유효했다.

Experiments & Results

→ 다른 Spoofing Model과 비교

- Default 버전으로 XLS-R(1B)를 사용.

<Result>

- 다른 모델과 비교했을 때 최고 성능 달성.

- wav2vec 2.0+VIB, wav2vec 2.0+ASP 보다 잘 잘동하는 원인은 정확히 확인하지 못 함.

Model	Front-end	min t-DCF	EER(%)
RawNet2 [3]	SincNet filter	0.1301	5.64
GAT-T [23]	LFB	0.0894	4.71
LCNN [2]	STFT	0.1028	4.53
GMM [24]	LFCC	0.0904	3.50
RW-ResNet [25]	Raw Waveform	0.0817	2.98
LCNN-LSTM-sum [26]	LFCC	0.0524	1.92
Non-OFD [13]	CQT	-	1.35
RawGAT-ST [5]	SincNet filter	0.0335	1.06
AASIST [6]	SincNet filter	0.0275	0.83
GCN based model [7]	LFB	0.0166	0.58
wav2vec 2.0 + VIB [17]	BASE [8]	0.0107	0.40
wav2vec 2.0 + ASP [12]	XLSR-53	-	0.31
wav2vec 2.0 + AASIST (Ours)	**XLS-R(1B)**	**0.0063**	**0.22**
wav2vec 2.0 + RawNet2 (Ours)	**XLS-R(1B)**	**0.0032**	**0.12**

Table 4. Comparison with recently established spoofing detection systems.

Conclusion

- Spoofing detection에서 고도화된 feature encoder를 사용하는 것이 대체로 유리하다.

- Wav2vec 2.0을 활용할 때에 data adaptation을 위해 fine-tuning하는 것이 좋다.

- Downstream task에 맞는 #TTL, #FTL을 찾아주는 것이 좋다.

THANK YOU