

Advances in Data-Efficient Multilingual and Crosslingual Speech Recognition Powered by LLMs

Saierdaer Yusuyin¹, Hexin Liu², Eng Siong Chng², Zhijian Ou³

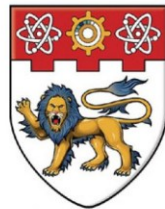
¹Xinjiang University (XJU)

²Nanyang Technological University (NTU)

³Tsinghua University (THU)



新疆大学
Xinjiang University



NANYANG
TECHNOLOGICAL
UNIVERSITY



清华大学
Tsinghua University

Introduction

- Neural ASR systems have achieved significant progress, which are, however, **data-hungry**.
- An important **challenge** for the speech community is how to develop ASR systems to new languages **efficiently** with limited resources.
- Multilingual and crosslingual ASR (**MCL-ASR**)
 - multilingual pretraining and crosslingual finetuning.
 - Large acoustic models (LAMs)/Speech encoders: Wav2Vec, Whisper, WavLM and Whistle.
 - Large language models (LLMs): BERT, GPT.
- Cooperation between acoustic and language models is important for ASR systems and has long been studied.
 - A recent trend of ASR research: combining LAMs and LLMs -- **LLM-ASR**.
- **Speech–language interfaces** define two broad approaches to LLM-ASR: continuous-interface and discrete-interface.

This tutorial provides a **systematic introduction to LLM-ASR**, organized around a taxonomy of speech–language interfaces, with particular emphasis on data-efficient MCL-ASR.

Content

I. Basics of Speech Recognition and LLMs (50 min) Zhijian Ou

1. Modern speech recognition models
2. Data-efficiency in Multilingual and Crosslingual Speech Recognition
3. Introduction to LLMs
4. Overview and taxonomy of LLM-ASR

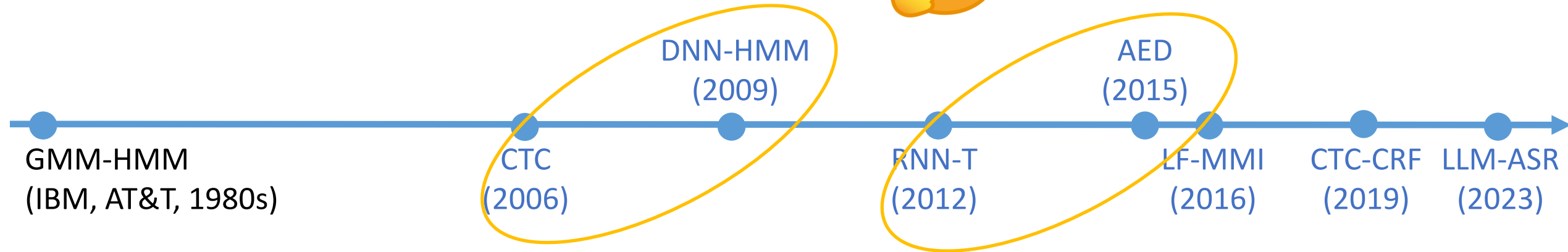
II. Continuous-interface based approach (50 min) Hexin Liu, Eng Siong Chng

III. Discrete-interface based approach (50 min) Zhijian Ou

IV. Open questions and future directions (20 min) Eng Siong Chng

After each section, 10 minutes for QA and possible break.

Brief ASR History



- [CTC] Graves, et al., “Connectionist Temporal Classification: Labelling unsegmented sequence data with RNNs”, ICML 2006.
- [DNN-HMM] A. Mohamed, et al., “Deep belief networks for phone recognition”, NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009.
- [RNNT] A. Graves, “Sequence transduction with recurrent neural networks”, ICML 2012 Workshop on Representation Learning.
- [AED] D. Bahdanau, et al., “Neural machine translation by jointly learning to align and translate”, ICLR 2015.
- [LF-MMI] D. Povey, et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI", INTERSPEECH 2016.
- [CTC-CRF] Xiang&Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.
- [LLM-ASR] Wu et al. “On decoder-only architecture for speech-to-text and large language model integration”, ASRU 2023.

Modern/E2E ASR: Basics still very relevant for us to understand LLM-ASR and go beyond

ASR is a *sequence discriminative* problem

- For acoustic observations $\mathbf{x} \triangleq x_1, \dots, x_T$, find the most likely labels $\mathbf{y} \triangleq y_1, \dots, y_L$

Two key problems in E2E ASR:

- How to obtain $p(\mathbf{y} | \mathbf{x})$
- How to handle alignment, since $L \neq T$

- Need a differentiable *sequence-level* loss of mapping acoustic sequence $\mathbf{y}$ to label sequence $\mathbf{x}$

- Explicitly:** introduce hidden state sequence $\boldsymbol{\pi}$, as in Connectionist Temporal Classification (CTC), RNN Transducer (RNNT), CRF
- Implicitly:** as in Attention based Encoder-Decoder (AED)

Labels

$L \neq T$

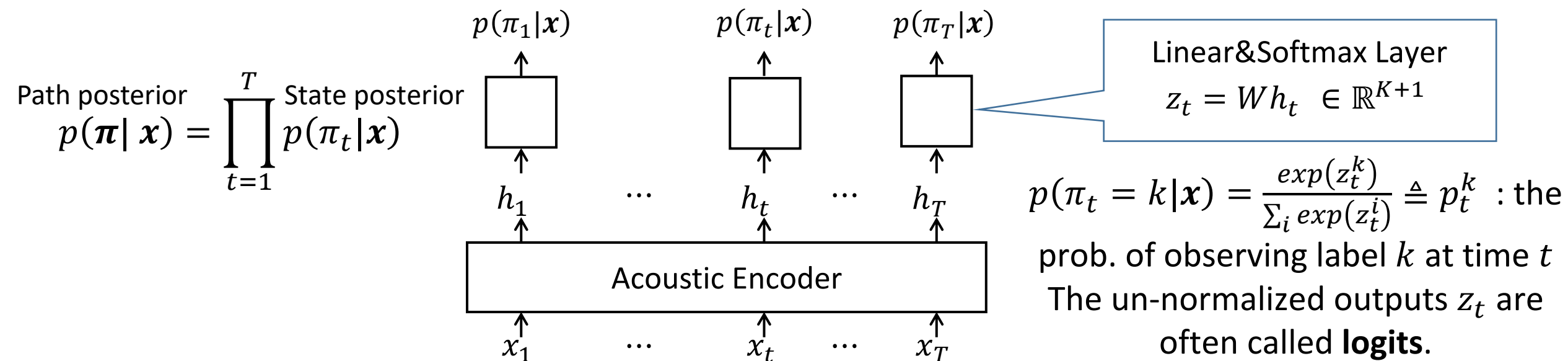
$\mathbf{y}$								
$\parallel$							π_7	π_8
y_1						π_6		
$\vdots$			π_3	π_4	π_5			
y_L	π_1	π_2						

Observations $\mathbf{x} = x_1 \dots x_T$

Example of explicit alignment

CTC: introducing **blank** symbol

- Motivation: training $p(\mathbf{y} | \mathbf{x})$ without the need for frame-level alignments between the acoustics $\mathbf{x}$ and the transcripts $\mathbf{y}$
 - Introduce a state sequence $\boldsymbol{\pi} \triangleq \pi_1, \dots, \pi_T$, where $\pi_t \in \text{the-alphabet-of-labels} \cup \text{}$



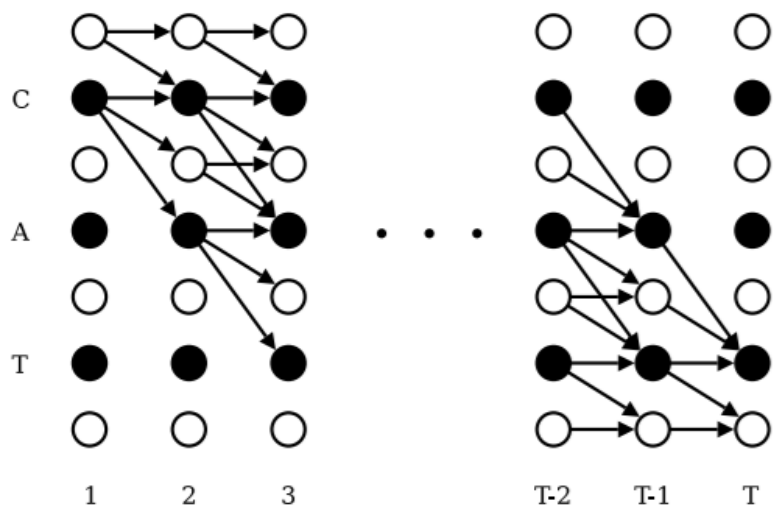
CTC topology

- **State topology** refers to the state transition structure in π , which basically determines the mapping $\mathcal{B}_{CTC}$ from π to $\mathbf{y}$

CTC topology : a mapping $\mathcal{B}_{CTC}$ maps π to $\mathbf{y}$ by

1. reducing repetitive symbols to a single symbol;
2. removing all blank symbols.

$$\mathcal{B}(-CC - -AA - T -) = CAT$$



Path posterior

$$p(\pi|\mathbf{x}) = \prod_{t=1}^T p(\pi_t|\mathbf{x})$$

Label-seq posterior / CTC likelihood

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\pi: \mathcal{B}_{CTC}(\pi)=\mathbf{y}} p(\pi|\mathbf{x})$$

Summing over all possible paths, which map to $\mathbf{y}$

B	B	c	B	B	a	a	B	B	t
B	c	c	B	a	B	B	B	B	t
...									
B	c	B	B	a	B	B	t	t	B

CTC: Label units

- For WERs in apple-to-apple comparisons
 - mono-phone **clearly better** mono-char, over WSJ-80h, Switchboard-300h, Librispeech-960h [Xiang&Ou, 2019]
 - For low degree of grapheme-phoneme correspondence (e.g., English), wordpiece **slightly worse** than mono-phone; For high degree (e.g., German), **equally strong** [Zheng, et al., 2021]
- **Longer span, more training data needed**
 - Word-level CTC targets, trained on 3,400 hours of speech [Li et al., 2018]

Basic Units of Labels	Label Sequence
phoneme	DH AE1 T N IY1 DH ER0 AH1 V DH EH1 M HH AE1 D K R AO1 S T DH AH0 TH R EH1 SH OW2 L D S IH1 N S DH AH0 D AA1 R K D EY1
character /grapheme	t h a t _ n e i t h e r _ o f _ t h e m _ h a d _ c r o s s e d _ t h e _ t h r e s h o l d _ s i n c e _ t h e _ d a r k _ d a y _
subword /wordpiece	t h a t _ n e i t h e r _ o f _ t h e m _ h a d _ c r o s s e d _ t h e _ t h r e s h o l d _ s i n c e _ t h e _ d a r k _ d a y _
word	that neither of them had crossed the threshold since the dark day

- Zheng, et al., "Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers", 2021.
- J. Li, et al., "Advancing acoustic-to-word CTC model", ICASSP 2018.

CTC: shortcoming

- Conditional independence assumption

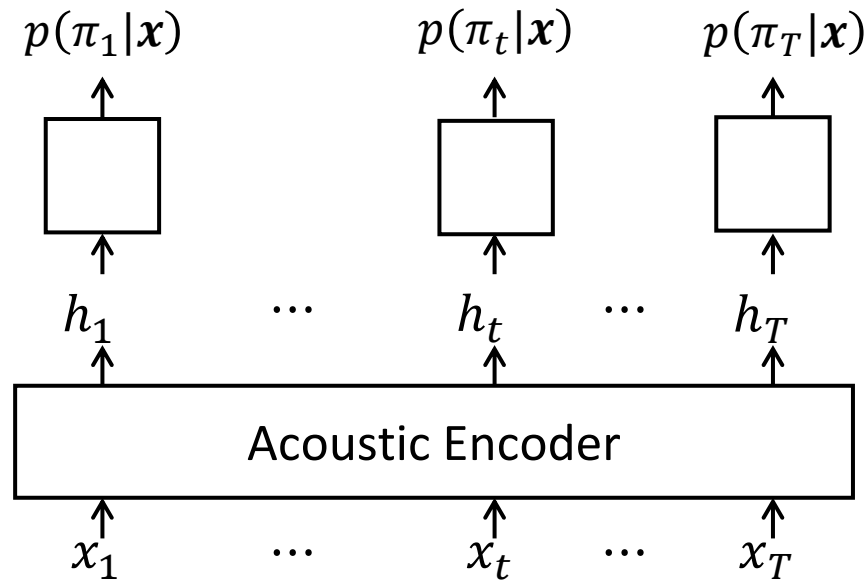
$$p(\boldsymbol{\pi} | \boldsymbol{x}) = \prod_{t=1}^T p(\pi_t | \boldsymbol{x})$$

Overcome

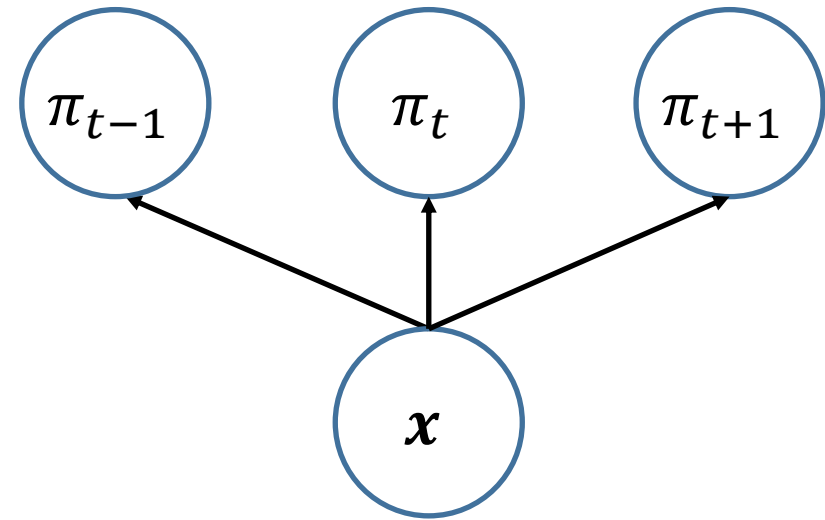
RNN-T

CTC-CRF

AED



Computational flow



Graphical Model Representation

Subsection summary

1.1 Modern speech recognition models

- End-to-end ASR models (CTC/CTC-CRF, RNNT, AED) are good supervised learners.
- Given large amounts of labeled speech for model training, they obtained impressive performance.
- How can we improve data-efficiency for ASR systems?

Content

I. Basics of Speech Recognition and LLMs (50 min)

1. Modern speech recognition models
2. Data-efficiency in Multilingual and Crosslingual Speech Recognition
3. Introduction to LLMs
4. Overview and taxonomy of LLM-ASR

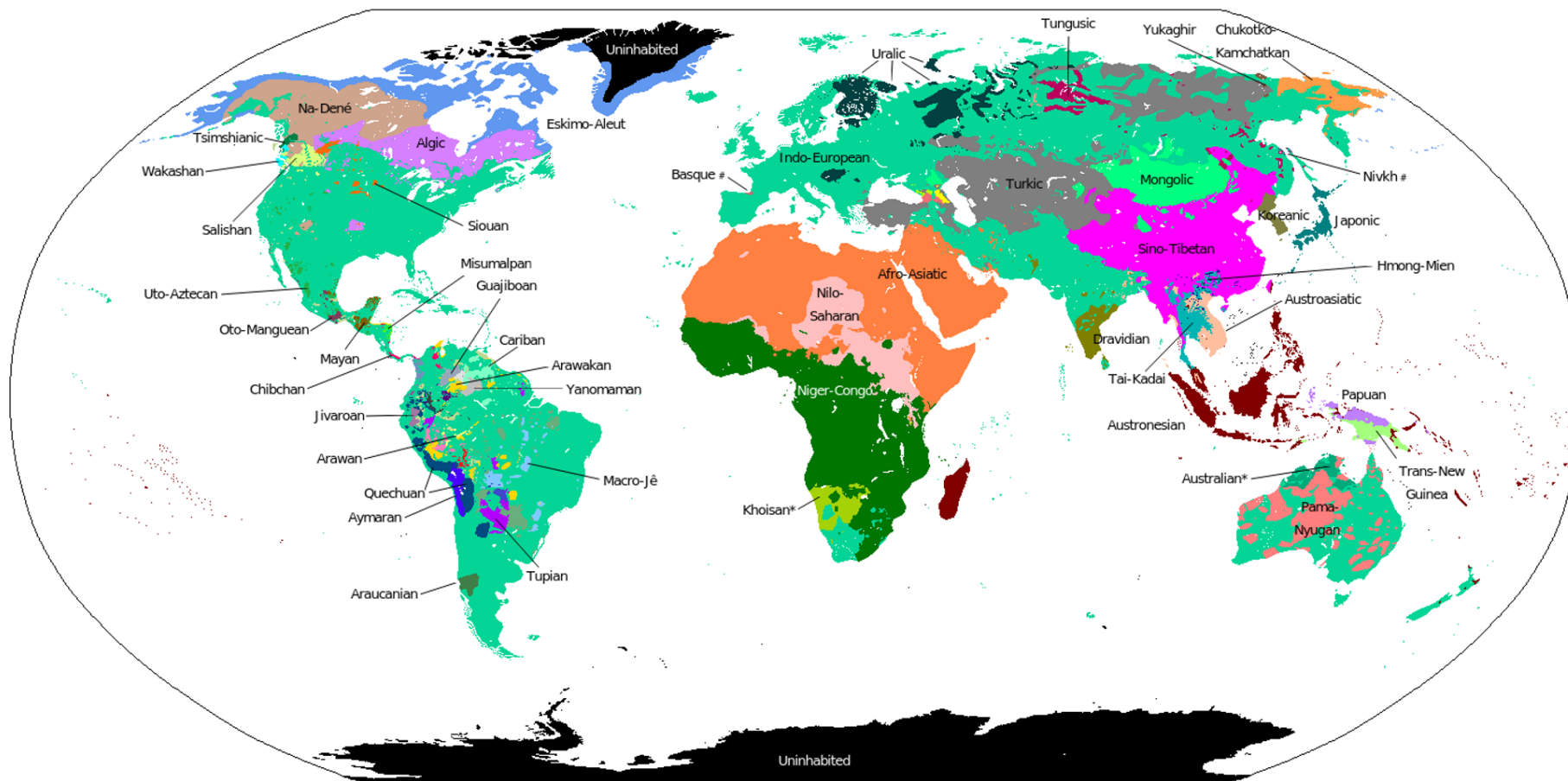
II. Continuous-interface based approach (50 min)

III. Discrete-interface based approach (50 min)

IV. Open questions and future directions (20 min)

Motivation

- There are more than 7100 languages in the world, and most of them are low-resourced languages.



Motivation

- There are more than 7100 languages in the world, and most of them are low-resourced languages.
- Multilingual speech recognition
 - Training data from a number of languages (seen languages) are merged to train a multilingual AM.
- Crosslingual speech recognition
 - The target language is unseen in training the multilingual AM.
 - In **few-shot** setting , the AM can be finetuned on limited target language data.
 - In **zero-shot** setting , the AM is directly used without finetuning*.

* Suppose that text corpus from the target language are available.

Intuitively, the key to successful multilingual and crosslingual recognition is to promote the information sharing in multilingual training and maximize the **knowledge transferring** from the well trained multilingual model to the model for recognizing the utterances in the new language.

Data-efficient

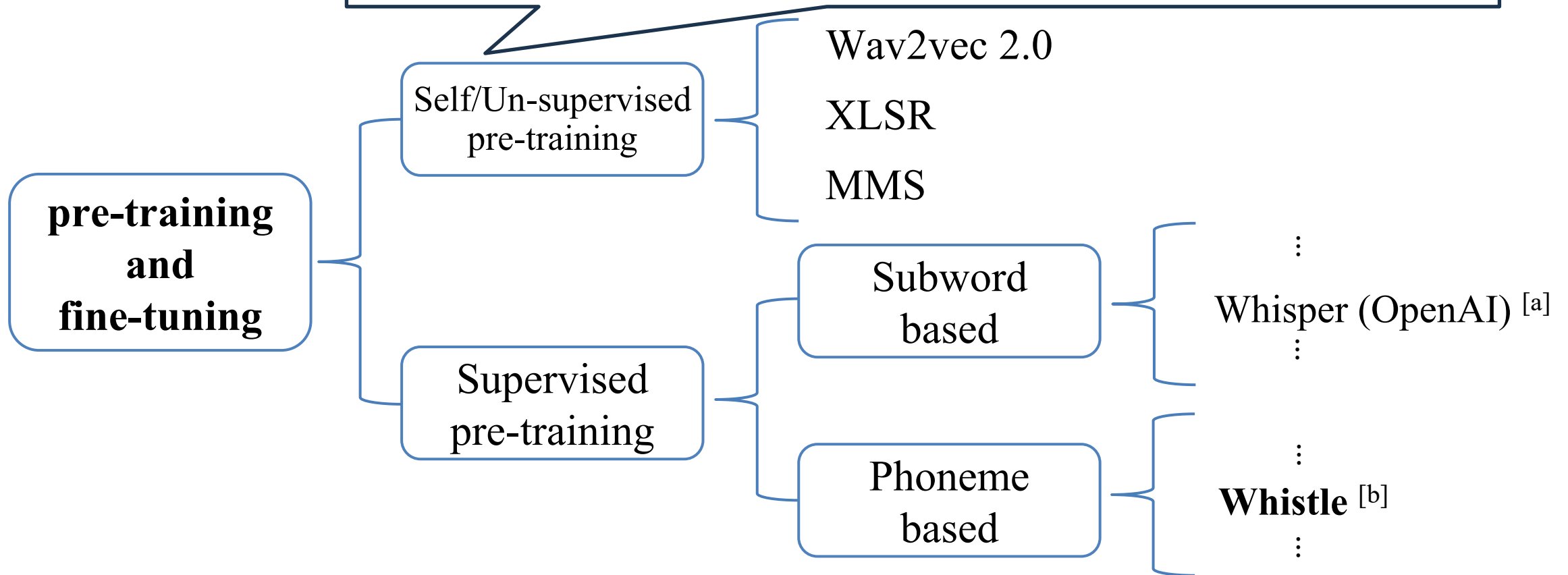
$$Efficiency = \frac{Performance}{Cost}$$

- Data-efficiency can refer to many things
 - https://en.wikipedia.org/wiki/Data_efficiency
 - Different from: computation-efficient (MIPS, million instructions per second), power-efficient (MIPS/Watt)
- In machine learning, data-efficiency mainly refers to: achieving better performance with the same amount of labeled/sampled data.
- Data-hungry: heavy reliance on supervised learning and large amounts of manually-labeled data.
- A spectrum of data-efficient modeling and learning methods
 - ✓ Model architecture
 - ✓ Unsupervised, self-supervised, semi-supervised learning /Pre-training

Data-efficiency is a multi-dimensional problem.

Background

- ☹ Only train Encoder /lack of an equivalently high quality pretrained decoder: “a crucial weakness which limits their usefulness” [a]
- ☹ Does not satisfy *principled unsupervised learning*, since “the unsupervised objective may be unrelated to the supervised task of interest” [c]



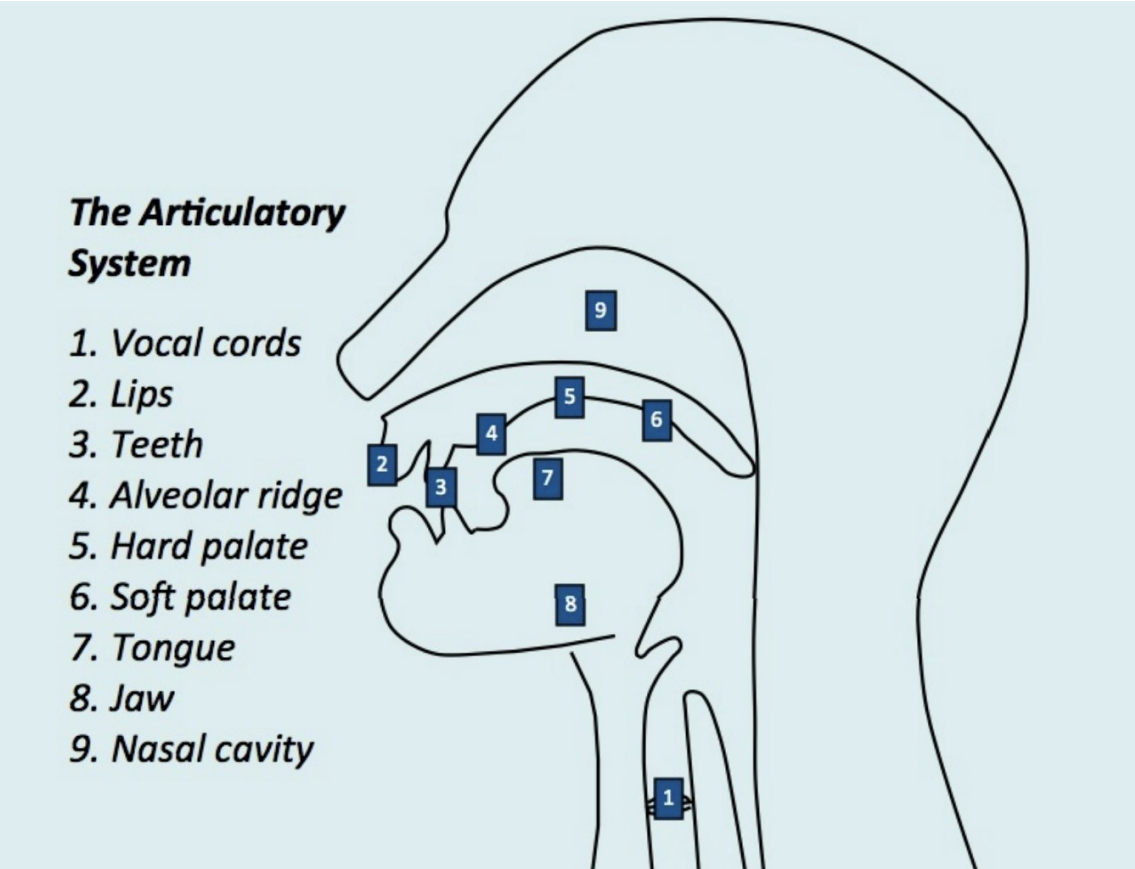
[a] (Whisper) “Robust speech recognition via large-scale weak supervision,” ICML 2023.

[b] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, “Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision,” IEEE Transactions on Audio, Speech and Language Processing, 2025.

[c] I. Sutskever, et al. “Towards principled unsupervised learning,” arXiv 2015.

Background: International Phonetic Alphabet (IPA)

All human languages, without exception, are produced by the same set of human vocal



since 1888 THE INTERNATIONAL PHONETIC ALPHABET (revised to 2015)

CONSONANTS (PULMONIC)												© 2015 IPA	
	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal		
Plosive	p b		t d			ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ		
Nasal	m	ɱ	n			ɳ	ɲ	ŋ	ɴ				
Trill	ʙ		r						ʀ				
Tap or Flap		ⱱ	ɾ			ɽ							
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ		
Lateral fricative			ɬ ɮ										
Approximant		ʋ	ɹ			ɻ	j	ɰ					
Lateral approximant			l			ɭ	ʎ	ʟ					

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

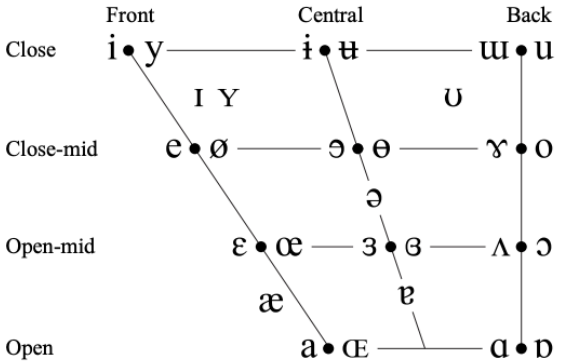
CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
◌ Bilabial	ɓ Bilabial	ʼ Examples:
Dental	ɗ Dental/alveolar	pʼ Bilabial
! (Post)alveolar	ɟ Palatal	tʼ Dental/alveolar
‡ Palatoalveolar	ɡ Velar	kʼ Velar
Alveolar lateral	ɠ Uvular	sʼ Alveolar fricative

OTHER SYMBOLS

ɱ Voiceless labial-velar fricative	ɕ ʑ Alveolo-palatal fricatives
ʋ Voiced labial-velar approximant	ɺ Voiced alveolar lateral flap
ɰ Voiced labial-palatal approximant	ɥ Simultaneous ʃ and x
ʜ Voiceless epiglottal fricative	Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary.
ʕ Voiced epiglottal fricative	
ʡ Epiglottal plosive	

VOWELS



Where symbols appear in pairs, the one to the right represents a rounded vowel.

SUPRASEGMENTALS

ˈ Primary stress	ˌ Secondary stress
ː Long	ˑ Half-long
ˑ Extra-short	ː Minor (foot) group
ˑ Major (intonation) group	

DIACRITICS Some diacritics may be placed above a symbol with a descender, e.g. ɲ̥

◌ Voiceless	◌ Breathy voiced	◌ Dental
◌ Voiced	◌ Creaky voiced	◌ Apical

Research Questions

- **Question 1:** When comparing supervised or self-supervised pre-training methods, which performs better for MCL-ASR?
- **Question 2:** When comparing phoneme-based and subword-based supervised pre-training methods, which performs better for MCL-ASR?

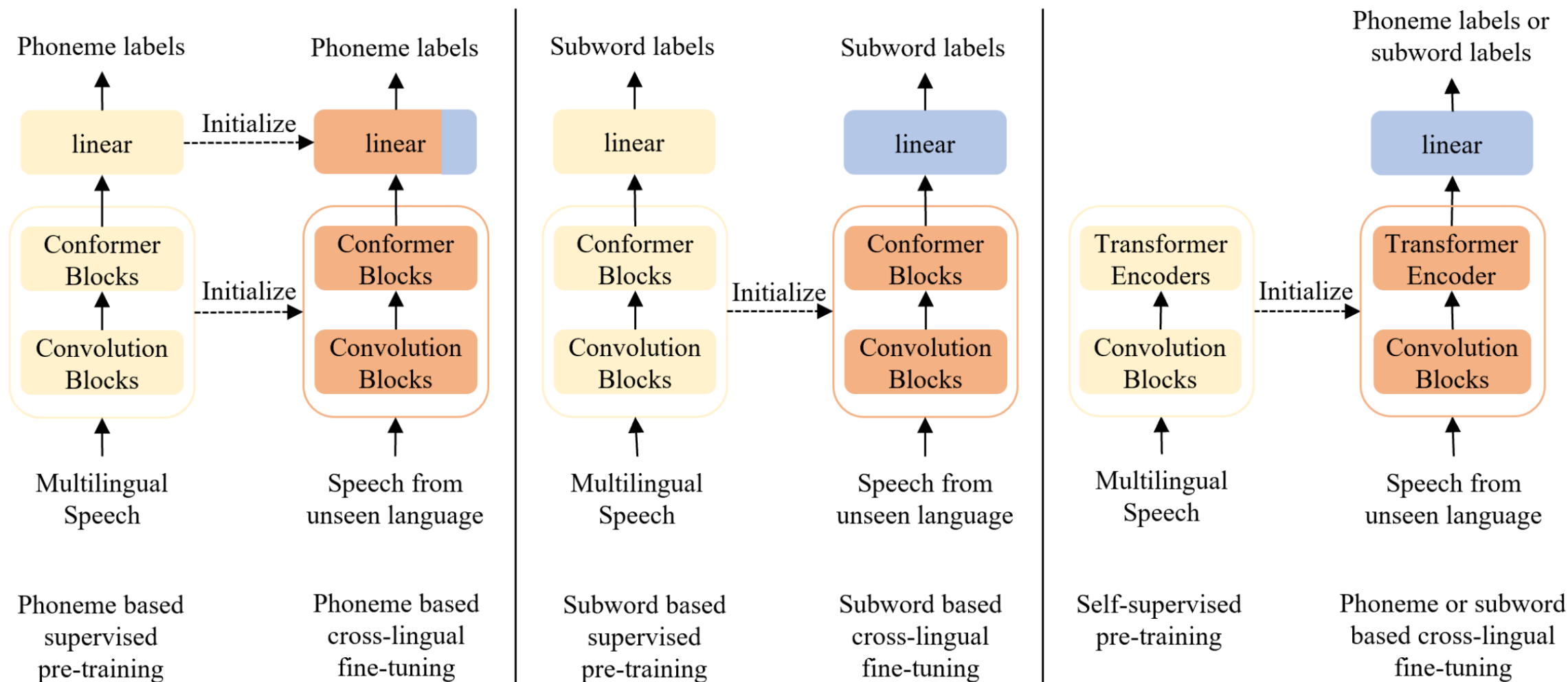
Conduct the comparison as fairly as possible
under identical settings (data volume, network architecture, model size, etc.)

Methods

Pretraining

Random initialization

Initialize from pretrained model



(a) Phonetic Supervision

(b) Subword Supervision

(c) Self-supervision

Fig.1. Illustration of the pretraining and finetuning procedures with (a) phonetic supervision, (b) subword supervision, and (c) self-supervision

Methods

- **Phoneme-based Supervised Pre-training:**

- Utilizes the IPA as a universal phoneme inventory
- Promotes information sharing across different languages
- Referred to as **phoneme-based weakly-supervised training** due to transcription errors (PER: 7%~45%)

- **Subword-based Supervised Pre-training:**

- Constructs a shared symbol set from multilingual text corpora
- Affects information sharing among multiple languages to some extent
- Requires careful trade-offs between tokenization granularity and ASR performance
- Demands complex tokenization redesign when adding new languages
- Necessitates random initialization of the weights in the linear classification layer

- **Self-supervised Pre-training:**

- Learns representations from multilingual unlabeled audio
- Faces a decoder-capacity bottleneck (mismatch with encoder performance)
- Supports flexible fine-tuning with either grapheme or phoneme-based approaches

Dataset

- We construct the **CV-Lang10** dataset comprises 4,096 hours of speech across 10 languages from CommonVoice v11
- 73 phoneme units
- 4,998 subword (BPE) units

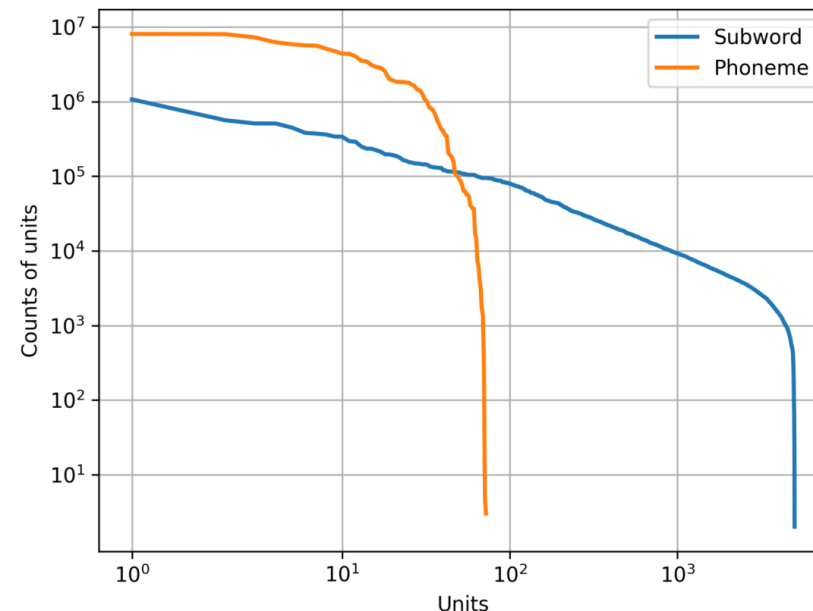
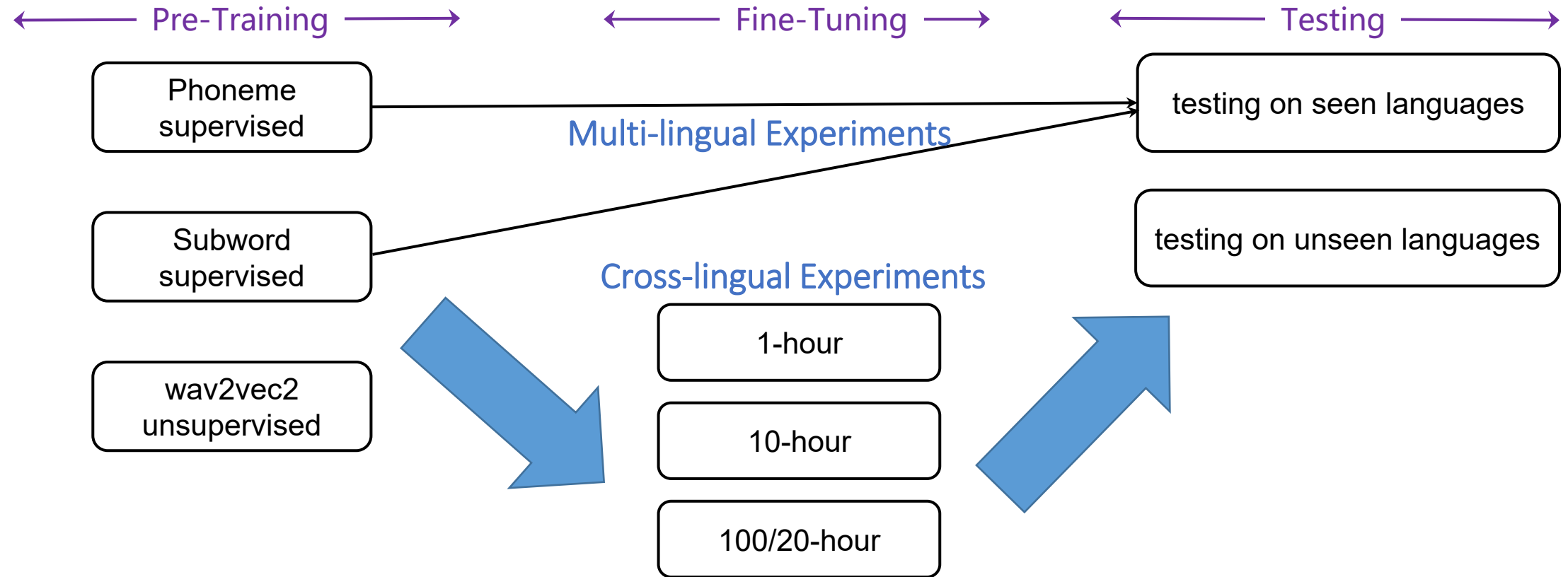


Fig. 2. Counts of phoneme and subword units in the CV-Lang10 training.

	Code	Text transcript	Transcription with subwords	Transcription with IPA symbols
English	en	i know everything about you	i know everything about you	ɑɪnəʊvɛvriθɪŋəbaʊtju
Spanish	es	no lo he visto	no lo he v ist o	noloebisto
French	fr	vous ne me comprenez pas	vous ne me comp ren ez pas	vynmkɔpʁənɛpa
Italian	it	è meglio separarci adesso	è me g lio separ ar ci ad esso	ɛmɛʎioseparartʃarsso
Kyrgyz	ky	менин эч кандай кунөөм жок	мен ин эч кандай кун өөм жок	mɛnɪnɛtʃkɑndɑjkynømdʒok
Dutch	nl	ze is een bekend model	ze is een bek end mod el	zeisenbəkɛntmɔdəl
Russian	ru	база данных обновлена	ба за дан ных об нов лен а	bazadanixobnovlɛna
Swedish	sv	hörni ta det lugnt	h ör ni ta det lug n t	hærnitɑdetlʉŋnt
Turkish	tr	bunlar en büyükleri	bun lar en b üy ük leri	bunɫaɾɛnbɯjyɫɛɾi
Tatar	tt	менәшулай яшәп ятабыз	мен әшулай яш әп я та быз	mjenæʃulajjæpɟatabɯz
Polish	pl	lubię muzykę klasyczną	lu b ię mu zy kę k la sy cz ną	luabviɛmuwzikɛklatsiɕɲɔɲ
Indonesian	id	semoga cepat sembuh	sem o ga c ep at sem b uh	sɛmʊgɑtʃɛpatɕɪmbʊh

Table 1 Example transcriptions for each language in CV-Lang10.

Experiment protocol



Data-efficiency in MCL-ASR

- **Pretraining data-efficiency** (multilingual data-efficiency)
 - Evaluates the performance of a pretrained, multilingual model in recognizing seen languages
- **Finetuning data-efficiency** (crosslingual data-efficiency)
 - Measures the performance of finetuned models in recognizing unseen languages.

Multi-lingual Experiments

PT: Pre-Training
FT: Fine-Tuning

IPA phoneme-based supervised pretraining (10 languages)



	Model		en	es	fr	it	ky	nl	ru	sv	tr	tt	Tot.
	Training data hours		2227	382	823	271	32	70	149	29	61	20	4069
	Model	Size(M)	en	es	fr	it	ky	nl	ru	sv	tr	tt	Avg.
O1	Monolingual phone	90	10.5	7.9	15.5	9.2	1.0	8.8	1.6	8.3	8.4	9.7	8.1
M4	Multi-lingual subword (BPE=5000)	92	12.0	9.8	12.4	9.9	3.2	9.6	3.3	9.9	9.1	13.5	9.3
M1	Multi-lingual phone (small)	90	10.7	8.6	16.0	9.9	1.0	7.3	1.5	6.1	7.6	7.3	7.6
M2	Multi-lingual phone (medium)	218	9.8	7.8	14.9	9.0	0.9	6.5	1.6	5.6	7.2	7.3	7.1
M3	Multi-lingual phone (large)	543	8.8	7.0	14.0	8.1	0.9	6.2	1.4	5.0	7.0	6.9	6.5

↓18%

- Multilingual **phone** supervised PT outperforms Multilingual **BPE** supervised PT.
- A single large multilingual phone supervised PT outperforms monolingual separately-trained 10 models.

Multilingual Data-efficiency

We provide two points to understand **why** phonetic supervision obtains better multilingual data-efficiency.

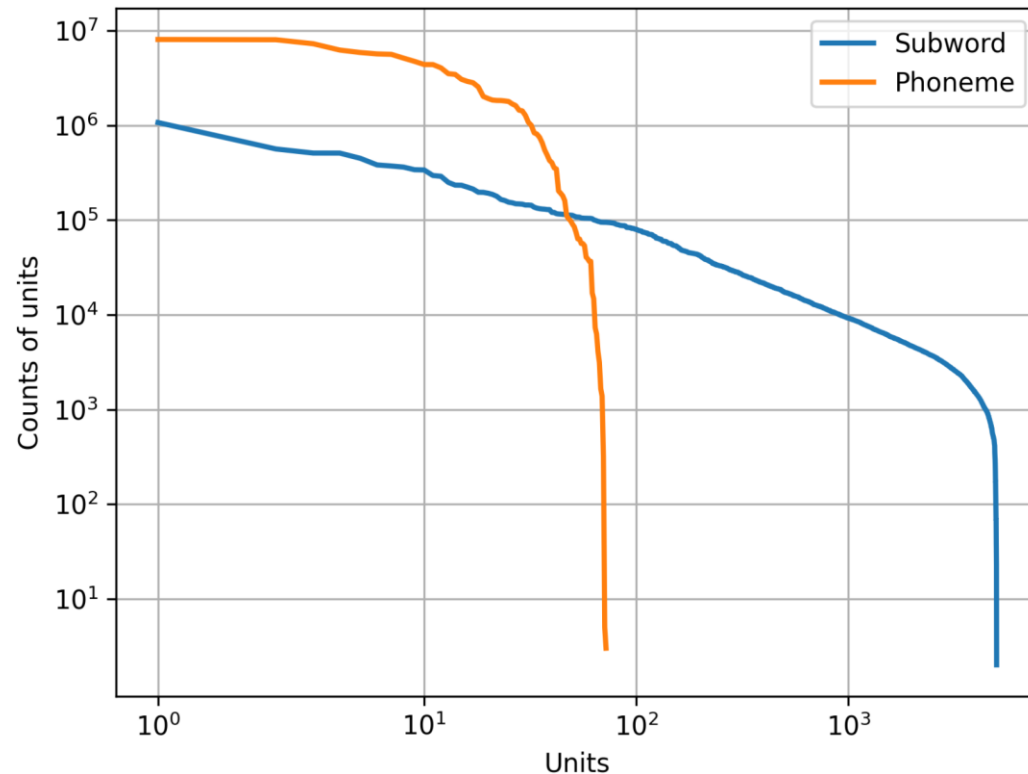


Fig. 2. Counts of phoneme and subword units in the CV-Lang10 training.

- From the perspectives of **data balance**:
Graphemic supervision suffers from a more severe data imbalance.

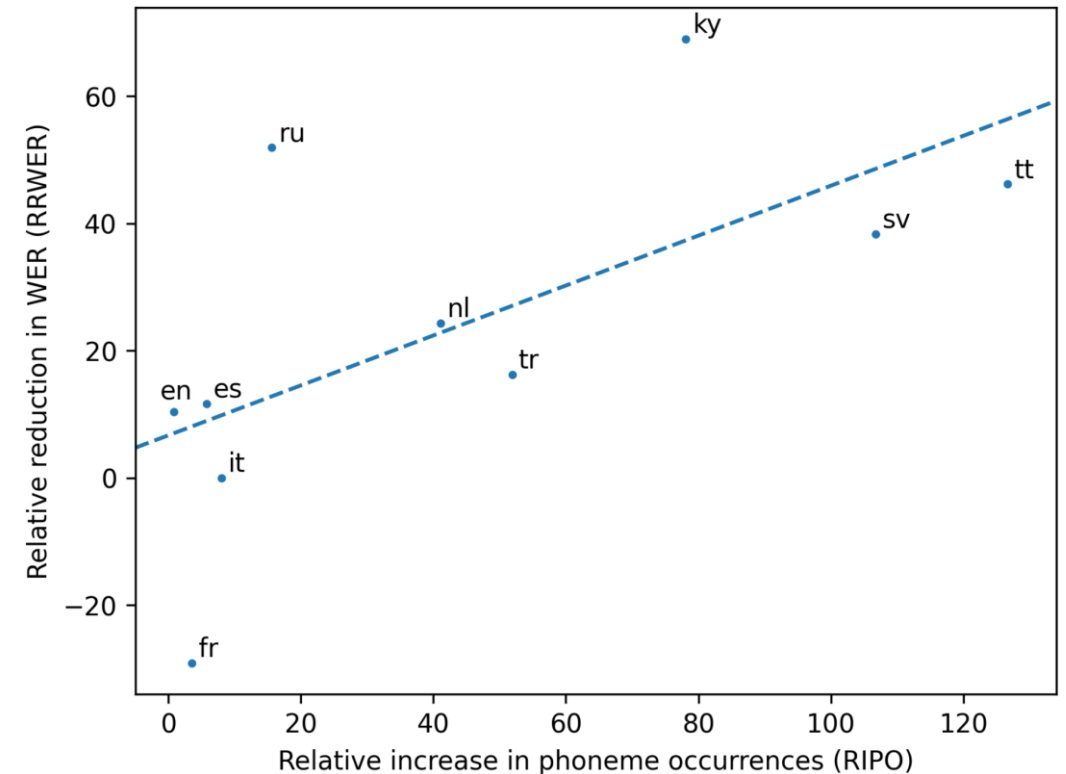


Fig3. RRWER (phone PT vs subword PT) as a function of RIPO.

- From the perspectives of **data augmentation**:
Phonetic supervision enables a more efficient data sharing (implicit data augmentation).

Cross-lingual Experiments

Phoneme FT on Polish (phone=37, #word=43,746)

- 1h: **Phone PT** + phone FT outperforms **BPE PT** + BPE FT, both outperforms wav2vec2 PT
- 10h/130h: **Phone/BPE PT** + BPE FT is best, better than Phone PT + Phone FT
- 130h: wav2vec2 PT starts getting better

	Model	1h		10h		130h	
		PER	WER	PER	WER	PER	WER
1	Monolingual phone	86.0	99.9	30.3	13.8	2.8	4.9
2	Wav2vec2.0-base (En) phone FT	25.7	11.0	16.6	6.7	5.8	4.5
3	Wav2vec2.0-base (10 lang) phone FT	21.1	7.9	12.6	5.6	6.0	4.4
4	phone PT (10 lang) and phone FT	17.9	6.9	10.4	5.2	1.9	4.3

Cross-lingual is mostly meaningful in low-resource setting.

Subword FT on Polish (#BPE=500, #word=43,746)

	Model	1h (WER)		10h (WER)		130h (WER)	
		w/o LM	w LM	w/o LM	w LM	w/o LM	w LM
1	Monolingual subword	98.4	98.3	90.9	59.4	19.3	7.1
2	Wav2vec2.0-base (En) subword FT	100	100	45.6	7.0	8.5	3.8
3	Wav2vec2.0-base (10 lang) subword FT	99.9	100	36.9	5.7	7.4	3.4
4	subword PT (10 lang) and subword FT	70.1	9.1	31.9	4.8	5.4	3.7
5	phone PT (10 lang) and subword FT	69.5	8.6	31.8	4.8	5.8	3.8

The effects of different PT methods diminish

Cross-lingual Experiments

The advantages of phone PT + phone FT are more obvious, across all the three scales of data settings.

Phoneme FT on Indonesian (#phone=37, #word=13,658)

	Model	1h		10h		20h	
		PER	WER	PER	WER	PER	WER
1	Monolingual phone	96.5	100	27.3	7.7	5.7	3.2
2	Wav2vec2.0-base (En) phone FT	31.3	6.7	10.8	3.3	6.8	2.8
3	Wav2vec2.0-base (10 lang) phone FT	24.9	3.7	10.3	2.7	6.3	2.4
4	phone PT (10 lang) and phone FT	21.6	3.2	7.9	2.5	4.7	2.4

Subword FT on Indonesian (#BPE=500, #word= 13,658)

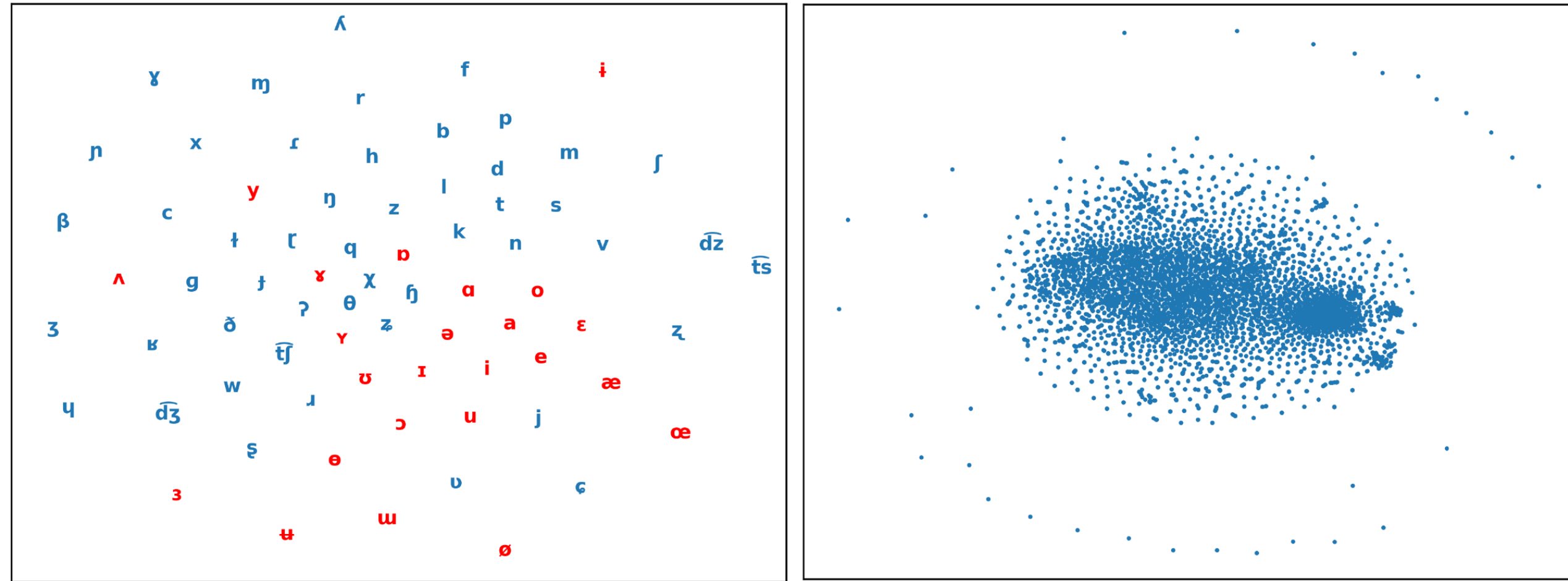
	Model	1h (WER)		10h (WER)		20h (WER)	
		w/o LM	w LM	w/o LM	w LM	w/o LM	w LM
1	Monolingual subword	96.6	96.4	69.5	49.6	31.9	10.8
2	Wav2vec2.0-base (En) subword FT	100	100	19.9	5.2	11.6	3.5
3	Wav2vec2.0-base (10 lang) subword FT	99.6	99.9	19.0	4.5	12.0	3.1
4	subword PT (10 lang) and subword FT	64.0	23.5	19.4	3.9	13.1	3.0
5	phone PT (10 lang) and subword FT	67.7	24.5	18.2	3.5	12.4	2.9

Crosslingual Data-efficiency

- In summary, we find that compared to both graphemic supervision and self-supervision, **phonetic supervision** excels in crosslingual data-efficiency.
- The **efficiency advantage** is more significant when the data are more limited and may diminish when the finetuning data are abundant.

Embedding visualization

- Representation learning in the subword model is not so balanced as in the phoneme-model.



(a) **Fig. 4. Visualization of embeddings by t-SNE:** (b)

(a) **Phoneme embeddings** from M1 (blue=consonants, red=vowels), (b) **Subword embeddings** from M4.

Test of catastrophic forgetting

Multilingual models pre-trained on CV-Lang10 are fine-tuned with 10 minutes of Polish.

- **WARD (Word Accuracy Relative Degradation)** quantifies the relative WER increase across the ten original CV-Lang10 languages.

Phoneme-based model (M1) after 10-min Polish fine-tuning:

- Obtains 11.0% WER on Polish; Word accuracy drops by 48% across 10 pretrained languages

id	Model	pl	en	fr	es	it	ru	nl	tr	ky	sv	tt	Avg.	WARD
M19	M1 + 10min phoneme FT	11.0	68.5	69.3	57.1	50.3	48.3	60.9	31.8	58.4	42.3	33.0	52.0	48
M20	M4 + 10min subword FT	93.2	92.2	95.0	92.5	92.5	262.5	103.6	241.5	125.9	180.5	254.4	154.1	160

Subword-based model (M4) after 10-min Polish fine-tuning:

- Obtains 93.2% WER on Polish; Breaks down in recognizing the ten old languages

Phoneme PT and FT are more robust in overcoming catastrophic forgetting.

Training efficiency

Phoneme-based models tend to be more training efficient, i.e., they can converge with fewer optimization steps

id	Model	Batch size	Epochs for converging
M1	phoneme PT	640	63
M11	M1 + pl subword FT	320	195
M4	subword PT	640	83
M10	M4 + pl subword FT	320	223

24% fewer epochs for PT

12% fewer epochs in Polish FT

Table 10. Training efficiency of phoneme-based vs. subword-based pre-training (PT) and fine-tuning (FT).

- Phoneme labels can provide **more efficient** supervision for sound classification than subword labels
- It takes a longer, **less efficient** path for neural networks to learn sound classification from subword supervision

Iu-Mien experiment

- Iu-Mien is a major branch of the Yao language, spoken by roughly 840,000 people across China, Southeast Asia, and diaspora communities in the United States.
- It is tonal and monosyllabic. It is written using the 26 Latin letters, and each basic Mien writing unit corresponds to a basic pronunciation.
- MightLJSpeech dataset [1]: **9.7 hours**

	Iu Mien word	English meaning	syllable	initial consonant	vowel			tone
					initial	main	final	
	ginghgungv	Dragonfly	gingh	g	-	i	ng	h
			gungv	g	-	u	ng	v
	baengh	Flat	baengh	b	-	ae	ng	h
	nqaang	Later	nqaang	nq	-	aa	ng	-
	guinh	Circle	guinh	g	u	i	n	h

Table 11. Examples of Mien words.

[1] <https://github.com/mightmay/MightLJSpeech/tree/master/MightLJSpeech-1.1/csv>

Iu-Mien experiment

- Three independent experiments, each with an 8:1:1 split for training : dev : test.
- For both phoneme fine-tuning and subword fine-tuning, **the Whistle pre-trained model** achieved better recognition accuracy on the low-resource Mien language than other pre-training methods.

id	Model	Test (PER)	Test (WER)
O2	Mono.phoneme	4.22	4.69
M2	Whistle-small + phoneme FT	2.41	2.71
M5	Wav2vec2-cv10 + phoneme FT	2.53	2.76

Table 12. Mien speech recognition based on phoneme modeling

id	Model	Test w/o LM (WER)	Test with LM (WER)
O1	Mono.subword	9.71	6.87
M1	Whistle-small + subword FT	3.30	2.95
M3	Wav2vec2-cv10 + subword FT	3.76	3.06
M4	Mul10-subword + subword FT	4.33	3.46

Table 13. Mien speech recognition based on subword modeling

Subsection summary

1.2 Data-efficient multilingual and crosslingual ASR

- **Data efficiency** means better recognition at a fixed labeled-data budget, or comparable accuracy with fewer labels.
- **Multilingual pretraining** pools knowledge across languages.
- **Crosslingual finetuning** adapts the model to a new language with limited labeled speech.
- **Whistle** uses weak phonetic supervision and shared IPA labels, improving multilingual and low-resource ASR in controlled comparisons.

Phonetic pretraining helps with limited labels; Transfer gains vary by language and data budget.

Content

I. Basics of Speech Recognition and LLMs (50 min)

1. Modern speech recognition models
2. Data-efficiency in Multilingual and Crosslingual Speech Recognition
3. Introduction to LLMs
4. Overview and taxonomy of LLM-ASR

Connecting speech encoders to LLMs defines what we call **LLM-ASR**.

What's the technical that makes LLMs success?

Is it because of scaling law?
Are there deeper underlying reasons?



Language model (LM)

LM: a probabilistic model of language (token sequences)

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{historical tokens } x_1, \dots, x_{i-1}) \quad \text{next-token prediction}$$

current token

Autoregressive LM:

At each position, use the historical tokens $x_1, x_2, \dots, x_{i-1}$ (context)

Compute the (conditional) probability of the current token $P(x_i | x_1, \dots, x_{i-1})$

The best thing about AI is its ability to

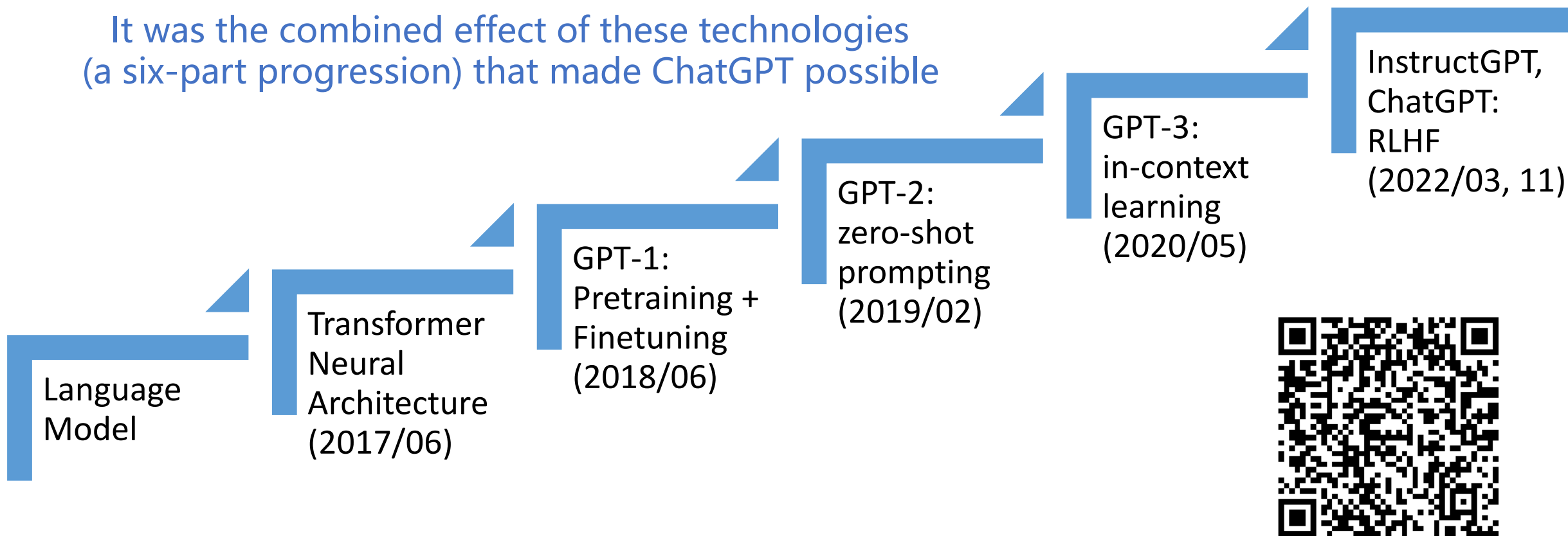
learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

The best thing about AI is its ability to,
The best thing about AI is its ability to learn,
The best thing about AI is its ability to learn from,
The best thing about AI is its ability to learn from experience,
The best thing about AI is its ability to learn from experience.,
The best thing about AI is its ability to learn from experience. It,
The best thing about AI is its ability to learn from experience. It's,
The best thing about AI is its ability to learn from experience. It's not

The progress of ChatGPT

Standing on the shoulders of giants from many years of artificial-intelligence research

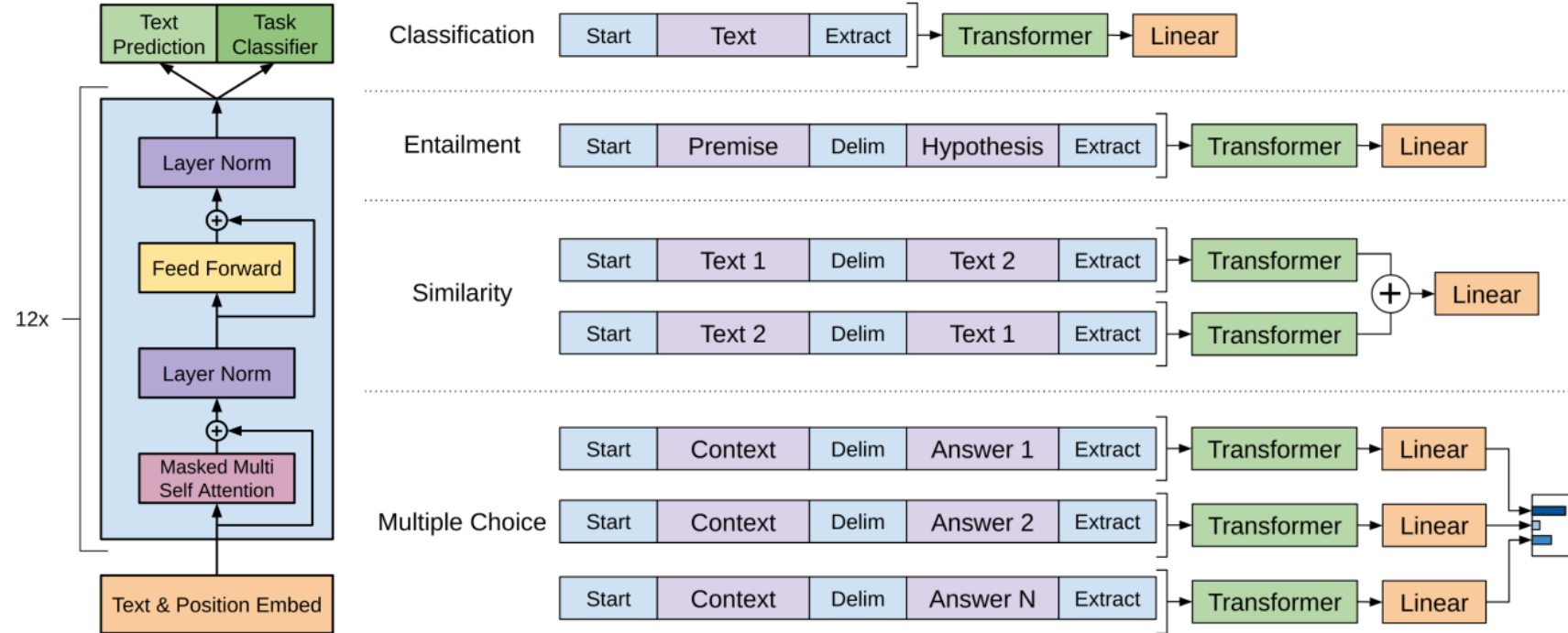
It was the combined effect of these technologies
(a six-part progression) that made ChatGPT possible



Chinese WeChat Blog: “严谨谈谈ChatGPT的进步、不足及AGI挑战”, Zhijian Ou, 2023/03/16.
“A Rigorous Discussion on ChatGPT’s Progress, Limitations, and the Challenge of AGI” 36

GPT and pretraining + finetuning techniques (2018)

Natural language understanding, consisting of a wide range of different tasks

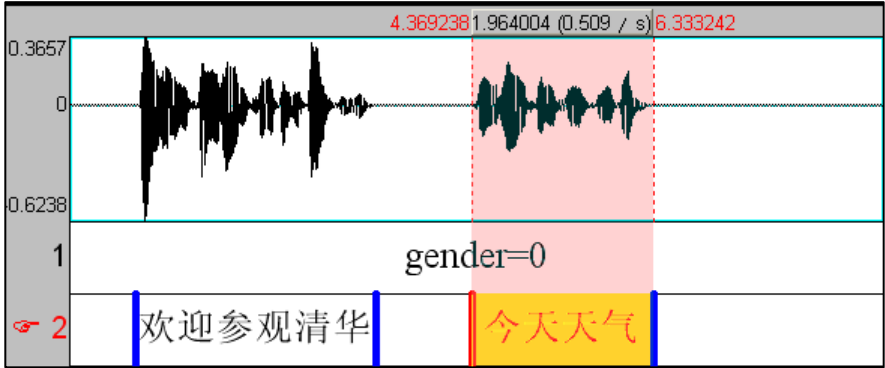


From the original GPT paper [4] "Improving language understanding by **Generative Pre-Training**" (2018)

"Our work broadly falls under the category of semi-supervised learning for natural language."

Unsupervised pre-training combined with supervised fine-tuning is a form of **semi-supervised learning**.

Supervised learning from labeled data $\{(x, y)\}$: a huge success!

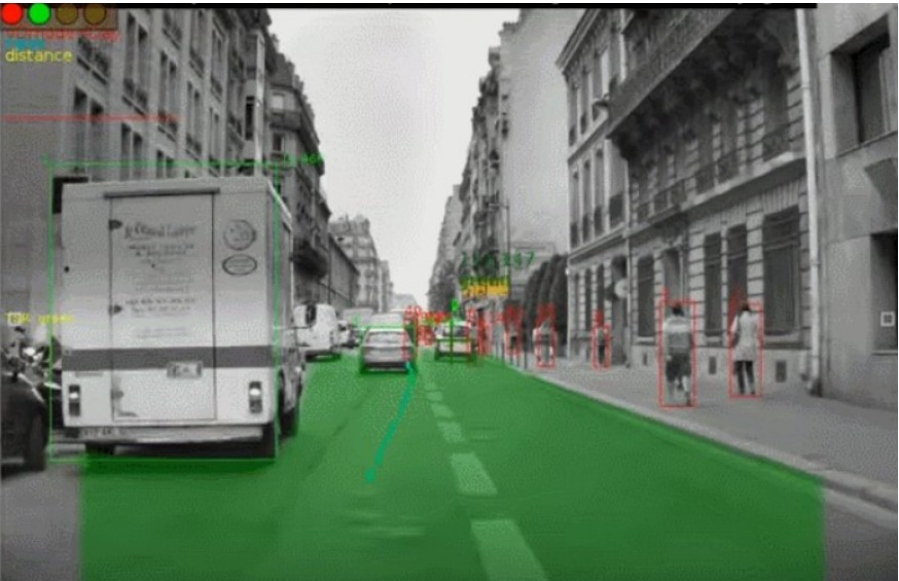


Speech recognition

ATIS UTTERANCE EXAMPLE IOB REPRESENTATION

Sentence	<i>show</i>	<i>flights</i>	<i>from</i>	<i>Boston</i>	<i>To</i>	<i>New</i>	<i>York</i>	<i>today</i>
Slots/Concepts	O	O	O	B-dept	O	B-arr	I-arr	B-date
Named Entity	O	O	O	B-city	O	B-city	I-city	O
Intent	<i>Find Flight</i>							
Domain	<i>Airline Travel</i>							

Intent detection, semantic slot filling, and named-entity recognition

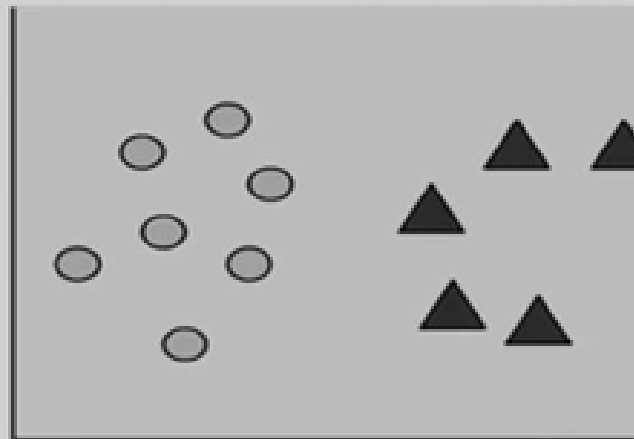


Object detection and tracking



Dialogue systems

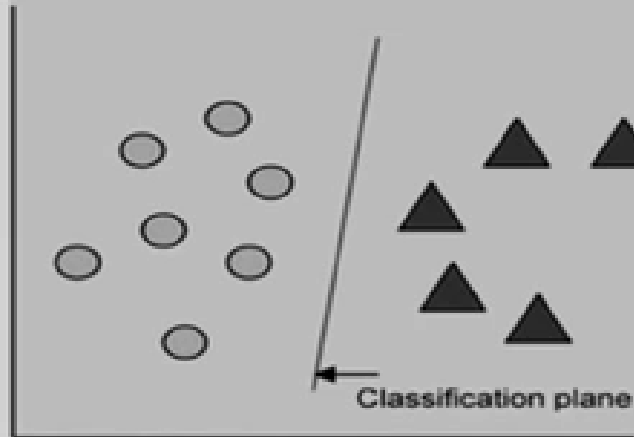
Semi-supervised learning (SSL)



Labeled Data

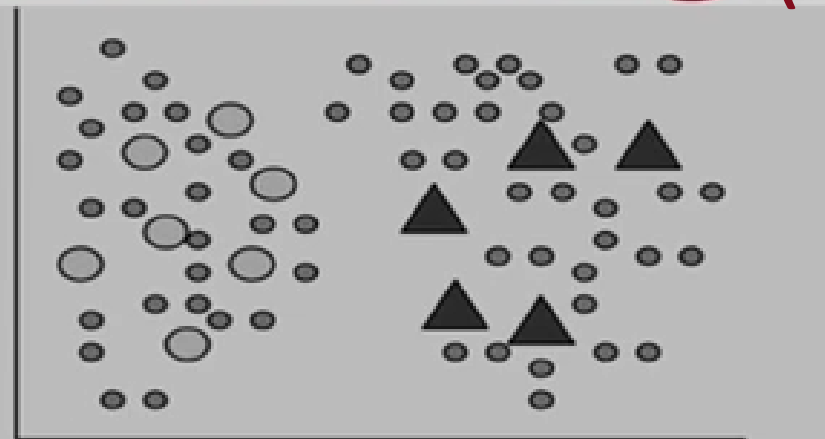
(a)

Supervised learning from labeled data $\{(x, y)\}$



Supervised Learning

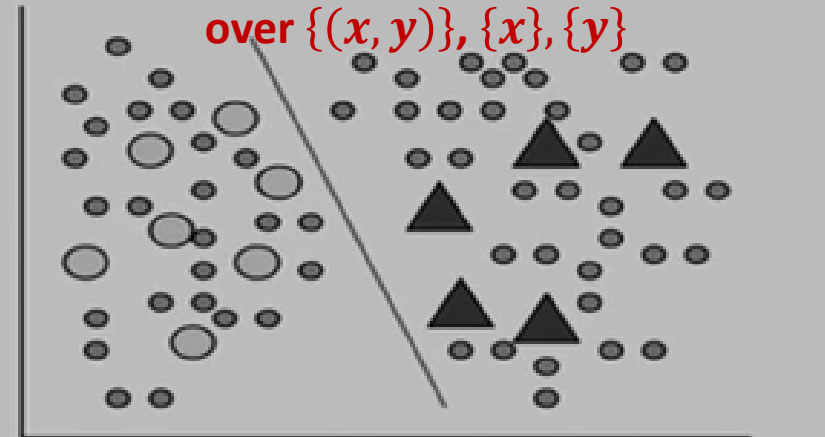
(c)



Labeled and Unlabeled Data

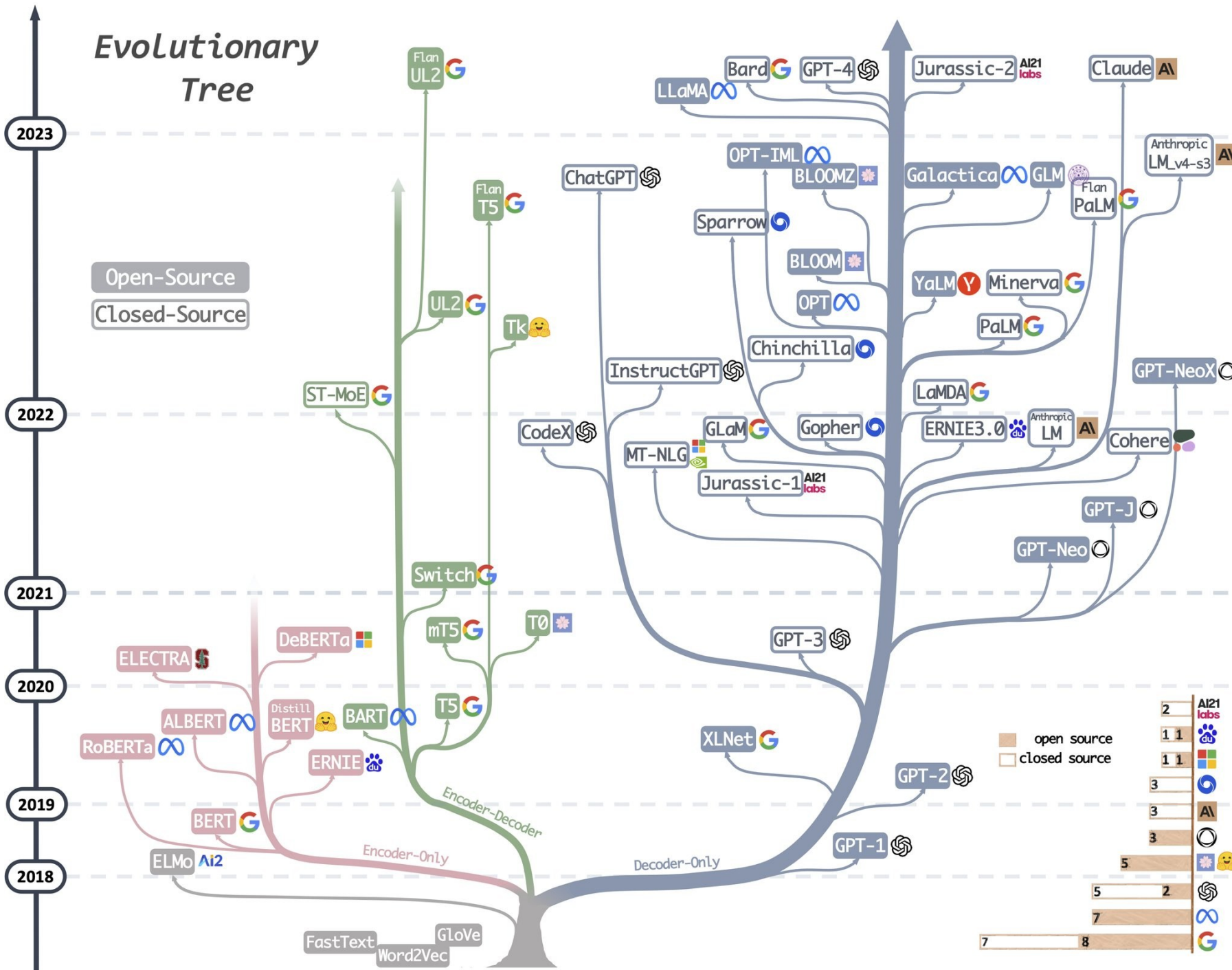
(b)

Collaborative supervised and unsupervised learning over $\{(x, y)\}, \{x\}, \{y\}$



Semi-Supervised Learning

(d)



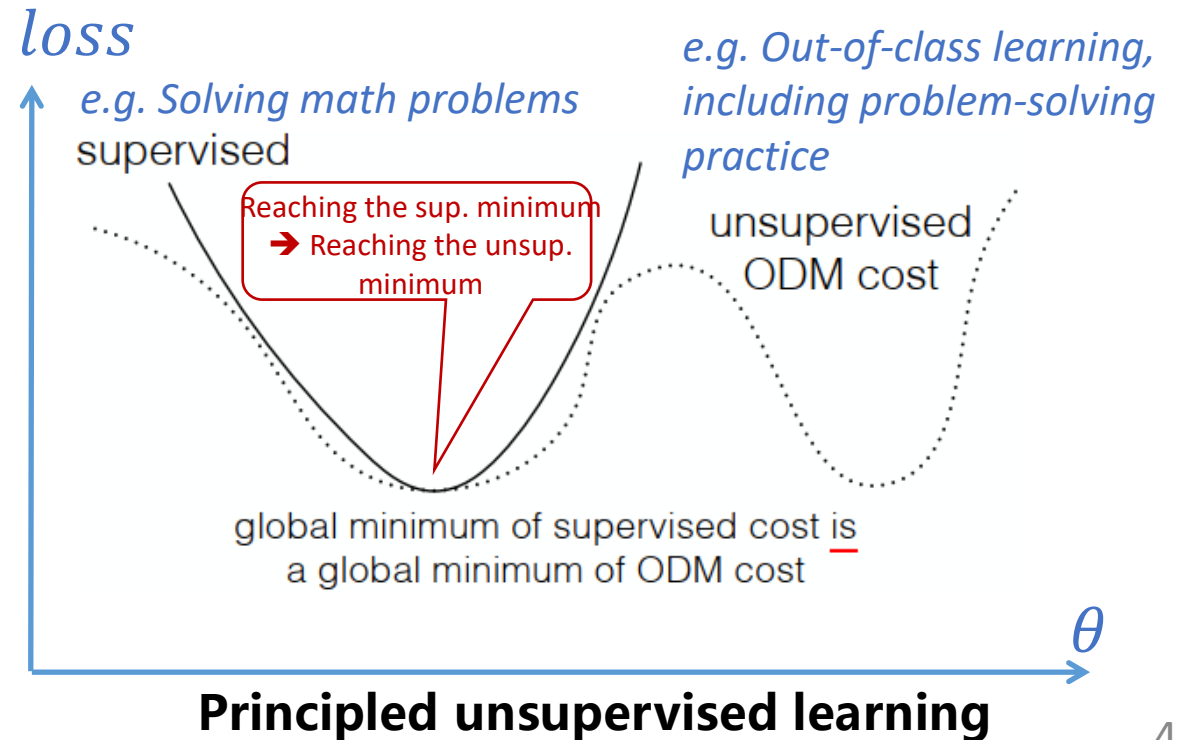
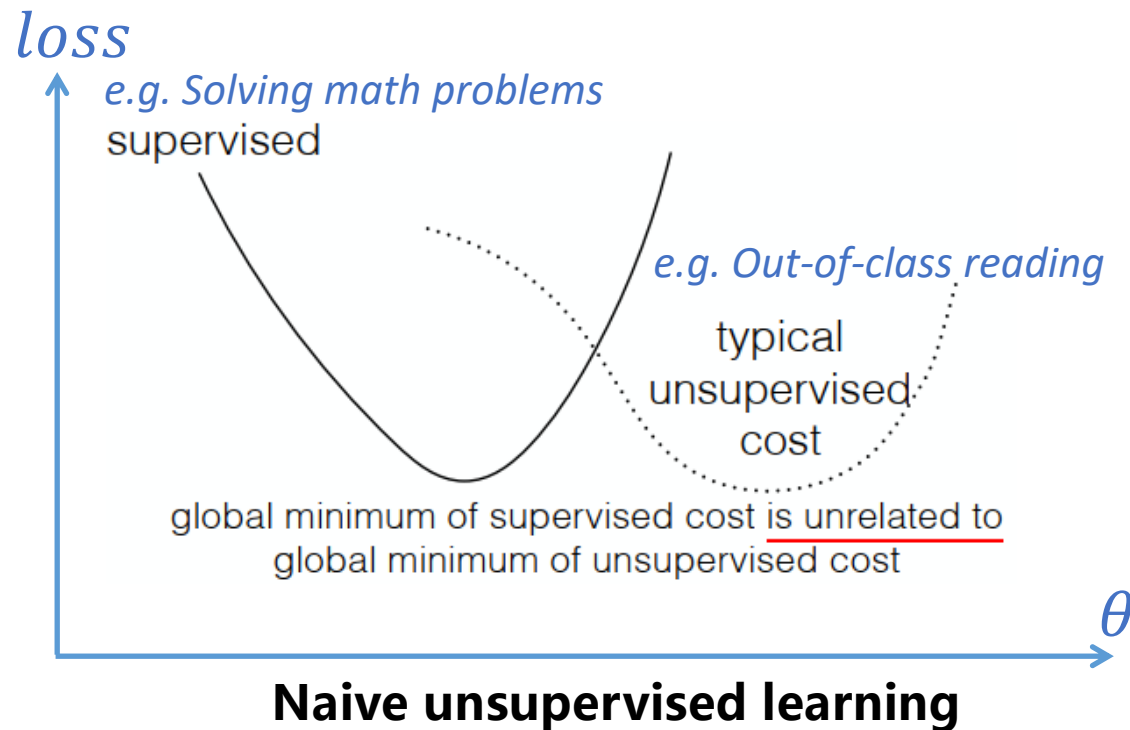
What advantages does **GPT** have, compared to encoder-only or encoder-decoder architectures?



TOWARDS PRINCIPLED UNSUPERVISED LEARNING

Ilya Sutskever¹, Rafal Jozefowicz¹, Karol Gregor², Danilo Rezende², Tim Lillicrap², Oriol Vinyals¹
Google Brain¹ and Google DeepMind²

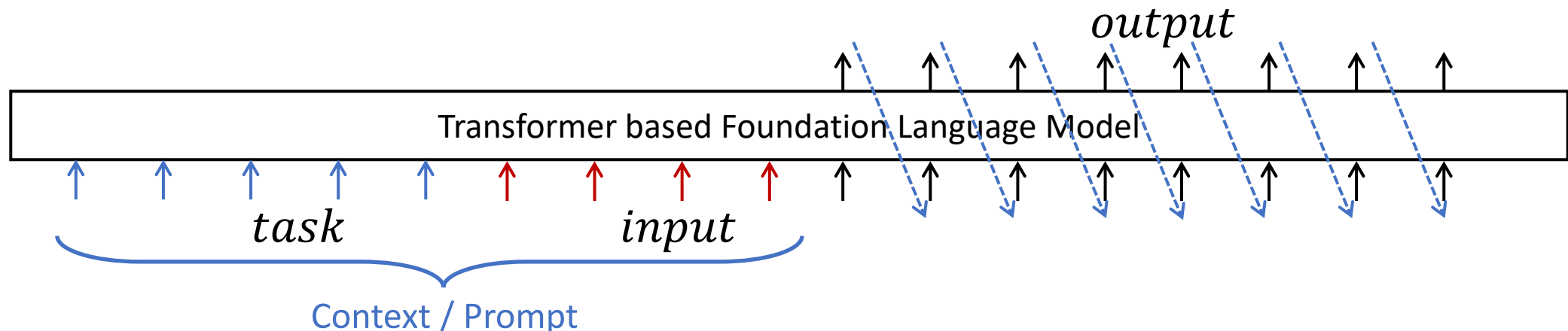
{ilyasu, rafalj, karolg, danilor, countzero, vinyals}@google.com



GPT-2 (2019)

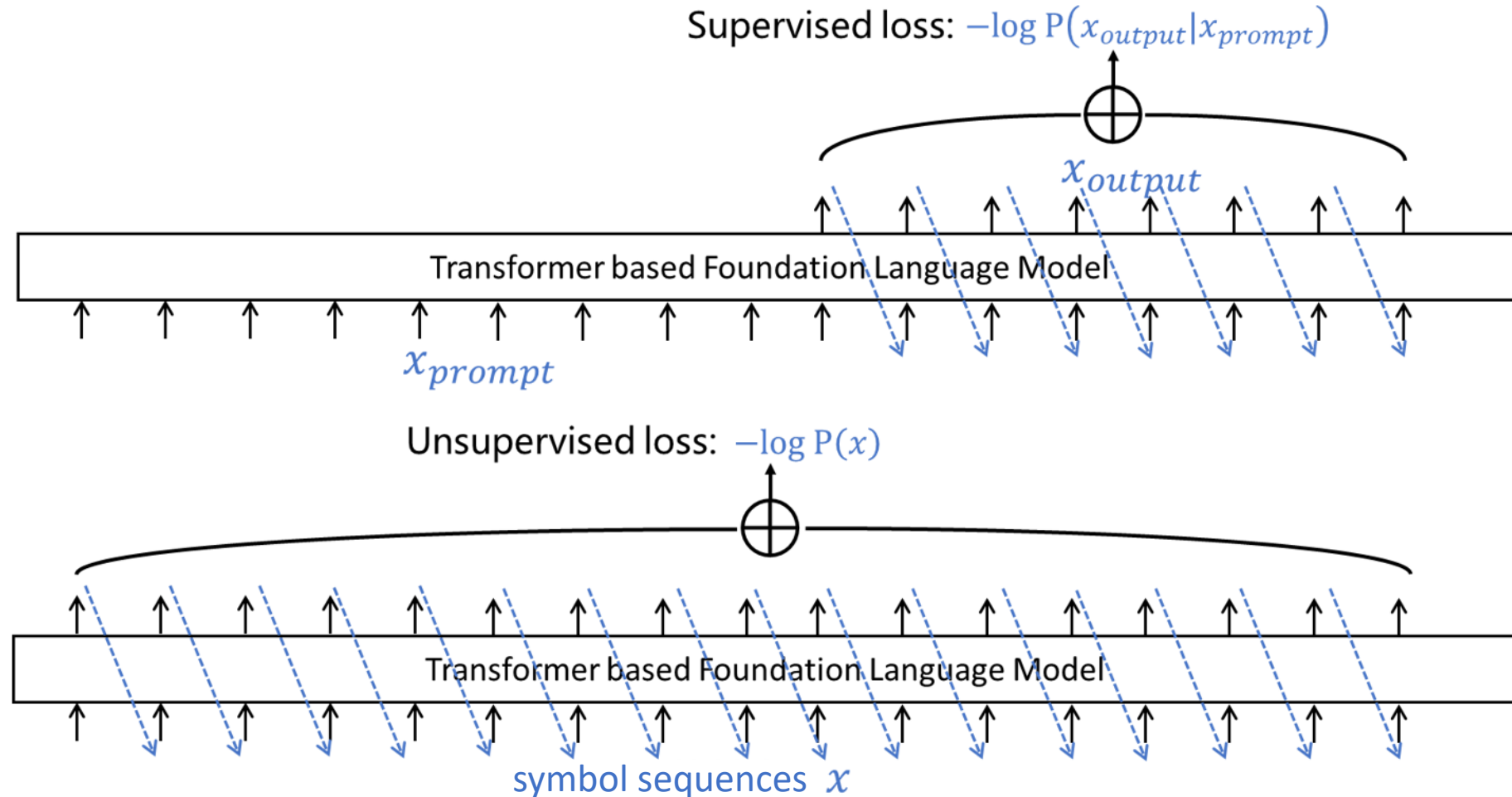
- Having a machine learn to perform a natural-language-understanding task is essentially **estimating the conditional distribution** $P(\text{output} \mid \text{input})$
- An innovation by GPT-2 is to express $\text{task}, \text{input}, \text{output}$ as token sequences — **reducing the model $P(\text{output} \mid \text{task}, \text{input})$ to a language model:** given the context, recursively generate the next symbol.
- Training data for different tasks are uniformly organized as $\text{task}, \text{input}, \text{output}$

For example, (translate to french, english text, french text)
(answer the question, document, question, answer)



GPT-2 (2019)

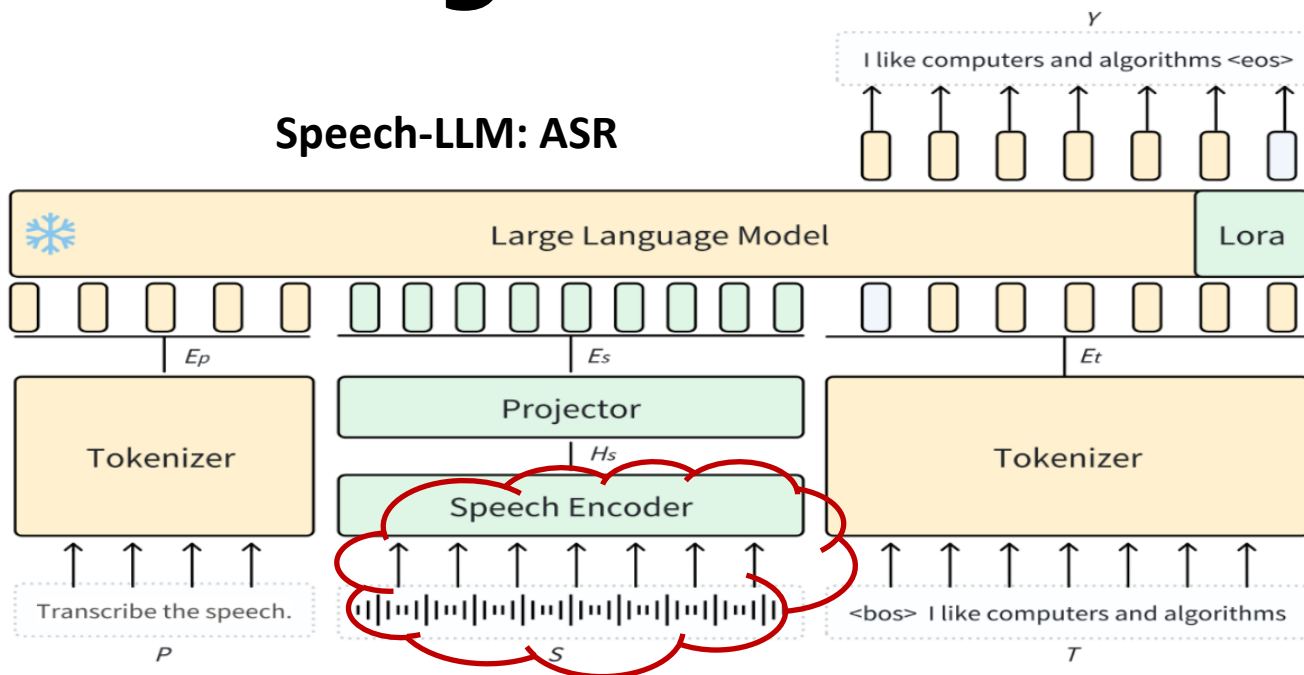
The Principled Semi-supervised Learning Philosophy Behind GPT's Success



In GPT, based on an autoregressive language model, **the supervised objective** naturally becomes part of **the unsupervised objective**, achieving principled unsupervised learning!

Challenge: cross-modality tasks

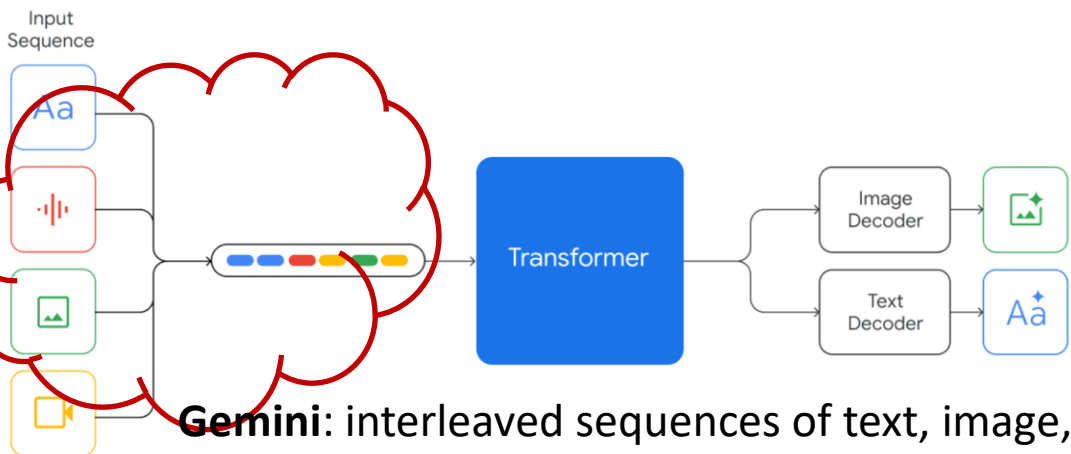
Speech-LLM: ASR



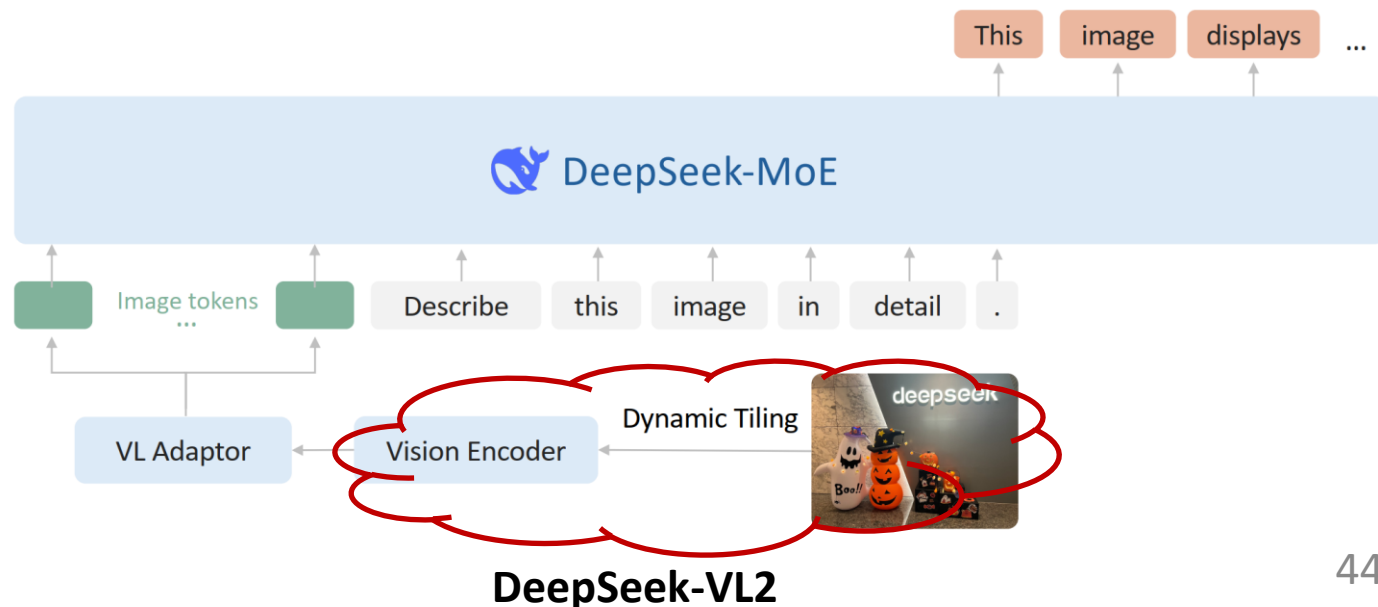
Many speech and image tasks are **cross-modal**; for example, ASR maps from the speech modality to the text modality and faces a new challenge:

Principled unsupervised learning is broken 😞

- Encoder and LLM are separately trained, losses **unrelated**.
- What should the **interface** between the two modalities be? How to fuse for cross-modality?



Gemini: interleaved sequences of text, image, audio, and video as **input**, responses with interleaved image and text as **output**



Subsection summary

1.3 Introduction to large language models

- **Autoregressive language models** learn by next token prediction.
- **Generative Pre-Training (GPT)** “broadly falls under the category of semi-supervised learning for natural language” (pretraining + finetuning).
- **The key advantage of GPT: achieving principled unsupervised learning** —— A unified token sequence can represent a task input and its output; Conditional task prediction fits naturally within language modeling!
- **LLM-ASR adds a cross-modal challenge.** Separately pretrained speech encoders and LLMs need an interface and objective that connect acoustics with text.

The central question for LLM-ASR is how to connect pretrained acoustic representations with language modeling.

Content

I. Basics of Speech Recognition and LLMs (50 min)

1. Modern speech recognition models
2. Data-efficiency in Multilingual and Crosslingual Speech Recognition
3. Introduction to LLMs
4. Overview and taxonomy of LLM-ASR

II. Continuous-interface based approach (50 min)

III. Discrete-interface based approach (50 min)

IV. Open questions and future directions (20 min)

1.4 Overview of LLM-ASR

- **LLM-ASR** combines a speech encoder and an LLM through an explicit speech–language interface.
- **Continuous interfaces** use a connector or projector to map acoustic features into the LLM embedding space.
- **Discrete interfaces** pass text hypotheses or phonemes to the LLM for error correction or phoneme-to-grapheme generation.
- **System design** also depends on encoder choice, training data and adaptation. Component ablations and full-system benchmarks answer different questions.

Fair interface comparisons require controlled backbones, data budgets and adaptation protocols.

Advances in Data-Efficient Multilingual and Crosslingual Speech Recognition Powered by LLMs

Continuous interface for Speech LLM

Presenters:

CHNG Eng Siong

Nanyang Technological University

Dr. Hexin Liu

Nanyang Technological University

ROADMAP — Part 1 (Chng Eng Siong)

1 Using LLM for ASR ~5 mins

2A Encoders for Speech ASR – A Primer ~10 mins

2B Speech LLM for ASR —integrating Speech Encoder, Connector and LLM for ASR ~10 mins

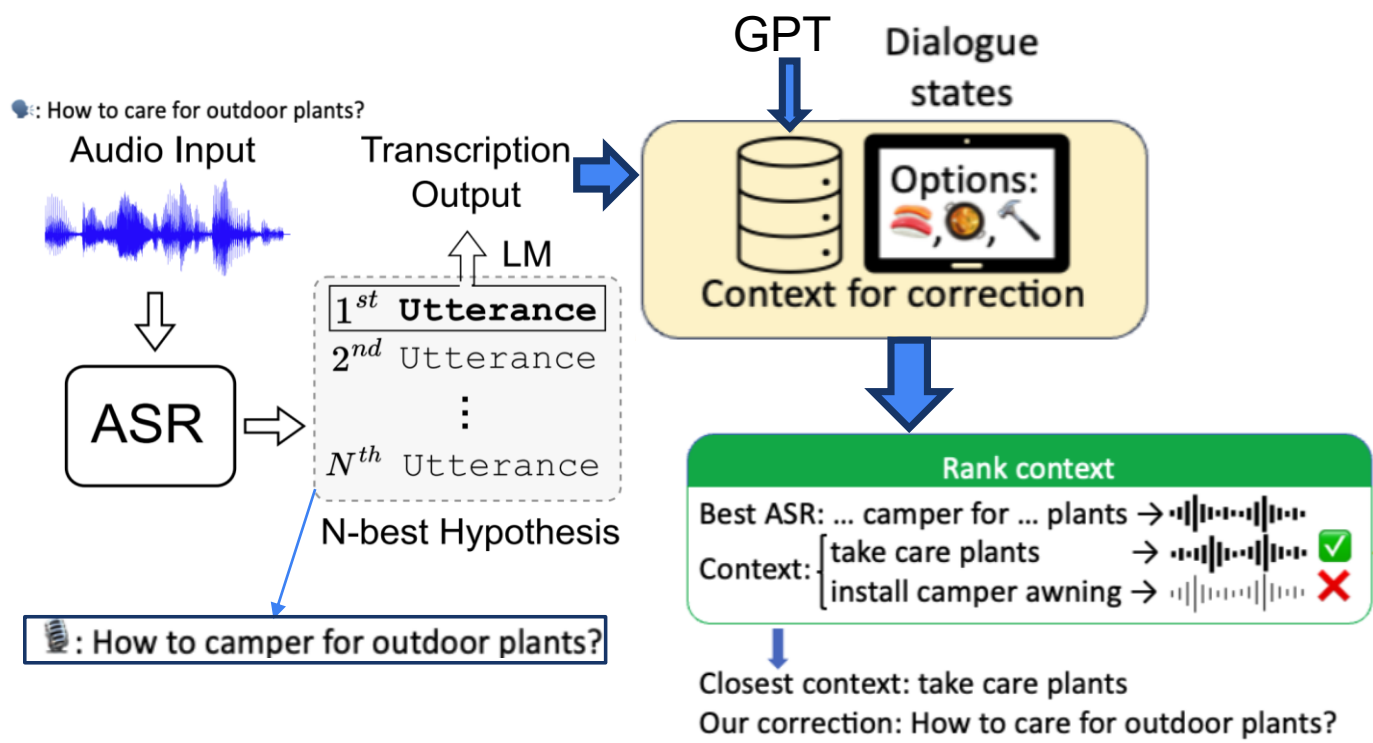
ROADMAP — Part 2 (Hexin)

3 Speech LLM for ASR - Continuous Interface ~10 mins

4 Data-efficient ASR with Speech LLM & Continuous Interface ~15 mins

1. Using LLM for ASR

Using LLM to perform error correction directly on text



Key Idea: speech decoded by traditional ASR. The set of N-Hypothesis is processed by LLM with context to perform error correction and hypothesis re-ranking.

HyPoradise: An Open Baseline for Generative Speech Recognition with Large Language Models

Chen Chen^{1,†} Yuchen Hu^{1,†} Chao-Han Huck Yang^{2*}
Sabato Marco Siniscalchi^{2,3} Pin-Yu Chen⁴ Eng Siong Chng¹

From: Huck Yang [view email]

[v1] Wed, 27 Sep 2023 14:44:10 UTC (1,092 KB)

[v2] Mon, 16 Oct 2023 05:47:42 UTC (1,093 KB)

ASR Error Correction using Large Language Models

Rao Ma*, Mengjie Qian*, Mark Gales, *Fellow, IEEE*, Kate Knill, *Senior Member, IEEE*

[v1] Sat, 14 Sep 2024 :

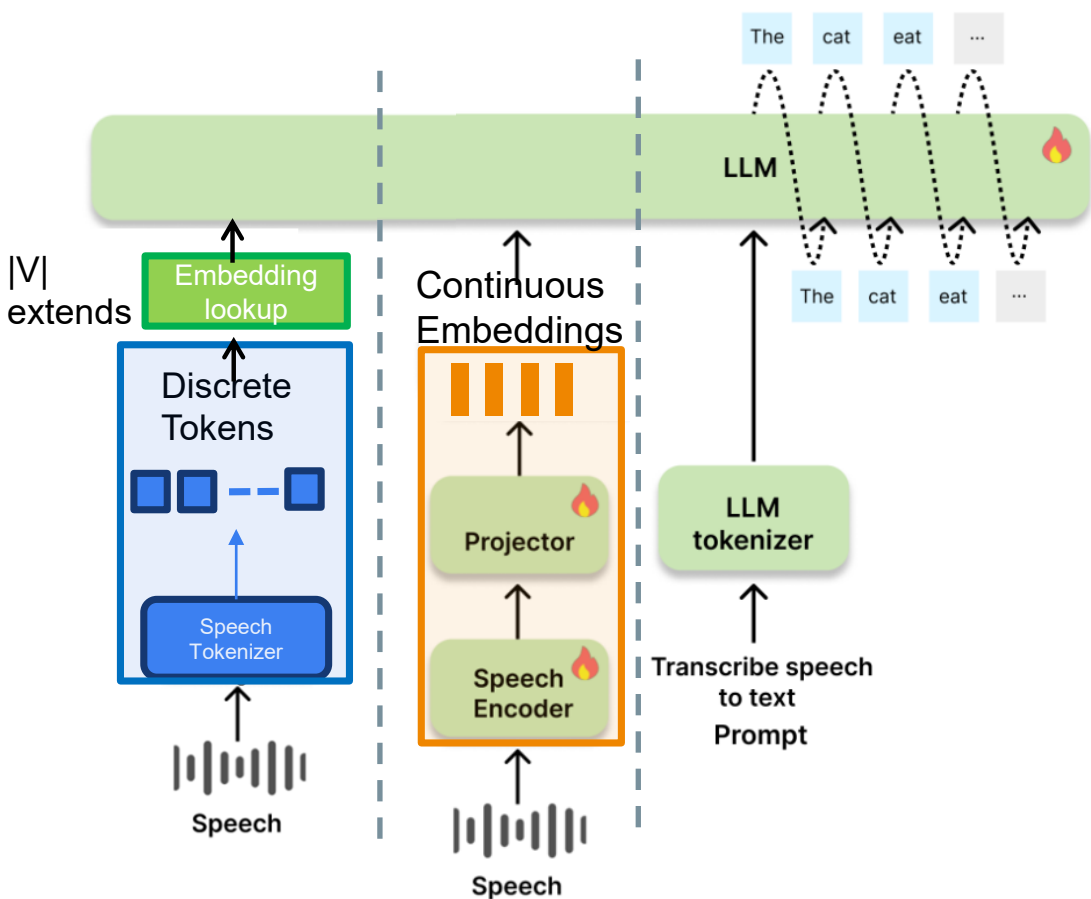
Contextual ASR Error Handling with LLMs Augmentation for Goal-Oriented Conversational AI

Yuya Asano¹, Sabit Hassan¹, Paras Sharma¹, Anthony Sicilia², Katherine Atwell²,
Diane Litman¹, and Malihe Alikhani²

[v1] Fri, 10 Jan 2025 1

1. Using LLM for ASR

Inputting continuous vs discrete representations to LLM for ASR



Discrete Speech Tokens

Discrete Tokens

GLM-4-Voice: An end-to-end speech based Large Language Model (LLM)

3 Dec 2024

Moshi: a speech-text foundation model for real-time dialogue

[v1] Tue, 17 Sep 2024

SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities

18 May 2023

continuous representations

Continuous Embeddings

When Large Language Models Meet Speech: A Survey on Integration Approaches

[v1] Wed, 26 Feb 2025

Zhengdong Yang, Shuichiro Shimizu, Yahan Yu, Chenhui Chu

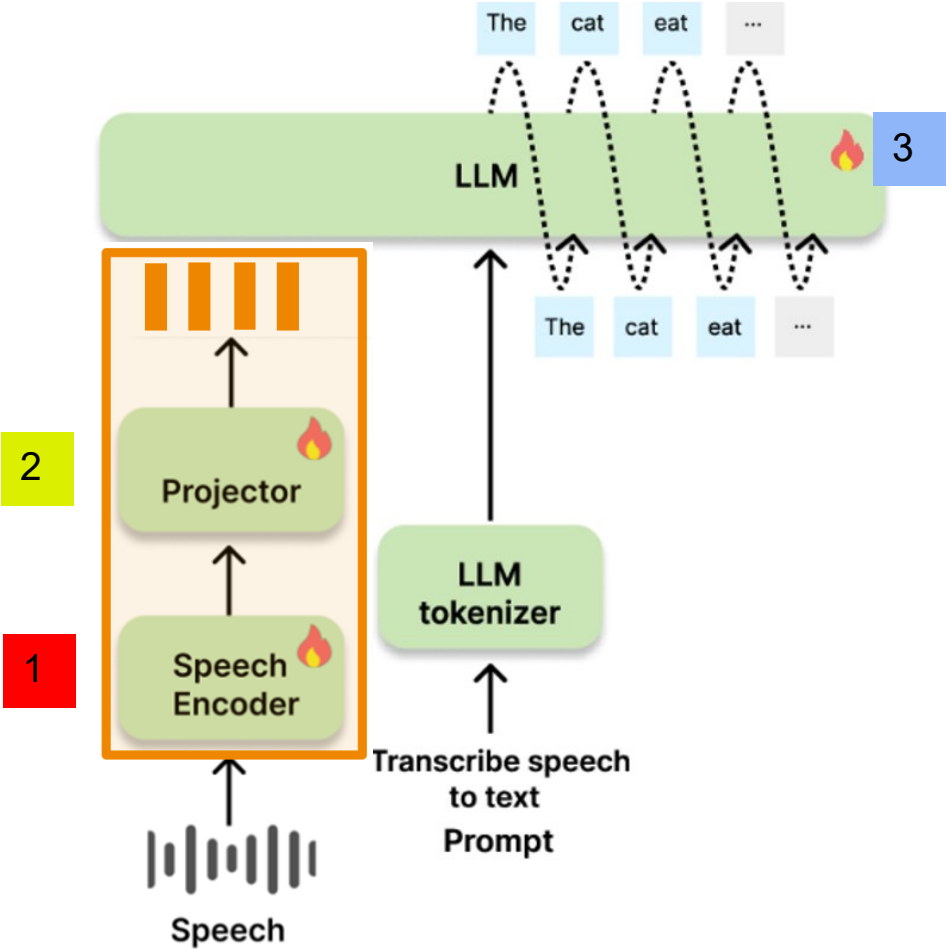
Qwen3-ASR Technical Report
Qwen3-Omni Technical Report

From: Jin Xu [view email]
[v1] Thu, 29 Jan 2026 06:58:13 UTC (2,049 KB)
[v2] Fri, 30 Jan 2026 02:58:48 UTC (2,051 KB)

Qwen Team

From: Jin Xu [view ema]
[v1] Mon, 22 Sep 2025

1.2 Using Continuous Representations for LLM ASR



Qwen3-ASR Technical Report

Qwen Team

 <https://huggingface.co/collections/Qwen/qwen3-asr>

SLAM-LLM: A Modular, Open-Source Multimodal Large Language Model Framework and Best Practice for Speech, Language, Audio and Music Processing

Ziyang Ma, Guanrou Yang, Wenxi Chen, Zhifu Gao, Yexing Du, Xiquan Li, Zhisheng Zheng, Haina Zhu, Jianheng Zhuo, Zhesu Song, Ruiyang Xu, Tiranrui Wang, Yifan Yang, Yanqiao Zhu, Zhikang Niu, Liumeng Xue, Yinghao Ma, Ruibin Yuan, Shiliang Zhang, Kai Yu, Eng Siong Chng, Xie Chen

CONNECTING SPEECH ENCODER AND LARGE LANGUAGE MODEL FOR ASR

Wenyi Yu¹, Changli Tang¹, Guangzhi Sun¹, Xianzhao Chen², Tian Tan², Wei Li², Lu Lu², Zejun Ma², Chao Zhang^{1,*}

¹ Department of Electronic Engineering, Tsinghua University, ² ByteDance

ML-SUPERB 2.0: Benchmarking Multilingual Speech Models Across Modeling Constraints, Languages, and Datasets

Jiatong Shi¹, Shih-Heng Wang^{2,*}, William Chen^{1,*}, Martijn Bartelds^{3,*}, Vanya Bannihatti Kumar¹, Jinchuan Tian¹, Xuankai Chang¹, Dan Jurafsky³, Karen Livescu^{3,4}, Hung-yi Lee², Shinji Watanabe¹

INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

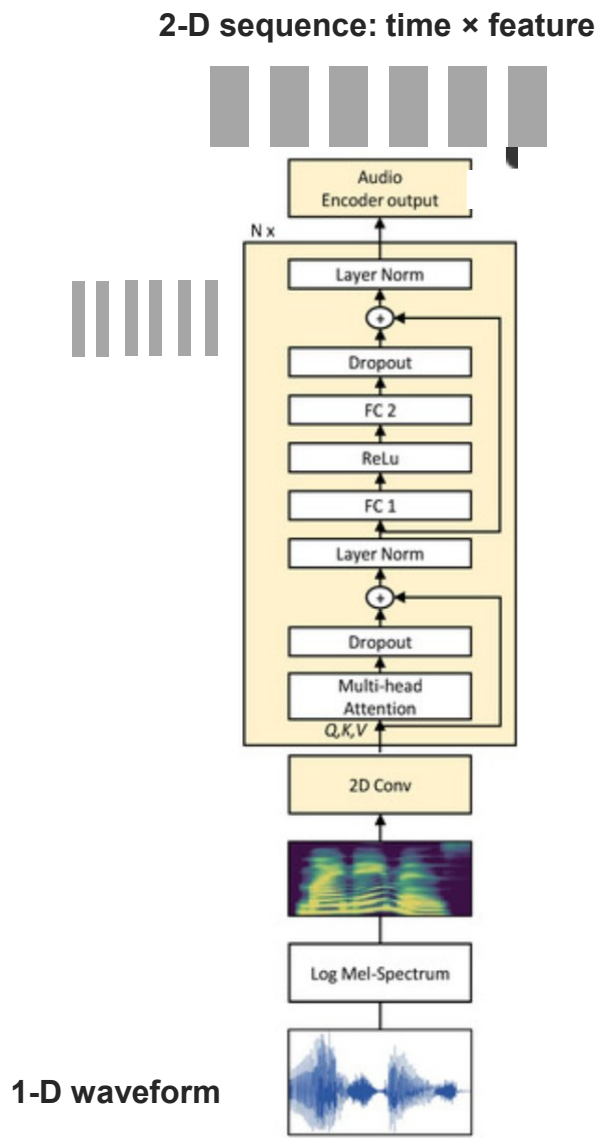


ML-SUPERB: Multilingual Speech Universal PERFORMANCE Benchmark

Jiatong Shi¹, Dan Berrebbi^{1,*}, William Chen^{1,*}, Ho-Lam Chung^{2,*}, En-Pei Hu^{2,*}, Wei Ping Huang^{2,*}, Xuankai Chang¹, Shang-Wen Li³, Abdelrahman Mohamed⁴, Hung-yi Lee², Shinji Watanabe¹

2A. Encoders in the LLM-ASR Pipeline: A Primer

A typical Encoder based on Transformer



The encoder's one job

Map a raw 1-D signal into a 2-D sequence of continuous vectors — one vector per time step, each summarising the acoustic / phonetic content around that moment.

What it does NOT do

- ✗ Produce text — that is the LLM's / decoder's job
- ✗ Produce phonemes or any linguistic symbol (partial exception: supervised models that ship their own decoder)
- ✗ Decide sequence length for the LLM — that is the connector's job

Why this framing matters: every encoder in this tutorial — SSL or supervised, 20 ms or 12.5 Hz — has exactly this same input/output contract. What differs is how it was trained, not what it emits.

2A.1 Encoder Characteristics: Size, Frame Rate, Data, Loss

Different front-ends, different objectives — but a striking convergence on frame rate

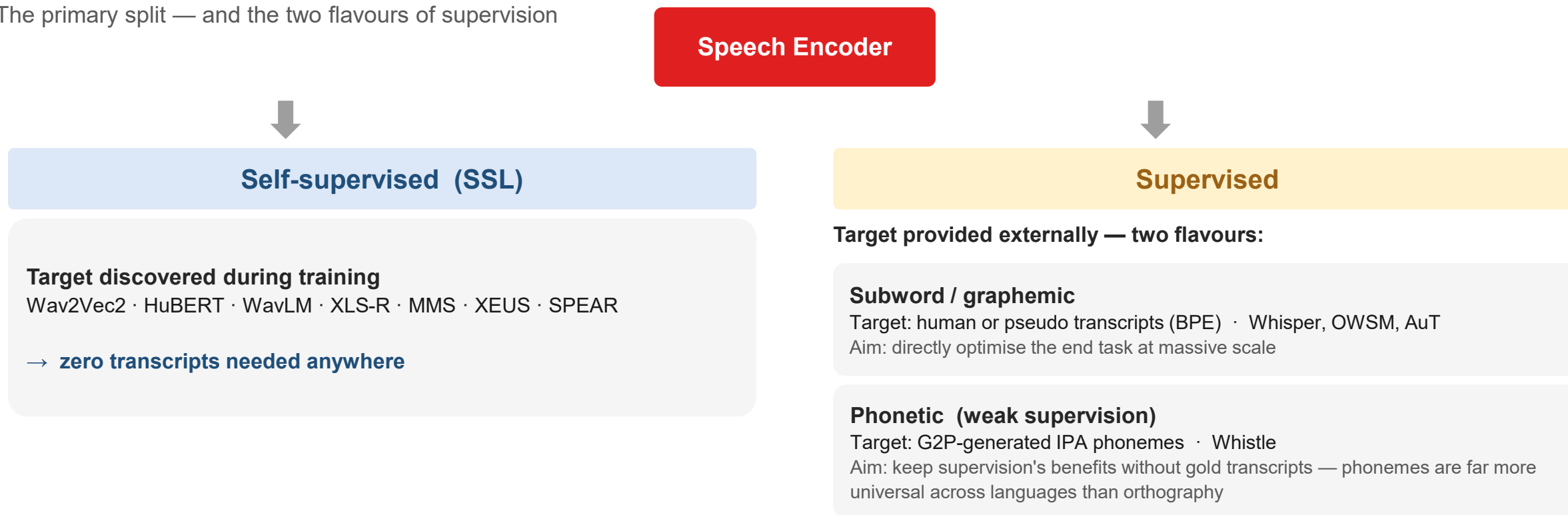
Encoder	Input	Frame rate	Layers	Hidden dim	Pre-train data	Training loss
Wav2Vec2 / HuBERT / WavLM	Raw waveform	~20 ms (50 Hz)	12 / 24 / 48	768 / 1024 / 1280	960 h → 94k h	Contrastive (W2V2) or masked-prediction classification (HuBERT / WavLM)
XLS-R / MMS	Raw waveform	~20 ms (50 Hz)	24 / 48	1024 / 1280	436k h / 491k h	Contrastive — scaled to 128 / 1,406 languages
Whisper (large)	80-ch log-Mel (25 ms win / 10 ms hop)	20 ms (50 Hz)	32	1280	680k h weak labels	Cross-entropy on human transcripts (BPE), jointly with its own decoder
Whistle (S / M / L)	80-dim FBank	~20 ms	14 / 22 / 22	512 / 640 / 1024	CV-Lang10 (10 languages)	CTC against G2P-generated IPA phonemes (weak phonetic supervision), 73-phoneme set
AuT (Qwen3-ASR)	128-dim FBank	12.5 Hz (8× downsample)	not disclosed	not disclosed	~40M h pseudo-lab.	AED, supervised — teacher-generated transcripts

Self-supervised Supervised

The convergence worth noticing: despite completely different front-ends — a learned CNN on raw waveform vs. hand-engineered log-Mel / FBank — Wav2Vec2 / HuBERT / WavLM, XLS-R / MMS, Whisper and Whistle all land on the same ~20 ms / 50 Hz output rate: the field's empirical sweet spot for phonetic content. AuT's 12.5 Hz is the outlier, trading temporal resolution for compute at 40M-hour scale.

2A.2 Classifying Encoders: What Supervises the Output?

The primary split — and the two flavours of supervision



The axis that matters for this tutorial:

- 1) What separates these families is which signal shaped the output during pre-training, and therefore how cheaply you can extend to a new low-resource language.
- 2) SSL needs no transcripts but no linguistic grounding either; graphemic supervision needs transcripts you may not have; phonetic supervision is the middle path built precisely for the low-resource case.

2A.3 Is Every Encoder a Transformer?

Mostly —

The shared skeleton — every encoder here follows it: front-end (learned CNN on waveform, or FBank / log-Mel) → stack of context-encoding layers → continuous frame embeddings

What varies is only what sits inside those context-encoding layers:

Standard Transformer	Conformer	E-Branchformer	Zipformer
<i>the default</i>	<i>interleaved</i>	<i>parallel, merged</i>	<i>multi-resolution</i>
Wav2Vec2 · HuBERT · WavLM XLS-R · MMS · Whisper · AuT	Whistle	XEUS	SPEAR
Self-attention + FFN per layer. Convolution appears only in the front-end, never inside a layer.	Each layer runs attention AND a convolution module in sequence, sandwiched between two FFNs.	Two parallel branches per layer — one attention, one convolutional gating — then merged.	Downsamples and upsamples the sequence inside the network — not one fixed frame rate.
HuBERT and its multilingual descendants reuse Wav2Vec2's backbone unchanged.	Local acoustic detail (conv) plus long-range context (attention) at every single layer.	XEUS ablation: swapping it in improved multilingual CER 30.5 → 29.0, LID 84.3 → 85.2.	Chosen over a plain Transformer for better ASR fine-tuning at lower compute.

Recent Encoders : Mamba based

AN INVESTIGATION OF INCORPORATING MAMBA FOR SPEECH ENHANCEMENT

Rong Chao^{‡§}, Wen-Huang Cheng[§], Moreno La Quatra[†]
Sabato Marco Siniscalchi⁺, Chao-Han Huck Yang^{*}, Szu-Wei Fu^{*}, Yu Tsao[†]

May, 2024

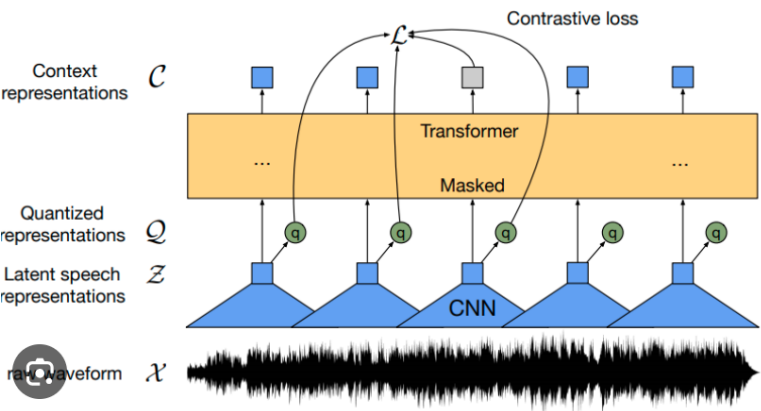
2A.4 Inside Self-Supervision: Contrastive vs. Masked Prediction

Architecture of Encoders look identical — both mask and 'fill in the blank' — but the prediction task is fundamentally different

Contrastive — Wav2Vec2

“Which of these candidates is the real masked segment?”
a **DISCRIMINATION** task

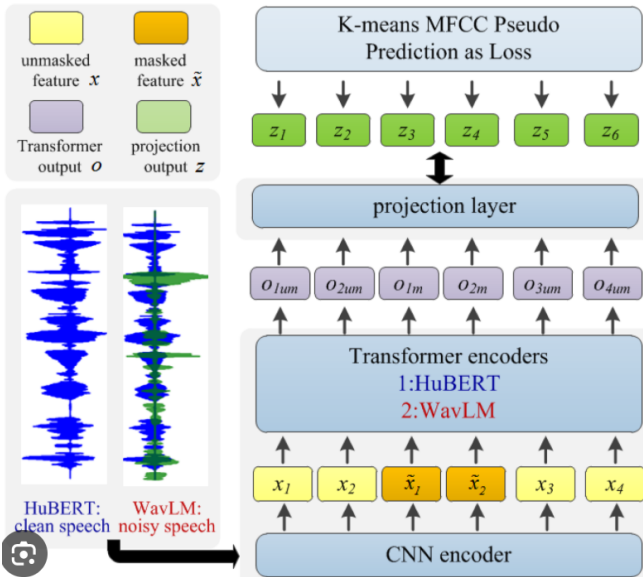
- Codebook is learned inside the model (Gumbel-softmax quantizer)
- Targets + encoder shaped by the same gradient, every step
- Loss: pick the true target from a lineup of distractors (InfoNCE)



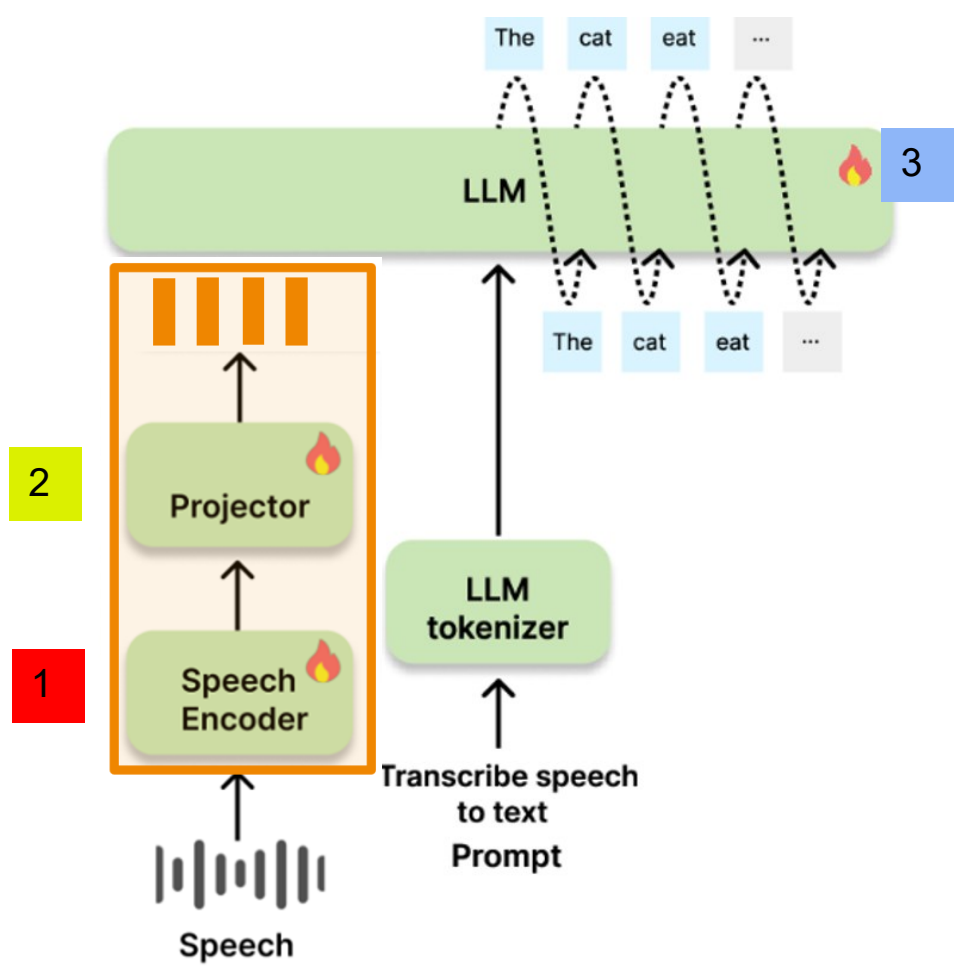
Masked Prediction — HuBERT / WavLM

“Which of these fixed categories is this segment?”
a **CLASSIFICATION** task

- Codebook = offline k-means, run before training starts
- Targets are frozen integers — no gradient ever flows back to them
- Loss: cross-entropy over a fixed class list (BERT-style)



2B.1 Integrating Encoder → Connector → LLM: Five Studies, Two Acts



1 2 3 Qwen3-ASR Technical Report
Qwen Team
 <https://huggingface.co/collections/Qwen/qwen3-asr>

Act 2

1 2 3 SLAM-LLM: A Modular, Open-Source Multimodal Large Language Model Framework and Best Practice for Speech, Language, Audio and Music Processing
Ziyang Ma, Guanrou Yang, Wenxi Chen, Zhifu Gao, Yexing Du, Xiquan Li, Zhisheng Zheng, Haina Zhu, Jianheng Zhuo, Zhesu Song, Ruiyang Xu, Tiranrui Wang, Yifan Yang, Yanqiao Zhu, Zhikang Niu, Liumeng Xue, Yinghao Ma, Ruibin Yuan, Shiliang Zhang, Kai Yu, Eng Siong Chng, Xie Chen

2 3 CONNECTING SPEECH ENCODER AND LARGE LANGUAGE MODEL FOR ASR
Wenyi Yu¹, Changli Tang¹, Guangzhi Sun¹, Xianzhao Chen², Tian Tan², Wei Li², Lu Lu², Zejun Ma², Chao Zhang^{1,*}
¹ Department of Electronic Engineering, Tsinghua University, ² ByteDance

1 ML-SUPERB 2.0: Benchmarking Multilingual Speech Models Across Modeling Constraints, Languages, and Datasets
Jiatong Shi¹, Shih-Heng Wang^{2,*}, William Chen^{1,*}, Martijn Bartelds^{3,*}, Vanya Bannihatti Kumar¹, Jinchuan Tian¹, Xuankai Chang¹, Dan Jurafsky³, Karen Livescu^{3,4}, Hung-yi Lee², Shinji Watanabe¹

Act 1

1 ML-SUPERB: Multilingual Speech Universal PERFORMANCE Benchmark
Jiatong Shi¹, Dan Berrebbi^{1,*}, William Chen^{1,*}, Ho-Lam Chung^{2,*}, En-Pei Hu^{2,*}, Wei Ping Huang^{2,*}, Xuankai Chang¹, Shang-Wen Li³, Abdelrahman Mohamed⁴, Hung-yi Lee², Shinji Watanabe¹

2B.1 Integrating Encoder → Connector → LLM: Five Studies, Two Acts

Same three-part pipeline throughout — each study fixes a different piece, at a different resource scale

ACT 1 — Controlled Research: isolate one variable at a time

1	ML-SUPERB 1.0	Encoder only — no connector, no LLM anywhere in the pipeline
1	ML-SUPERB 2.0	+ encoder adaptation (frozen→LoRA) + decoder architecture variety — still no LLM
2 3	Yu et al. 2024	Encoder & LLM held fixed — connector architecture isolated (Q-Former wins)

ACT 2 — Real Systems: build the full pipeline, at scale

1 2 3	SLAM-LLM	The toolkit — encoder, connector, and LLM all real and configurable via YAML
	Qwen3-ASR	Production scale — 40M hrs + 3T tokens + RL; connector present, but unspecified

How they weave together:

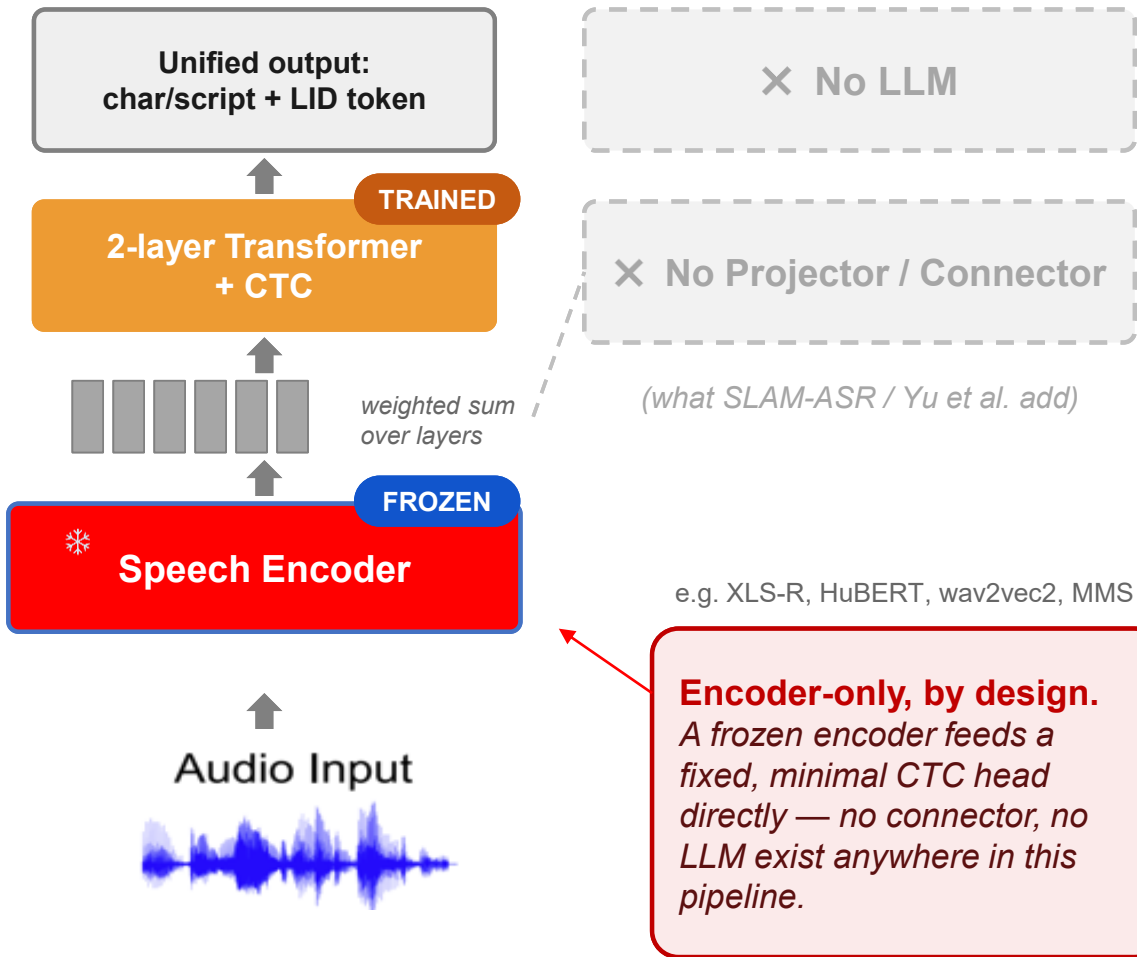
Act 1 answers “does this piece matter?” in isolation, one variable at a time.

Act 2 asks what happens once you stop isolating and just build the whole thing — does Act 1's careful answer survive contact with a real, resource-rich system?

2B.2 ML-SUPERB 1.0: Architecture and Setup

1

Architecture



Setup

Languages	143 total — 123 “normal” + 20 reserved for few-shot, across 15 corpora
Data scale	10-min / 1-hour splits (main track); only 5 utterances for the 20 few-shot languages
Tracks	Multilingual ASR + LID (main); separate Monolingual ASR track (9 languages, 14 experiments)
Metric	CER for ASR, Accuracy for Language ID (LID)
Run as	ASRU 2023 shared-task Challenge — 12 model submissions, 8 new-language submissions
Compute cost	≈ 3 days for a full evaluation cycle, on 4× RTX 2080Ti GPUs

Encoder is the only variable: same downstream head for every submission.

2B.2 ML-SUPERB 1.0: Results

1

INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland



ML-SUPERB: Multilingual Speech Universal PERFORMANCE Benchmark

Jiatong Shi¹, Dan Berrebbi^{1*}, William Chen^{1*}, Ho-Lam Chung^{2*}, En-Pei Hu^{2*}, Wei Ping Huang^{2*},
Xuankai Chang¹, Shang-Wen Li³, Abdelrahman Mohamed⁴, Hung-yi Lee², Shinji Watanabe¹

FINDINGS OF THE 2023 ML-SUPERB CHALLENGE: PRE-TRAINING AND EVALUATION OVER MORE LANGUAGES AND BEYOND

Jiatong Shi¹, William Chen¹, Dan Berrebbi¹, Hsiu-Hsuan Wang², Wei-Ping Huang²,
En-Pei Hu², Ho-Lam Chuang², Xuankai Chang¹, Yuxun Tang³, Shang-Wen Li⁴,
Abdelrahman Mohamed⁵, Hung-yi Lee², Shinji Watanabe¹

From: Jiatong Shi [view email]
[v1] Thu, 18 May 2023 00:01:27 UTC (107 KB)
[v2] Fri, 11 Aug 2023 17:39:21 UTC (107 KB)
[v3] Mon, 24 Feb 2025 18:06:55 UTC (109 KB)

Table 4: 1-hour set ML-SUPERB benchmark.

SSL	Monolingual ASR	Multilingual ASR		LID	Multilingual ASR + LID			SUPERB _s
	CER/PER	Normal CER	Few-shot CER	Normal ACC	Normal ACC	CER	Few-shot CER	
FBANK	63.7	59.3	57.4	9.3	43.5	58.6	58.1	0
wav2vec2-base [3]	35.9	35.5	44.3	80.8	83.6	32.1	42.6	827.2
wav2vec2-large [3]	35.4	35.7	43.9	8.0	78.2	34.7	42.2	586.9
robust-wav2vec2-large [41]	35.7	31.1	42.2	72.1	62.9	33.7	46.0	768.6
wav2vec2-base-23 [19]	35.1	32.0	42.2	71.9	66.3	30.9	43.0	798.0
wav2vec2-large-23 [19]	34.2	35.3	42.4	64.2	49.7	35.2	43.1	724.9
XLSR-53 [7]	34.9	26.9	40.6	87.1	76.9	28.6	44.6	894.0
XLSR-128 [6]	30.6	22.0	39.3	87.9	85.6	22.9	42.4	996.0
HuBERT-base [4]	35.3	31.4	42.7	86.1	86.0	30.9	41.8	884.9
HuBERT-large [4]	32.2	37.7	43.5	64.1	77.7	35.1	42.2	783.6
HuBERT-base-cmn [42]	35.6	43.2	46.6	85.3	86.1	31.8	42.1	810.2
HuBERT-large-cmn [42]	33.7	39.6	45.1	57.3	75.6	37.1	44.4	713.2
mHuBERT-base [43]	33.0	33.4	43.6	72.5	70.9	29.7	43.1	812.7

2023 findings:

Multi-Lingual ASR:
Testing different encoders with fixed downstream head
(2-layer Transformer + CTC, no LM)

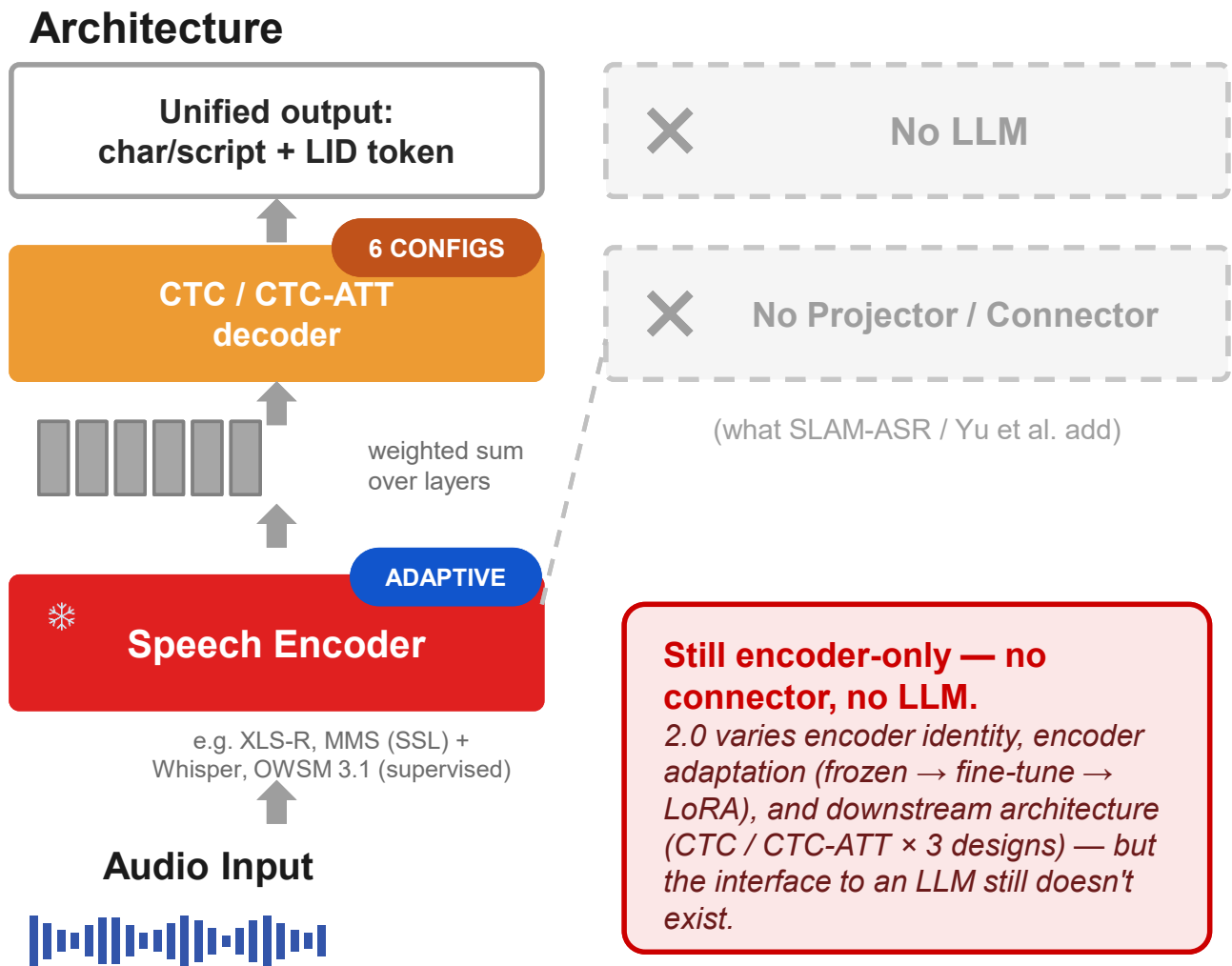
Takeaways:
• Broader multilingual pretraining helps: **XLSR-128** (128 languages) gets the best CER and LID.

Table 3. {10-minute / 1-hour} set ML-SUPERB public benchmark (143 languages).

SSL	Monolingual ASR	Multilingual ASR		LID	Multilingual ASR + LID			SUPERB _s
	CER/PER	Normal CER	Few-shot CER	Normal ACC	Normal ACC	CER	Few-shot CER	
FBANK	72.1 / 63.7	62.4 / 59.3	58.3 / 57.4	11.1 / 9.3	35.9 / 43.5	62.0 / 58.6	58.9 / 58.1	0 / 0
wav2vec2-base [3]	44.2 / 35.9	43.0 / 35.5	45.7 / 44.3	54.4 / 80.8	66.9 / 83.6	40.6 / 32.1	44.2 / 42.6	590.4 / 707.8
wav2vec2-base-23 [52]	49.2 / 35.1	37.7 / 32.0	43.4 / 42.2	58.7 / 71.9	45.1 / 66.3	37.2 / 30.9	44.3 / 43.0	563.0 / 676.5
XLSR-128 [5]	39.7 / 30.6	29.2 / 22.0	40.9 / 39.3	66.9 / 87.9	55.6 / 85.6	28.4 / 22.9	42.1 / 42.4	734.9 / 854.2
HuBERT-base [4]	42.8 / 35.3	39.8 / 31.4	44.5 / 42.7	61.2 / 86.1	71.5 / 86.0	39.2 / 30.9	43.8 / 41.8	650.8 / 757.0
HuBERT-large [4]	38.2 / 32.2	44.4 / 37.7	48.2 / 43.5	46.5 / 64.1	55.4 / 77.7	45.6 / 35.1	49.3 / 42.2	541.8 / 659.8
mHuBERT-base [53]	41.0 / 33.0	40.5 / 33.4	45.6 / 43.6	52.4 / 72.5	46.6 / 70.9	36.8 / 29.7	44.2 / 43.1	580.3 / 692.1
MMS-300m	33.8 / 30.5	28.7 / 24.0	36.5 / 36.5	62.3 / 84.3	71.9 / 74.3	31.5 / 30.0	30.9 / 29.2	826.7 / 841.3
MMS-1b	33.3 / 25.7	21.3 / 18.1	30.2 / 30.8	84.8 / 86.1	73.3 / 74.8	26.0 / 25.5	25.4 / 24.8	983.5 / 943.2

2B.3 ML-SUPERB 2.0: Architecture and Setup

1



Setup

Languages	142 total — 122 “normal” + 20 few-shot (vs. 143/123 in 1.0)
Data scale	1-hour per language-dataset pair (main); only 5 utterances for few-shot
Tracks	Multilingual ASR + LID only — separate Monolingual track from 1.0 dropped
Metric	CER for ASR, Accuracy for Language ID (LID)
Param budget	≤100M tunable parameters for every configuration
New axes	Encoder: 4 models × 5 adaptations. Downstream: 2 frameworks × 3 architectures

Not just the encoder anymore: *adaptation and downstream design are variables too — but still no connector, no LLM.*

ML-SUPERB 2.0: Benchmarking Multilingual Speech Models Across Modeling Constraints, Languages, and Datasets

From: Jiatong Shi [view email]

[v1] Wed, 12 Jun 2024 21:01:26 UTC (75 KB)

Jiatong Shi¹, Shih-Heng Wang^{2,*}, William Chen^{1,*}, Martijn Bartelds^{3,*}, Vanya Bannihatti Kumar¹, Jinchuan Tian¹, Xuankai Chang¹, Dan Jurafsky³, Karen Livescu^{3,4}, Hung-yi Lee², Shinji Watanabe¹

Table 1: Results of the downstream architecture experiments, showing the downstream model, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). T., C., E-B. are abbreviations for Transformer, Conformer, and E-Branchformer. + indicates the use of the CTC-ATT framework. † refers to the original ML-SUPERB setting [13].

Models	Method	Param. (M)	ACC	CER		
				Normal	FS	WL
XLS-R	T.†	323.7 (6.3)	90.9	24.8 ± 12.1	34.4 ± 21.1	75.1
MMS	T.†	321.8 (6.3)	90.3	24.7 ± 12.3	31.0 ± 18.6	67.6
XLS-R	T.	408.5 (91.1)	93.7	20.7 ± 10.8	33.3 ± 20.8	68.0
	C.	408.9 (91.5)	82.3	22.9 ± 12.8	33.4 ± 20.5	86.9
	E-B.	409.6 (92.2)	94.1	18.2 ± 10.6	32.3 ± 20.9	69.5
	T.+	416.0 (98.6)	93.6	19.2 ± 11.9	33.6 ± 21.0	76.2
	C.+	416.3 (98.9)	83.7	23.9 ± 19.1	34.8 ± 22.6	102.9
	E-B.+	417.1 (99.7)	94.7	16.9 ± 10.7	32.3 ± 21.1	63.8
MMS	T.	406.6 (91.1)	93.6	21.0 ± 11.2	31.7 ± 19.3	67.4
	C.	407.0 (91.5)	85.3	22.7 ± 14.2	31.7 ± 17.7	94.6
	E-B.	407.7 (92.2)	93.0	20.4 ± 10.6	31.0 ± 19.1	61.5
	T.+	414.1 (98.6)	94.3	18.8 ± 11.8	31.9 ± 19.0	73.1
	C.+	414.4 (98.9)	84.0	23.8 ± 16.7	33.6 ± 18.5	106.1
	E-B.+	415.2 (99.7)	95.2	16.6 ± 11.8	32.6 ± 20.4	69.8

Table 4: Results of the supervised model experiments, showing whether fine-tuning (FT) is performed, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). Asterisks indicate that only the encoder is used. † refers to the original ML-SUPERB setting [13].

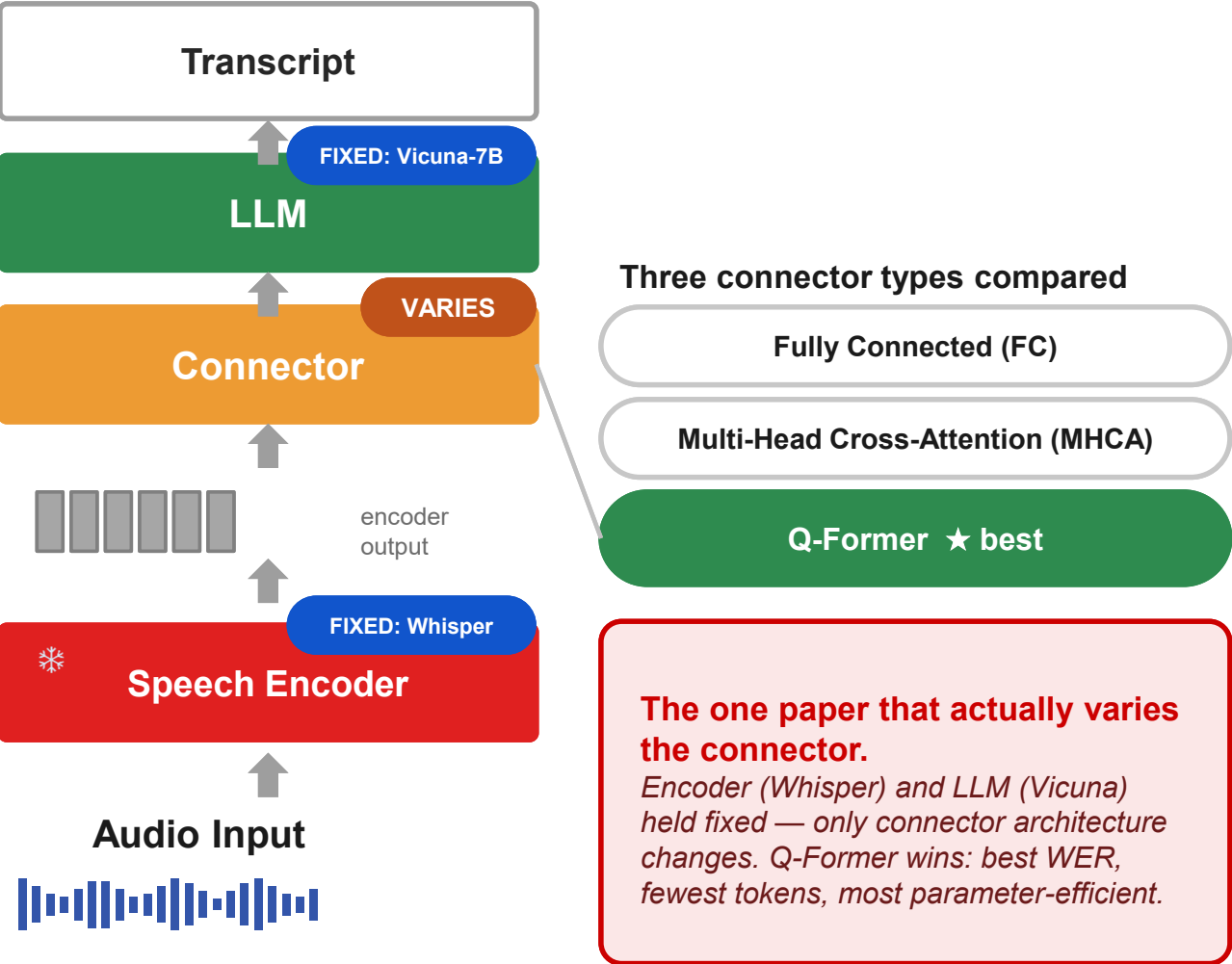
Models	FT	Param. (M)	ACC	CER		
				Normal	FS	WL
XLS-R	$\chi^\dagger$	323.7 (6.3)	90.9	24.8 ± 12.1	34.4 ± 21.1	75.1
MMS	$\chi^\dagger$	321.8 (6.3)	90.3	24.7 ± 12.3	31.0 ± 18.6	67.6
Whisper	χ^*	515.8 (91.1)	91.7	21.0 ± 12.5	27.4 ± 13.3	82.9
	✓*	431.0 (90.7)	83.9	26.8 ± 15.0	29.6 ± 13.5	93.5
	✓	762.3 (84.4)	85.5	25.6 ± 19.4	35.0 ± 17.5	107.2
OWSM	χ^*	671.2 (88.4)	77.8	27.8 ± 22.6	31.7 ± 17.3	99.9
	✓*	612.1 (88.4)	71.0	24.9 ± 14.9	31.5 ± 16.9	99.7
	✓	1016.9 (100.8)	80.5	40.0 ± 41.8	40.0 ± 24.9	337.6

2B.4 Yu et al. 2024: Architecture and Setup

2

3

Architecture



Setup

Fixed components	Whisper encoder + Vicuna-7B LLM — both frozen; only the connector varies
Connector types	Fully Connected (FC), Multi-Head Cross-Attention (MHCA), Q-Former
Training data	LibriSpeech-960 (English only)
Metric	WER (%), test-clean / test-other
Key results	FC(300tok): 2.29/5.44 · MHCA(75tok): 3.22/7.54 · Q-Former(80tok): 2.28/5.20 ★
Robustness check	Q-Former, zero-shot on Switchboard/Eval2000 — 12% relative WER reduction, out-of-domain

The connector-ablation study: the only paper in this series where encoder and LLM are both fixed and the connector itself is the variable.

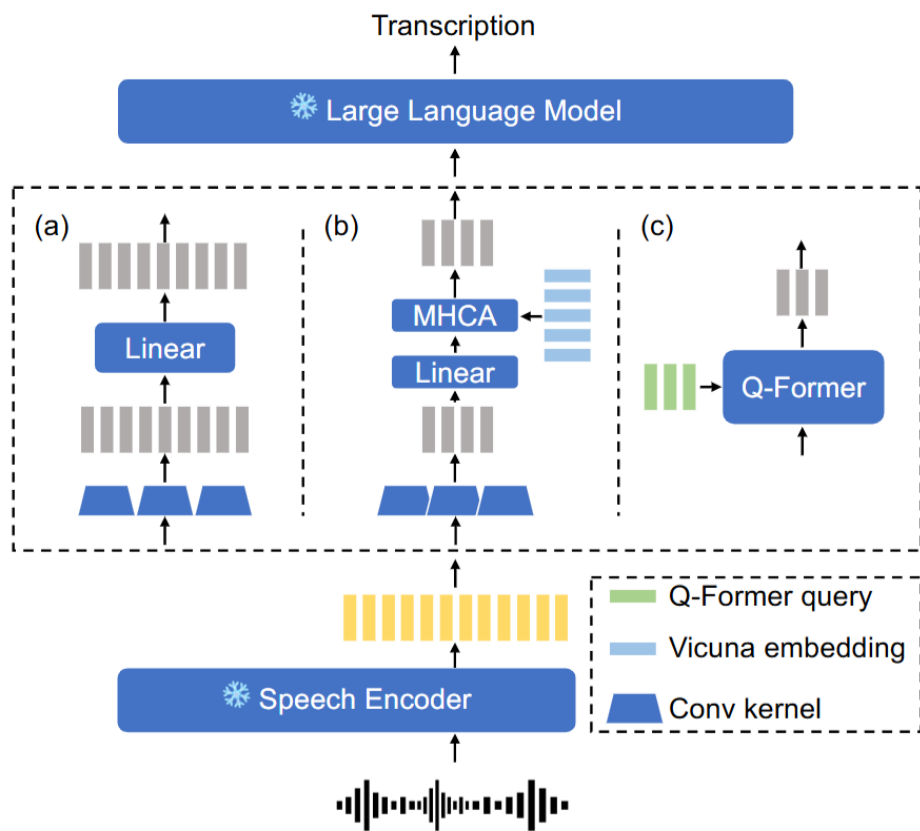


Fig. 1. Illustration of integrating a speech encoder and an LLM into an ASR system with a module connector of: (a) fully connected layers, (b) multi-head cross-attention, and (c) Q-Former.

Model	#Tokens	#Params	LibriSpeech	
			test-clean	test-other
FC	75	24.6M	3.00	6.70
FC	300	23.6M	2.29	5.44
CA	75	133.4M	3.22	7.54
QF	60	24.5M	2.33	5.43
QF	80	24.5M	2.28	5.20

Table 2. %WERs of LLMs with different connectors. FC, CA, and QF refer to fully connected layers, multi-head cross-attention, and Q-Former connectors respectively. #Tokens and #Params are the numbers of output tokens and model parameters of the connectors.

CONNECTING SPEECH ENCODER AND LARGE LANGUAGE MODEL FOR ASR

Wenqi Yu¹, Changli Tang¹, Guangzhi Sun¹, Xianzhao Chen²,
Tian Tan², Wei Li², Lu Lu², Zejun Ma², Chao Zhang^{1,*}

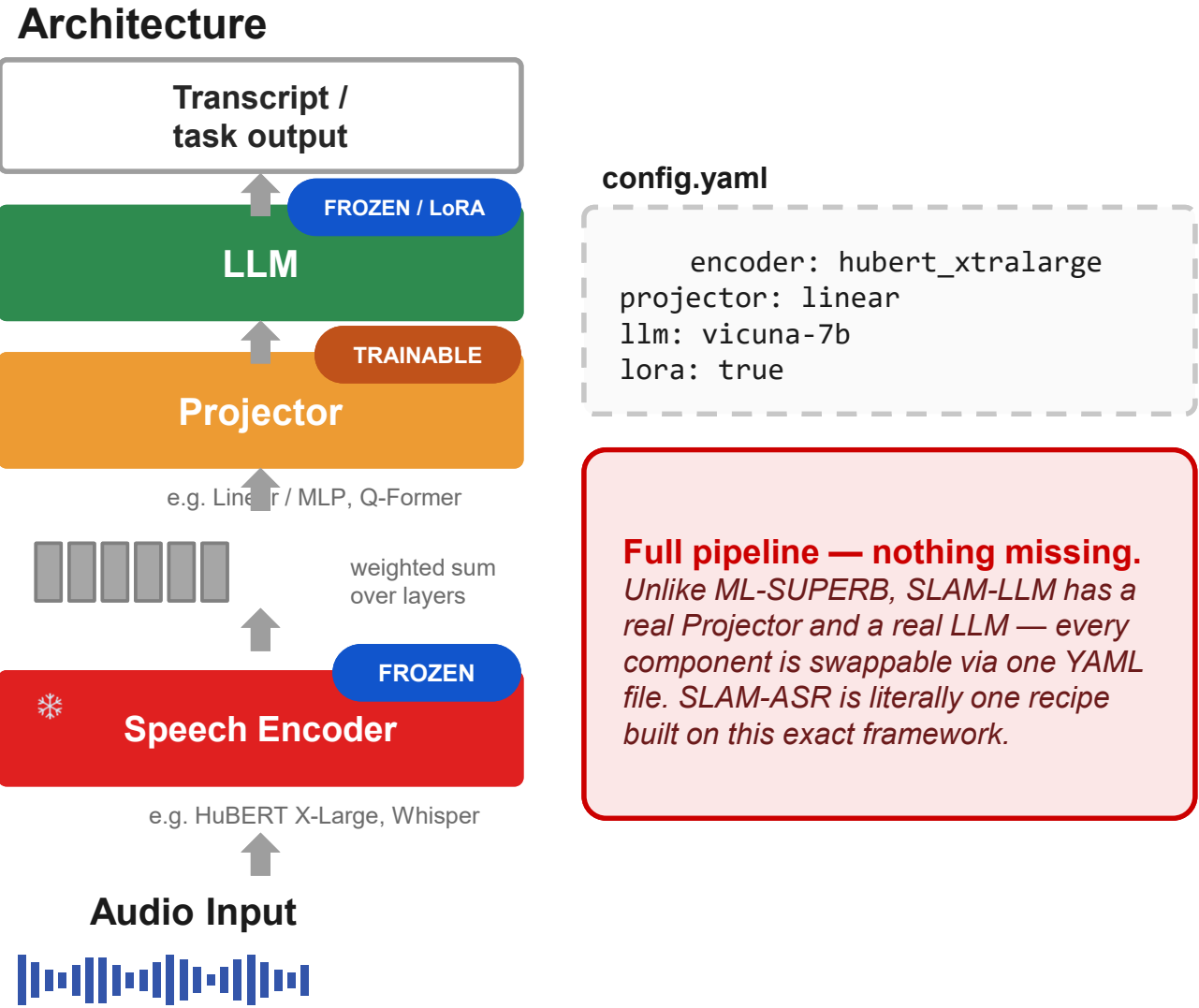
¹ Department of Electronic Engineering, Tsinghua University, ² ByteDance

From: Changli Tang [view email]

[v1] Mon, 25 Sep 2023 08:57:07 UTC (161 KB)

[v2] Tue, 26 Sep 2023 11:09:25 UTC (161 KB)

2B.5 SLAM-LLM: Architecture and Setup



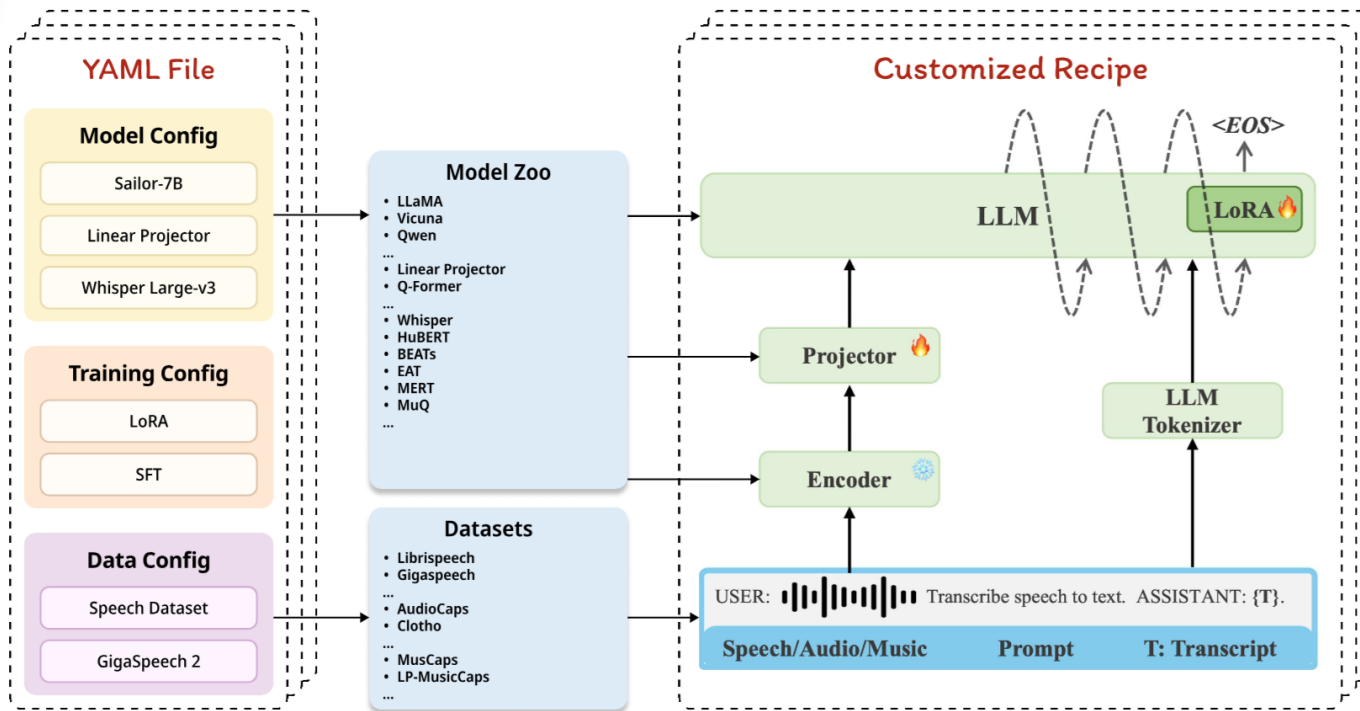
Setup

Framework	Modular, open-source, YAML/Hydra-configured Encoder → Projector → LLM(+LoRA) toolkit
Recipes	ASR (SLAM-ASR), contextual ASR, S2TT, TTS, captioning, voice dialogue (SLAM-Omni), and more
Reference cost	~4 hrs per 1,000h of speech, 1 epoch, 4×A100 — frozen encoder + LLM, projector-only training
Scale support	Up to 100,000-hour industrial training with dynamic frame batching (~4× speedup)
License / adoption	MIT license, ~1,000 GitHub stars, actively maintained
Low-resource example	SEA languages (Sailor-7B) — last 2/32 encoder layers fine-tuned + LoRA on the LLM

Not an ablation study — a toolkit. SLAM-ASR, low-resource SEA ASR, and many other recipes are all built on this one framework.

2B.5 SLAM-LLM: Architecture and Setup

- [Update Jan. 15, 2026] The [SLAM-LLM Framework](#) was accepted by IEEE Journal of Selected Topics in Signal Processing (JSTSP), where we put the framework design and some unpublished experiments and insights for speech, audio, and music MLLM.
- [Update Apr. 28, 2024] Recipes for [automated audio captioning \(AAC\)](#) have been supported.
- [Update Mar. 31, 2024] Recipes for [automatic speech recognition \(ASR\)](#) have been supported.



SLAM-LLM: A Modular, Open-Source Multimodal Large Language Model Framework and Best Practice for Speech, Language, Audio and Music Processing

Ziyang Ma, Guanrou Yang, Wenxi Chen, Zhifu Gao, Yexing Du, Xiquan Li, Zhisheng Zheng, Haina Zhu, Jianheng Zhuo, Zhesu Song, Ruiyang Xu, Tiranrui Wang, Yifan Yang, Yanqiao Zhu, Zhikang Niu, Liumeng Xue, Yinghao Ma, Ruibin Yuan, Shiliang Zhang, Kai Yu, Eng Siong Chng, Xie Chen

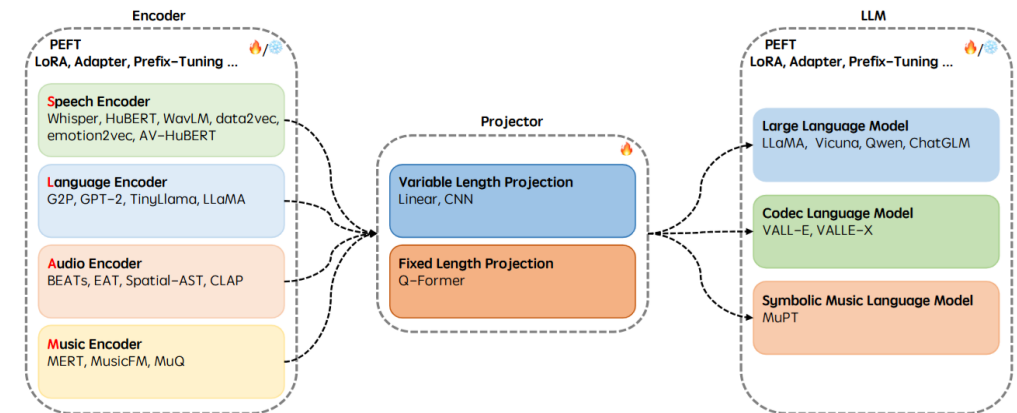


Fig. 2: SLAM-LLM modularizes the Encoder-Projector-LLM three-part components, which are assembled according to the configuration at training and inference time.

2B.5 SLAM-LLM: Results

1 2 3

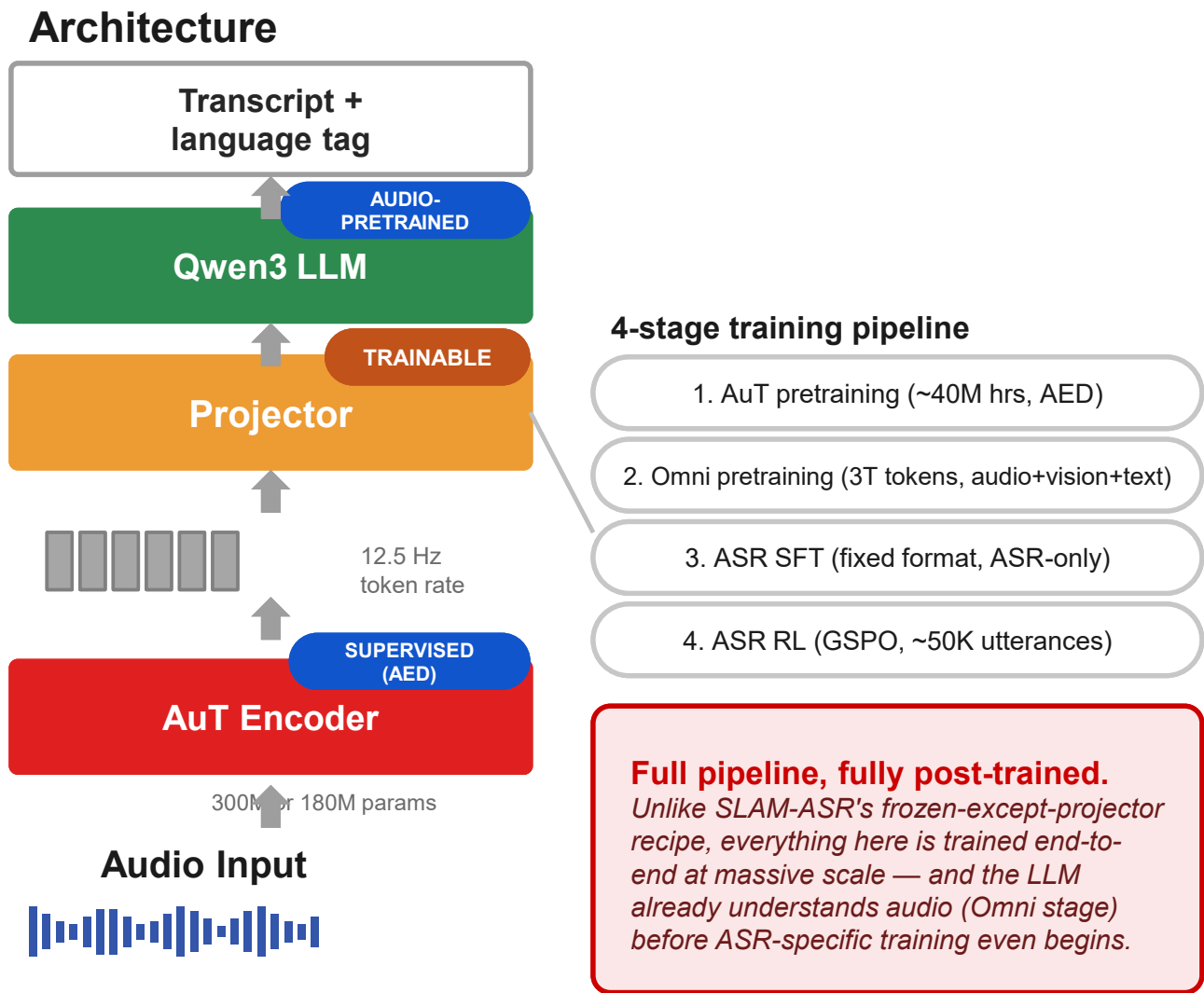
TABLE I: Different combinations of supervised speech encoders and LLMs

Speech Encoder	Chat Model			
	LLaMA-2-Chat-7B		Vicuna-v1.5-7B	
	clean	other	clean	other
Whisper-tiny	8.97	18.77	7.07	16.01
Whisper-base	6.37	12.98	5.07	13.07
Whisper-small	4.51	8.94	4.19	9.50
Whisper-medium	2.71	6.37	2.72	6.79
Whisper-large	2.72	6.79	2.58	6.47

TABLE II: Explore the performance with different SSL speech encoders for LLM-based ASR. LS-960 means the Librispeech 960 hours dataset. FT donates Fine-tuning.

Speech Encoder	Encoder Params	Hidden Size	Projector Params	WER(%) ↓	
				test-clean	test-other
HuBERT Base	94.70M	768	16.26M	4.43	10.72
WavLM Base	94.38M	768	16.26M	4.14	9.66
HuBERT Large	316.61M	1024	18.88M	4.53	8.74
+ LS-960 FT	316.61M	1024	18.88M	2.10	4.26
WavLM Large	315.45M	1024	18.88M	2.13	4.73
+ LS-960 FT	315.45M	1024	18.88M	1.96	4.18
HuBERT X-Large	964.32M	1280	21.50M	2.41	4.49
+ LS-960 FT	964.32M	1280	21.50M	1.84	3.39

2B.6 Qwen3-ASR: Architecture and Setup



Setup

Model sizes	Qwen3-ASR-1.7B (300M AuT + Qwen3-1.7B); Qwen3-ASR-0.6B (180M AuT + Qwen3-0.6B)
Languages	30 languages + 22 Chinese dialects (52 total); separate ForcedAligner covers 11 languages
Training scale	~40M hrs (encoder) + 3T tokens (Omni) + SFT/RL — far beyond SLAM-ASR/Yu et al.
Distinctive features	Unified streaming + offline (1–8s dynamic window); long-form to 20 min; singing/BGM songs
Efficiency	~92ms time-to-first-token; RTF 0.064; 2,000s audio/s at concurrency 128 (vLLM)
Key results	Beats Whisper-large-v3 broadly; LID 97.9% vs. 94.1%; competitive with GPT-4o-Transcribe

Scale doesn't fully solve it: even 40M hrs + 3T tokens + RL still loses ground to Whisper on the full 30-language Fleurs set — long-tail languages remain the open problem.

2B.6 Qwen3-ASR: Results



Table 3: Evaluation on English, Mandarin Chinese, and a range of Chinese dialect benchmarks. For the commercial APIs and the open-source Whisper-large-v3 model, we obtained results by running inference on the test sets ourselves due to the absence of published numbers; for FunASR-MLT-Nano, we report the results from its official technical report. "N/A" denotes that we cannot get a reasonable result by the official API. "-" indicates that the corresponding benchmark result is not reported.

	GPT-4o -Transcribe	Gemini-2.5 -Pro	Doubao-ASR	Whisper -large-v3	Fun-ASR -MLT-Nano	Qwen3-ASR -0.6B	Qwen3-ASR -1.7B
<i>English (en)</i>							
LibriSpeech <i>clean</i> <i>other</i>	1.39 3.75	2.89 3.56	2.78 5.70	1.51 3.97	1.68 4.03	2.11 4.55	1.63 3.38
GigaSpeech	25.50	9.37	9.55	9.76	-	8.88	8.45
CV-en	9.08	14.49	13.78	9.90	9.90	9.92	7.39
Fleurs-en	2.40	2.94	6.31	4.08	5.49	4.39	3.35
MLS-en	5.12	3.68	7.09	4.87	-	6.00	4.58
Tedlium	7.69	6.15	4.91	6.84	-	3.85	4.50
VoxPopuli	10.29	11.36	12.12	12.05	-	9.96	9.15
<i>Chinese (zh)</i>							
WenetSpeech <i>net</i> <i>meeting</i>	15.30 32.27	14.43 13.47	N/A	9.86 19.11	6.35 -	5.97 6.88	4.97 5.88
AISHELL-2-test	4.24	11.62	2.85	5.06	-	3.15	2.71
SpeechIO	12.86	5.30	2.93	7.56	-	3.44	2.88
Fleurs-zh	2.44	2.71	2.69	4.09	3.51	2.88	2.41
CV-zh	6.32	7.70	5.95	12.91	6.20	6.89	5.35

ROADMAP — Part 2 (Hexin)

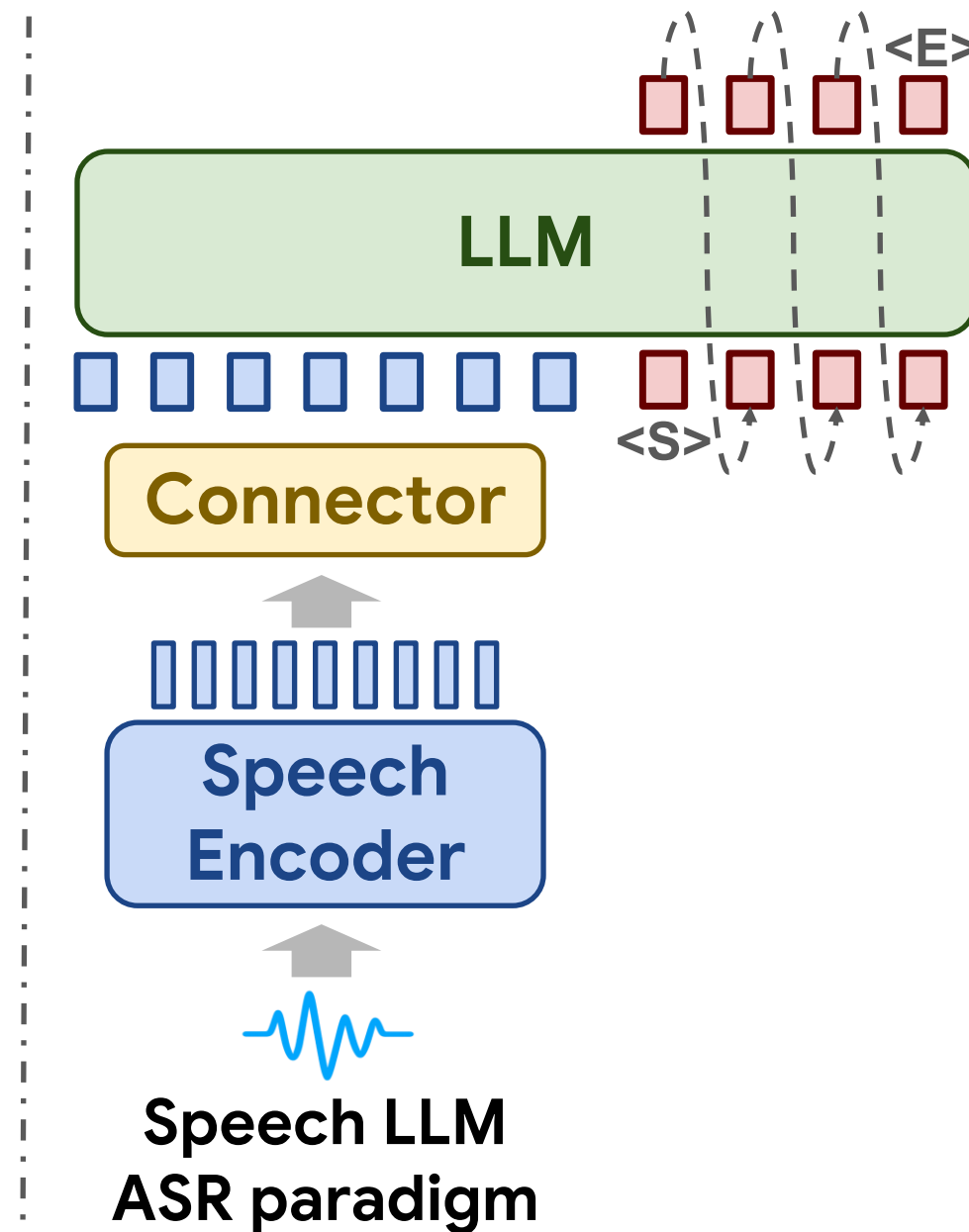
3 Speech LLM for ASR - Continuous Interface ~10 mins

4 Data-efficient ASR with Speech LLM & Continuous Interface ~15 mins

What a connector actually does:

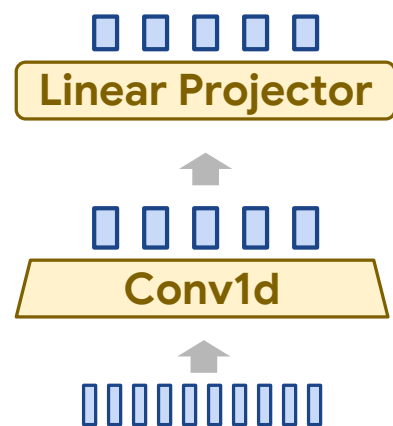
1. **Dimension mapping:**
 - Encoder hidden size → LLM embedding size
2. **Length adaptation:**
 - Long audio feature sequence → Shorter length
3. **Semantic alignment:**
 - Land in the LLMs' feature space via fine-tuning

Categories	Examples
Compression	CTC-based frame removal; Convolutional layer; Frame stacking/concatenation/splicing; Pooling
Attention-based resampling	Q-former

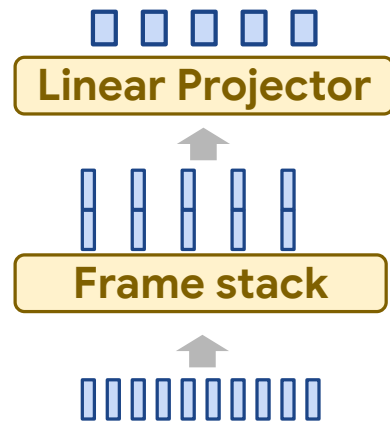


3 Speech LLM for ASR - Continuous Interface

Compression

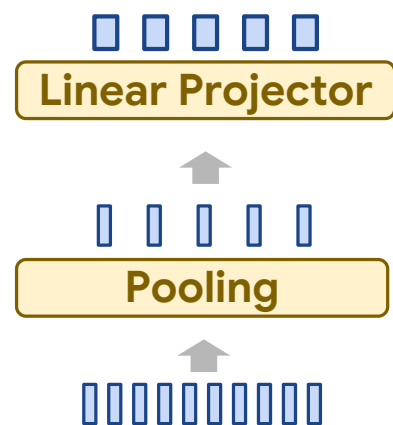


a) Conv1d

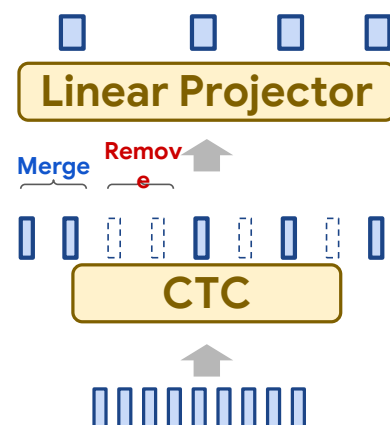


b) Frame stacking/splicing

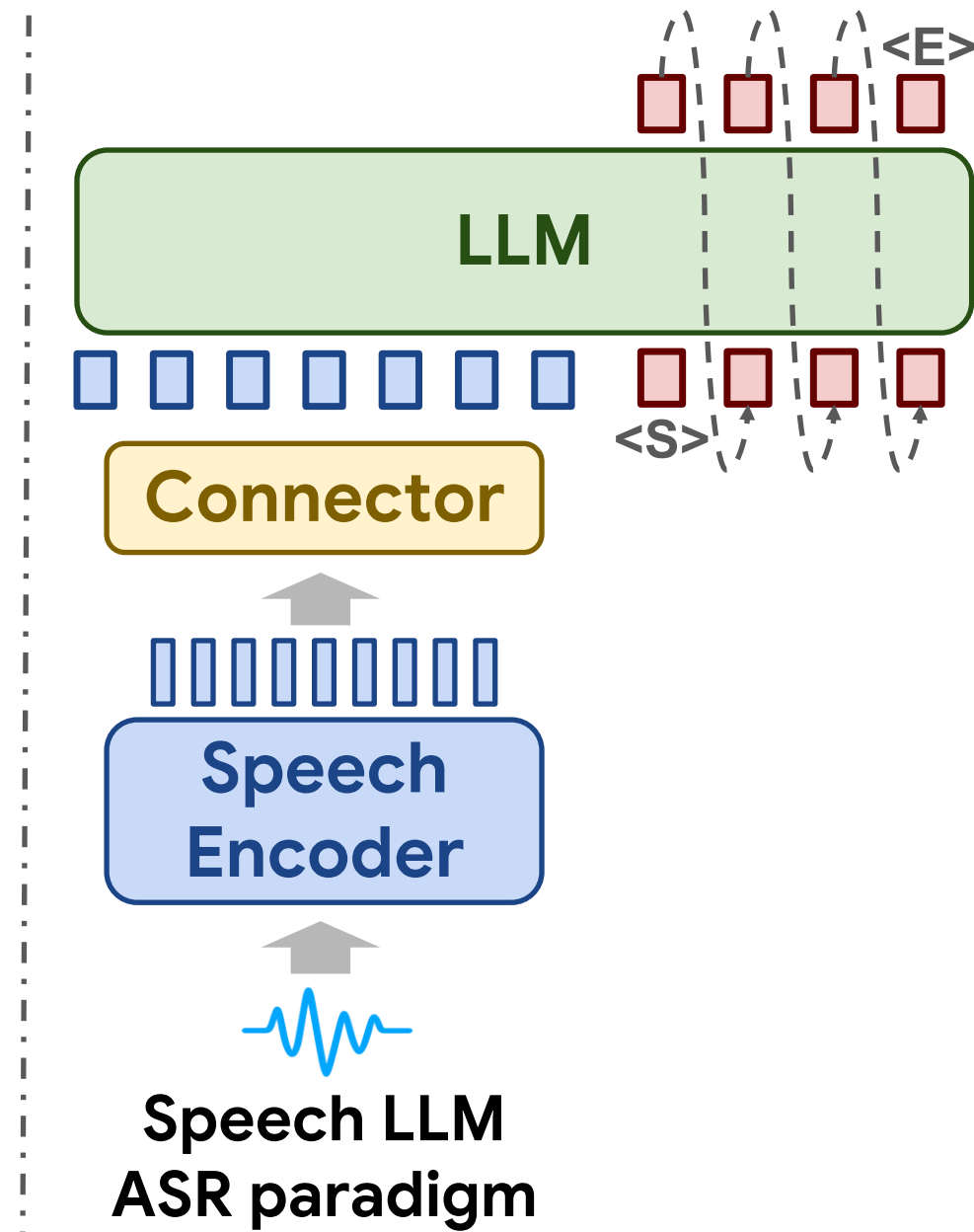
a) Conv1d is *mathematically equivalent* to b) Frame stacking + Linear



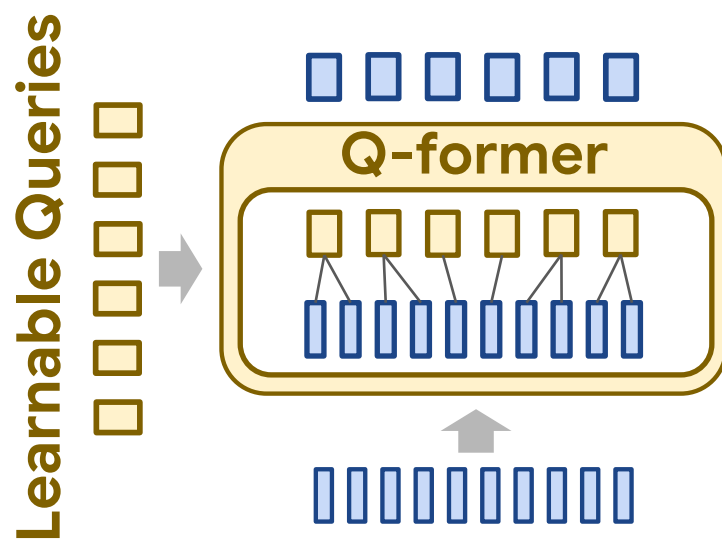
c) Pooling



d) CTC Compression

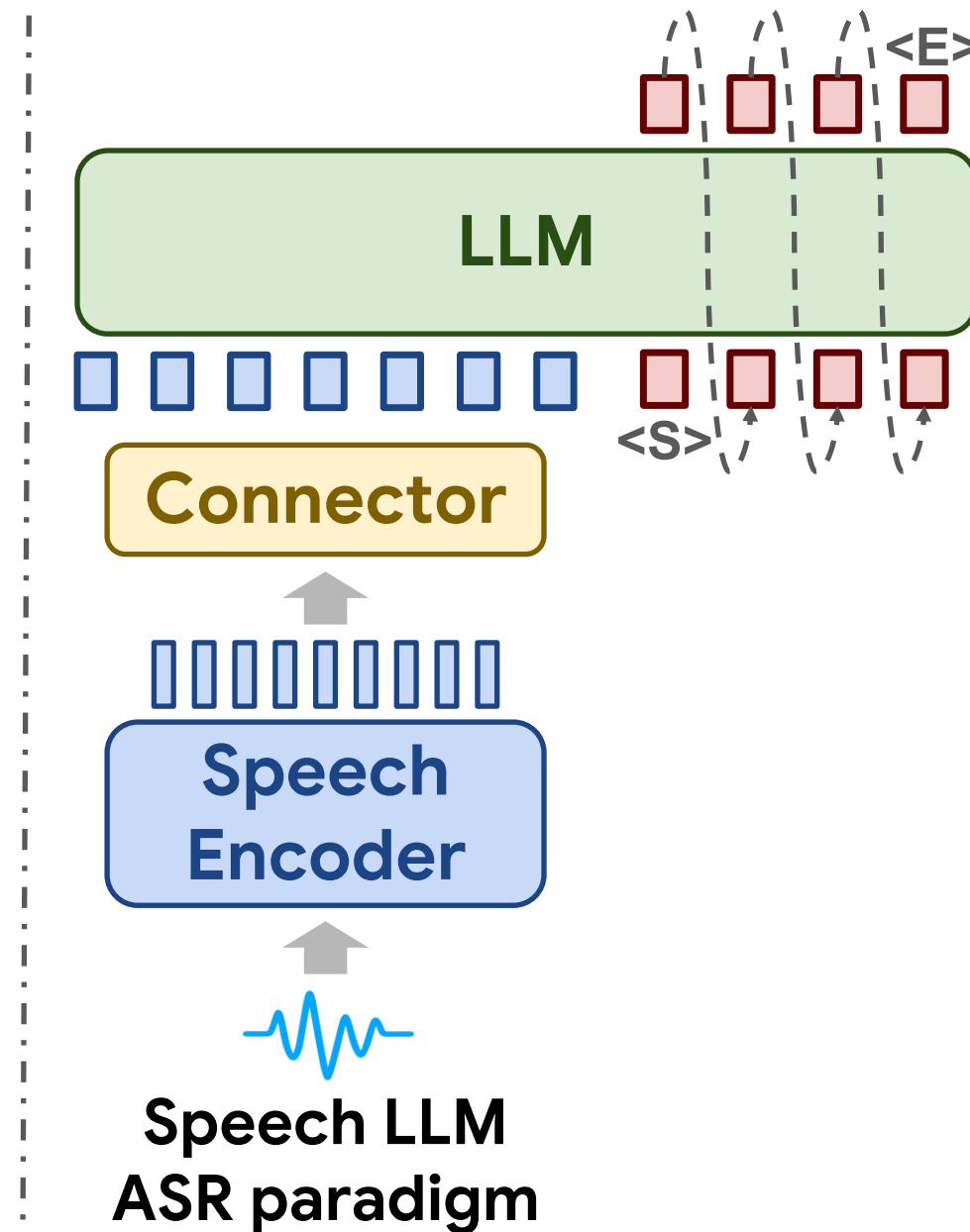


Attention-based resampling



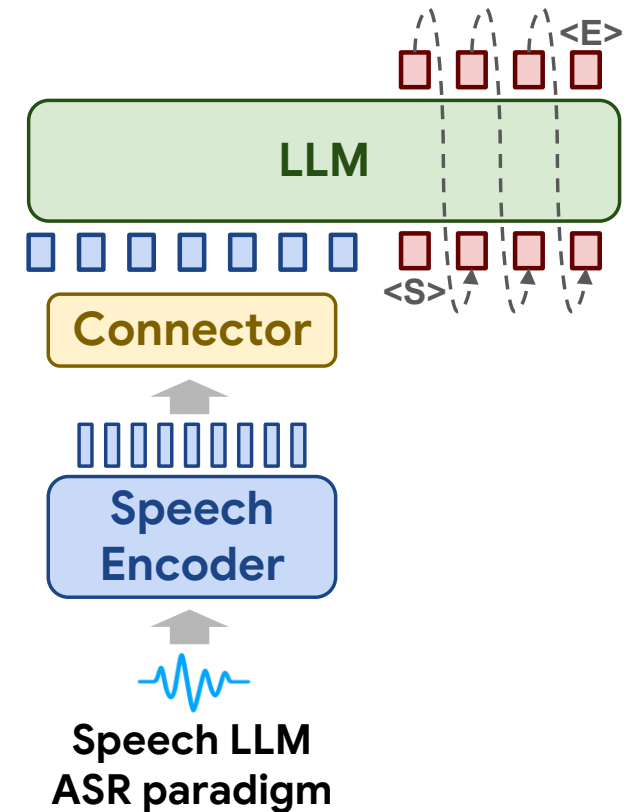
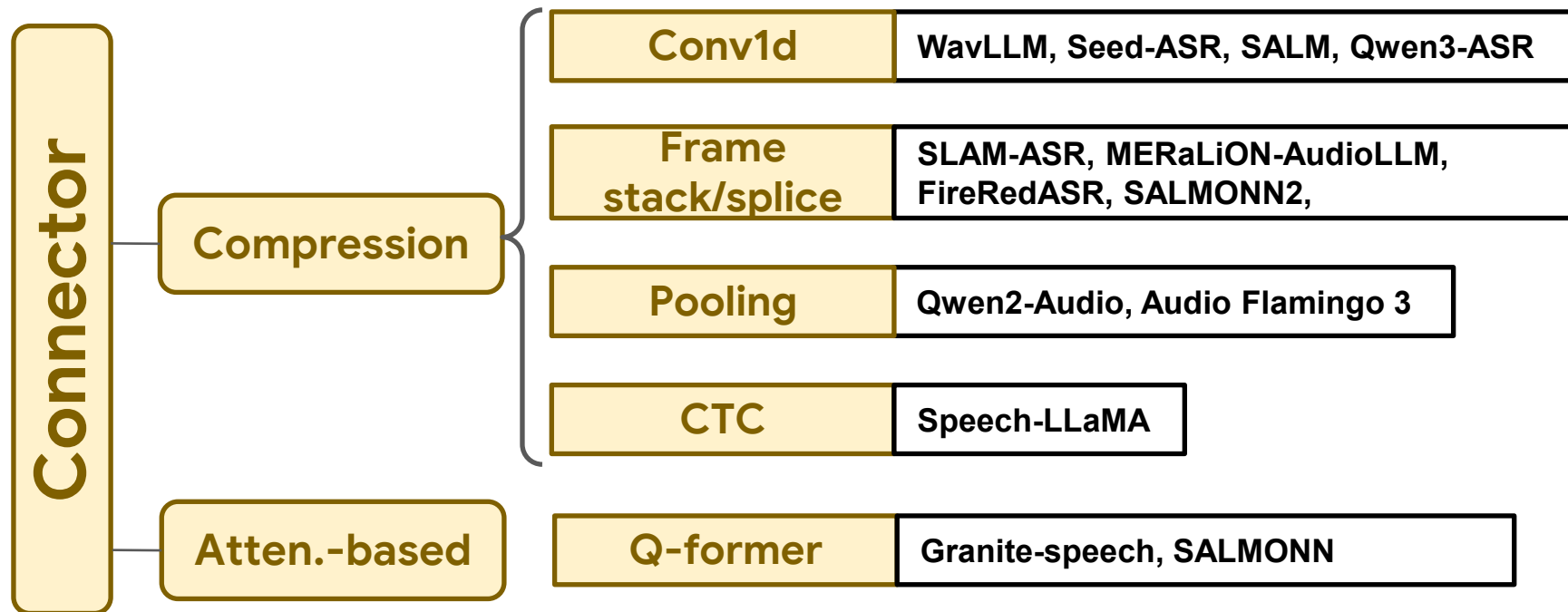
Q-former methods:

- Vanilla Q-former: **Fixed-length** output sequence
- Seg-level Q-former: **Dynamic** sampling rate
- Window-level Q-former: **Constant** sampling rate

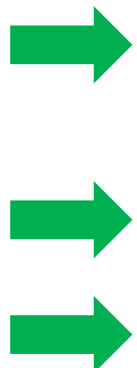


3 Speech LLM for ASR - Continuous Interface

Some examples (incomplete)



3 Speech LLM for ASR - Continuous Interface



Encoder	Adapter	Sampling Rate (Whisper 50Hz)	LLM	LibriSpeech (WER %)	
				test-clean	test-others
HuBERT-Large (ASR f.t. w/ CTC & LoRA)	Conv1d MLP	6.25 Hz	Vicuna-7b-v1.5 (LoRA f.t.)	1.78	3.62
	Downsampler MLP			1.59	3.89
	Conv1d Transformer			1.85	3.77
Whisper-Large-v2 (Frozen)	Conv1d MLP	2.5 Hz	Vicuna-13b-v1.5 (Frozen)	3.00	6.70
	Conv1d MLP	10 Hz		2.29	5.44
	Conv1d Transformer	2.5 Hz		3.22	7.54
	Q-former (60 Query)	2 Hz		2.33	5.43
	Q-former (80 Query)	3.75 Hz		2.28	5.20
	Q-former (300 Query)	10 Hz		2.19	5.25
	Seg-Q-former (80 Query)	3.75 Hz		2.35	5.37

Table. Comparison of various connectors for LLM-based ASR on English

Takeaways:

1. Q-former shows slightly **better performance** than Conv1d MLP (same speech token sampling rate) in these setups
2. Conv1d MLP is **cheap but effective**
3. A **higher** sampling rate is preferable

Encoder	Adapter	Sampling Rate	LLM	en	es	fr	it	de	avg
Whisper-Large-v3 (Frozen)	Transformer	50 Hz	Mistral-7B-Instruct-v0.3 (Frozen)	11.37	7.57	12.81	10.14	10.88	10.55
	Conv1d Transformer	12.5 Hz		11.78	7.60	13.23	10.67	11.52	10.96
	CIF Transformer	2 Hz		12.57	8.45	14.24	12.32	13.09	12.13
	CTC Transformer	3.85 Hz		14.69	10.63	17.15	15.09	16.50	14.81
	Window-level Q-former	3.12 Hz		11.82	8.21	13.60	15.77	12.55	12.39
	Transformer	50 Hz	LLaMA-3.1-8B-Instruct (Frozen)	9.90	6.33	11.27	8.52	9.09	9.02
	Conv1d Transformer	12.5 Hz		9.90	6.46	11.49	8.75	9.00	9.12
	CIF Transformer	2 Hz		12.82	8.53	14.31	12.80	13.53	12.40
	CTC Transformer	3.85 Hz		14.02	10.98	17.55	16.29	17.21	15.21
	Window-level Q-former	3.12 Hz		11.98	7.90	14.52	11.10	12.84	11.67

Table. Comparison of various connectors for LLM-based ASR on multilingual data

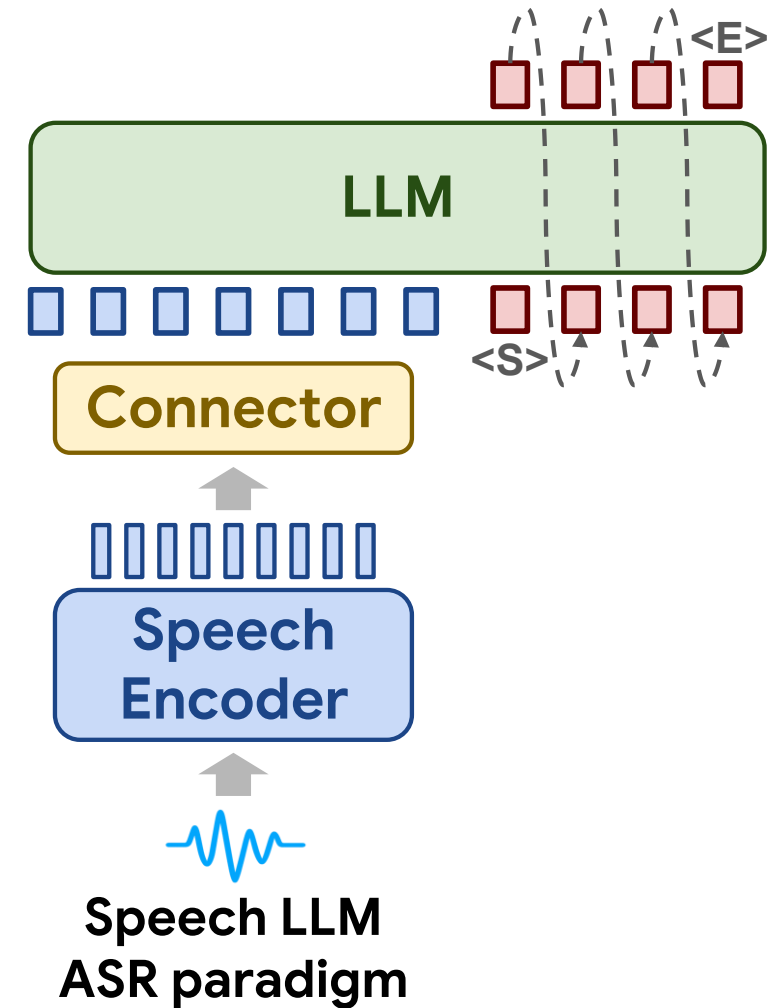
Takeaways:

1. There is no one-size-fits all solution for the adapter.
2. Conv1d MLP is **cheap but effective**

Concerns regarding Data-efficient ASR:

❑ Pre-trained Speech encoder & LLM:

- Pre-trained on high-resource languages
- Low data volume can hardly:
 - Achieve alignment for adapter
 - Fine-tune LLM and speech encoder (even with PEFT)



English performance of various training data scales:

- Training on different scales of LibriSpeech training data → 10h, 100h, and 960h
- Configs: **Speech encoder** 1) frozen, 2) LoRA f.t., 3) full f.t.
LLM 1) frozen, 2) LoRA f.t.
Connector 1) full f.t.

Encoder	Adapter	LLM	Trainable		960h		100h		10h	
			Speech Enc	LLM	test-clean	test-others	test-clean	test-others	test-clean	test-others
HuBERT Large	Conv1d MLP (6.25 Hz)	Vicuna-7b-v1.5	Frozen	Frozen	2.29	5.67	4.23	8.69	14.51	20.61
			LoRA		1.79	3.91	2.70	5.99	6.74	11.72
			Full		1.70	3.58	2.86	5.14	5.75	10.40
			Frozen	LoRA	2.02	4.31	3.45	7.19	13.30	18.75
			LoRA		1.78	3.62	2.57	5.27	7.44	11.49
			Full		1.74	3.16	2.28	4.92	4.97	9.11

Table. Comparison of tuning Speech LLM with various data scales of LibriSpeech training data

Takeaways:

1. LibriSpeech should be a very **“in-domain”** dataset for both Speech encoder and LLM.
2. Much **worse** than HuBERT Large fine-tuning [11] for ASR under 100h and 10h scenarios, clearly **better** after fine-tuning LLM

Cross-lingual and multilingual performance under low-resource scenario:

- Training on different scales of Common Voice (CV) Italian (it) training data → 10h, 50h, and 200h
- Only adapter is trained

Encoder	Adapter	LLM	200h		50h		10h		Whisper-large-v3-turbo	
			CV it	Fleurs it	CV it	Fleurs it	CV it	Fleurs it	CV it	Fleurs it
Whisper Large-v3-turbo	Downsample MLP (10 Hz)	EuroLLM 1.7B	6.4	13.2	9.2	20.6	14.0	35.2	7.1	5.8
		Salamandra 2B	7.7	65.7	12.2	56.6	33.6	111.4		

Table. Comparison of tuning Speech LLM with various data scales of Common Voice Italian training data

Takeaways:

- Fine-tuning speech LLM on low-resource data proves *insufficient*.

Cross-lingual and multilingual performance under low-resource scenario:

- First, pre-training adapter on LibriSpeech 100h, Common Voice English (en) 100h, and Common Voice Spanish (es) 100h
- Then, only adapter is trained

Encoder	Adapter	LLM	CV it	Scratch		LibriSpeech 100		Common Voice en 100		Common Voice es 200	
				CV it	Fleurs it	CV it	Fleurs it	CV it	Fleurs it	CV it	Fleurs it
Whisper Large-v3-turbo	Downsample MLP (10 Hz)	EuroLLM 1.7B	10	14.0	35.2	12.0	16.1	9.8	17.1	8.6	20.3
			15	13.9	34.0	12.8	15.7	10.0	16.9	8.8	16.1
			100	7.6	16.3	8.1	12.6	7.7	16.4	7.3	17.6
			200	6.4	13.2	6.6	11.6	6.7	15.8	6.4	17.3

Table. Comparison of tuning Speech LLM with various data scales of Common Voice Italian training data

Takeaways:

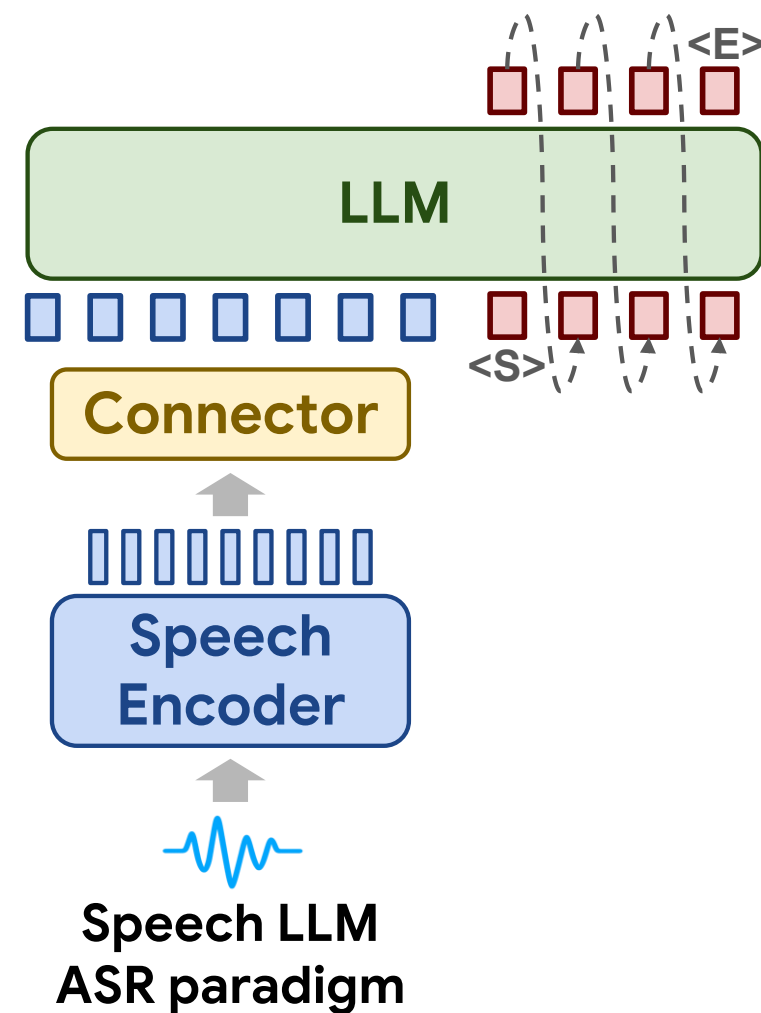
- Pre-training adapter on high-resource followed by fine-tuning unlocks **cross-lingual transfer**.
- Fine-tuning Speech LLM models (like Qwen3-ASR) on low-resource target data can be equivalent to a cross-lingual transfer (with large-scale pre-training)

4

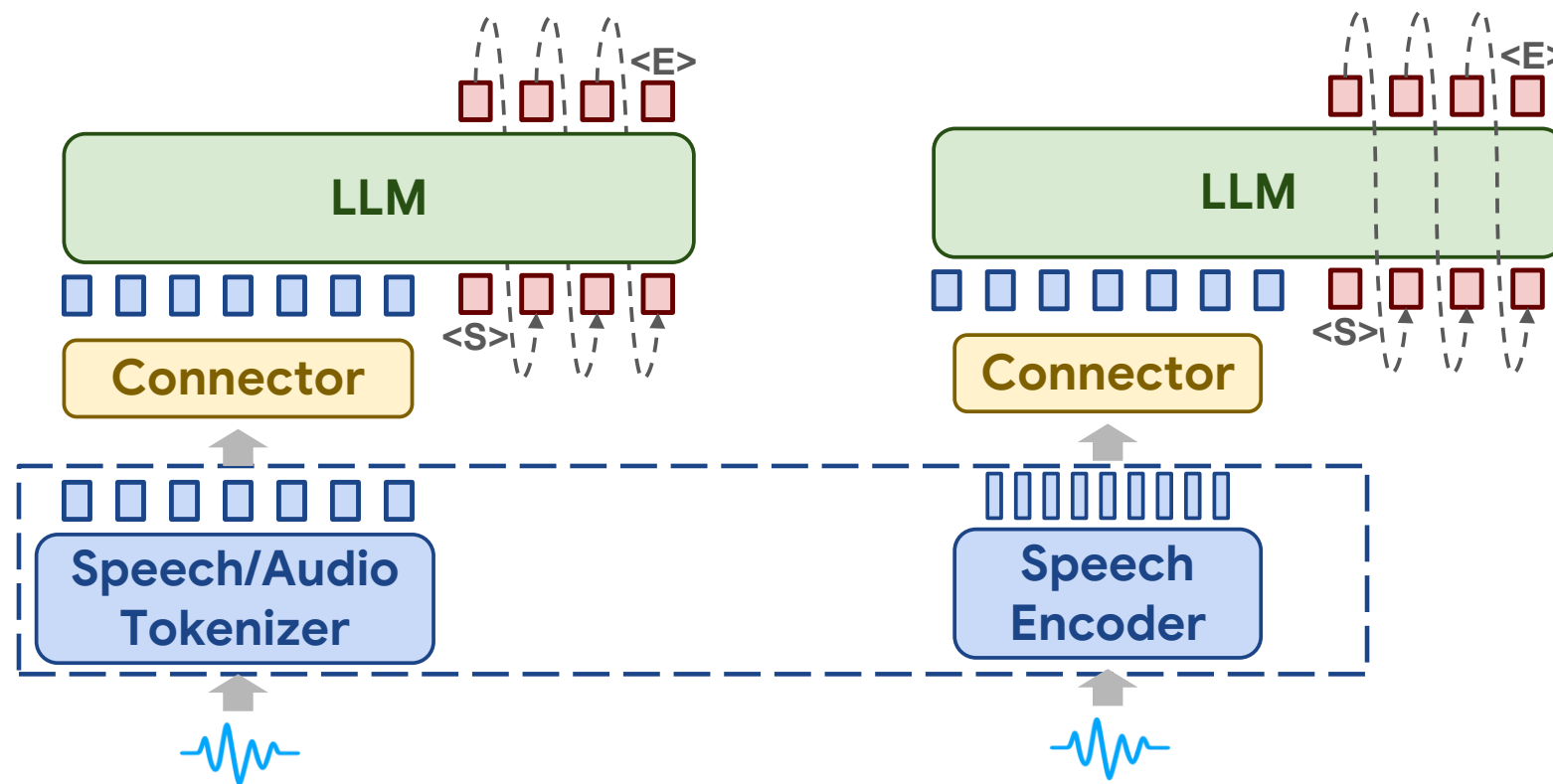
Data-efficient ASR with Speech LLM & Continuous Interface

Reaching SOTA ASR with Speech LLMs demands extensive compute and **large-scale training data**.

The same for Speech LLM with **discrete interface**.



An extension to Speech LLM w/ discrete interface



Speech Represent.:

Discrete tokens

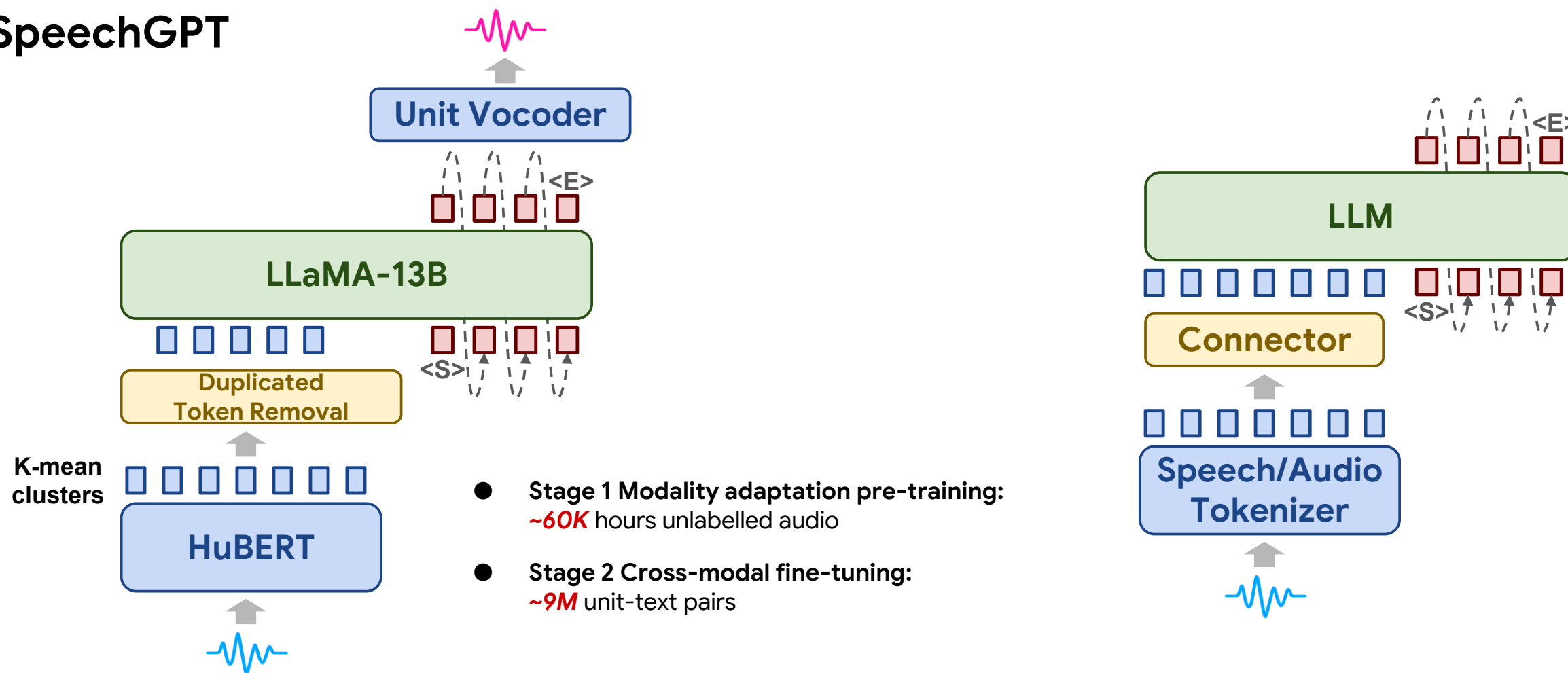
Continuous embeddings

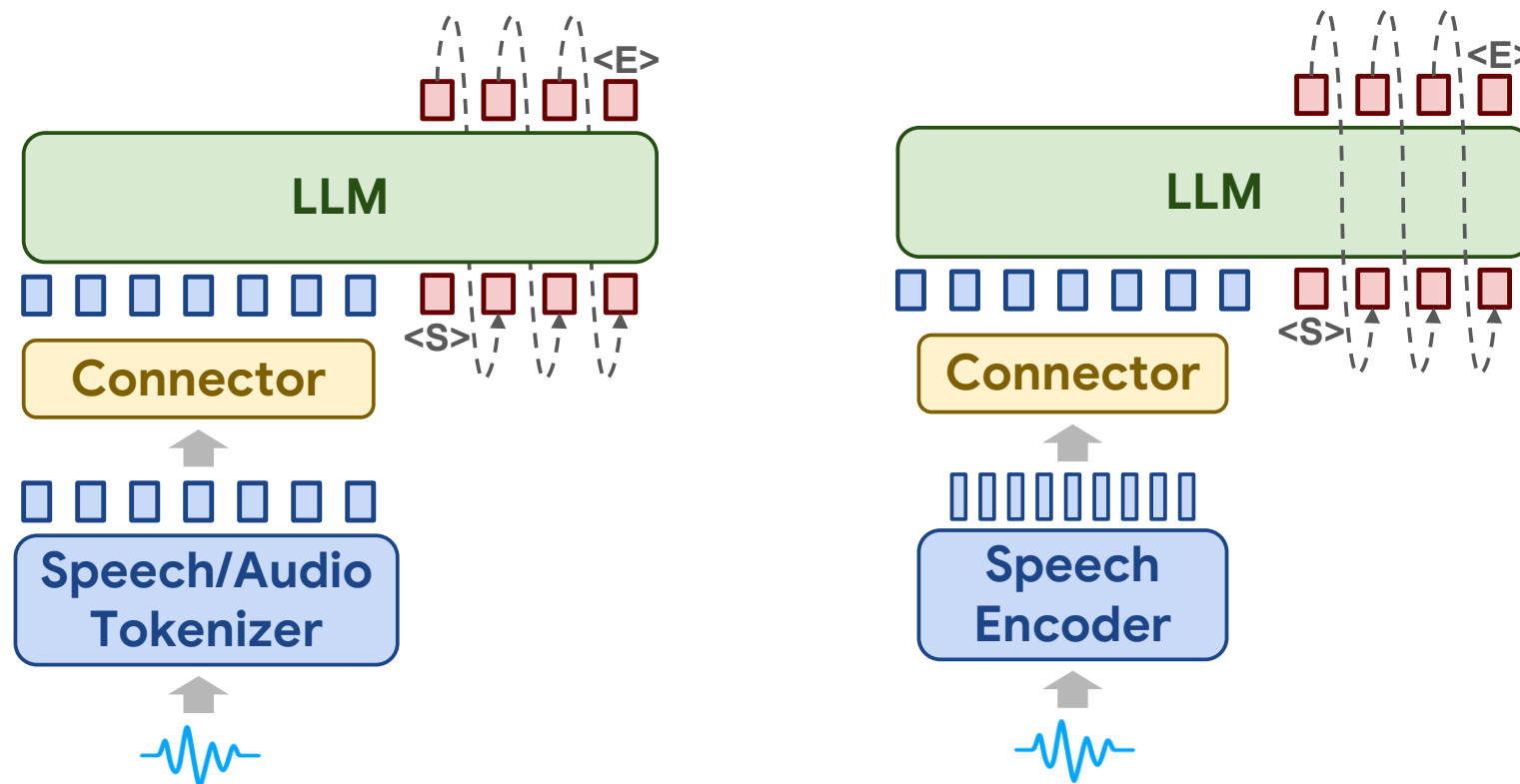
Extend LLM Vocab.:

✓

✗

SpeechGPT





Whether using continuous or discrete interfaces, Speech LLMs require **large-scale data** for modality alignment and task adaptation



Speech LLMs suitable to ASR when **sufficient resources** available

4 Data-efficient ASR with Speech LLM

Pre-training on high-resource
Fine-tuning on low-resource } → one data-efficient solution

Model	Setup	SEAME		ASRU 2019 CS
		devman	devsge	test
E-Branchformer	Expert model	16.4	23.2	11.8
Conformer	Expert model	16.6	23.3	12.8
XLS-R-300M + Conformer	SSL + Expert model	15.3	21.6	11.8
Qwen3-ASR-0.6B	Zero-shot	32.0	47.6	9.3
	Full f.t.	13.0	18.0	6.9
Qwen3-ASR-1.7B	Zero-shot	26.1	37.4	-
	Full f.t.	10.8	16.2	-

Table. Comparison of ASR performance on English-Mandarin code-switching data

Well-tuned Speech LLMs provide a stronger base for high-performing downstream fine-tuning, even for code-switching scenarios (spontaneous + accent + out-of-domain text patterns).

Content

I. Basics of Speech Recognition and LLMs (50 min)

II. Continuous-interface based approach (50 min)

III. Discrete-interface based approach (50 min)

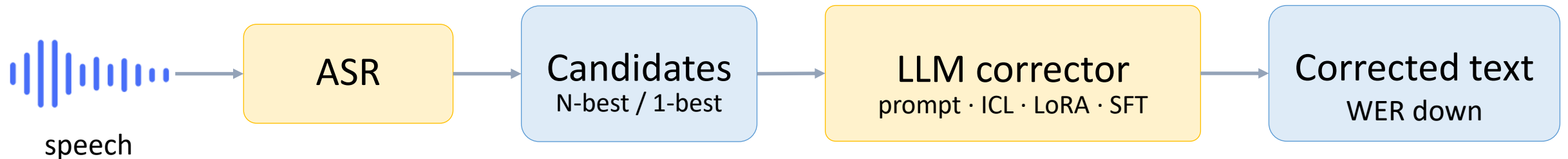
1. LLM-based ASR error correction with single-best and N-best recognition hypotheses
2. LLM-based phoneme-to-grapheme methods for phoneme-based speech recognition
3. Latent Softmax for Data-Efficient Phoneme-Based Multilingual ASR Across Tonal and Non-Tonal Languages
4. Pronunciation-lexicon free training via joint stochastic approximation

IV. Open questions and future directions (20 min)

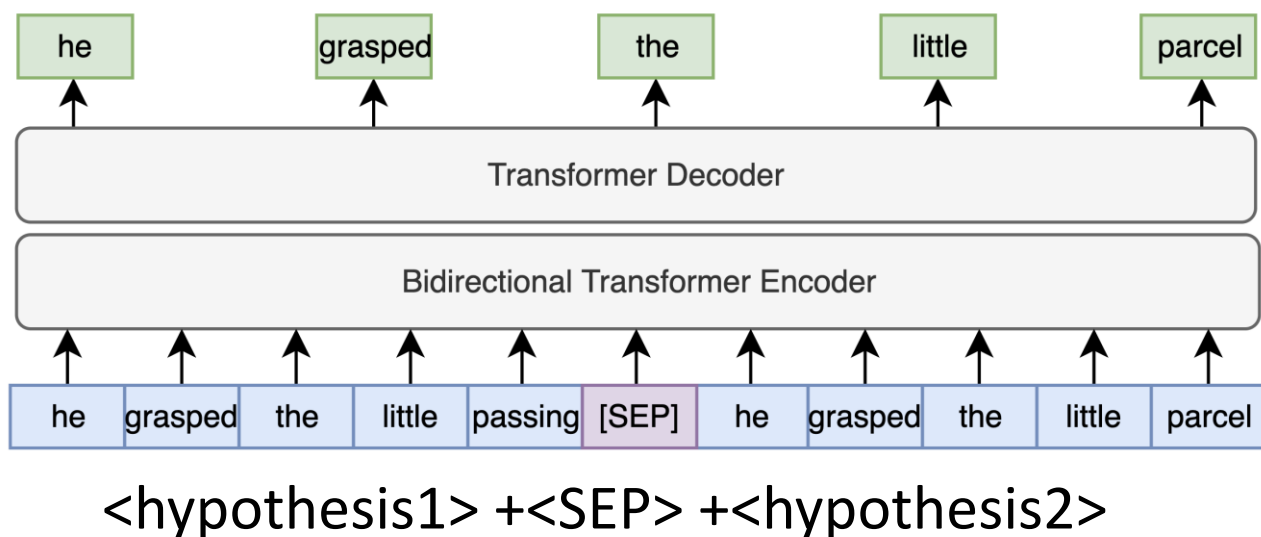
ASR Error Correction

Read the ASR output, write a better transcript — without touching the ASR itself.

- A speech recognizer produces 1-best, N-best, or a lattice.
- An LLM rewrites, selects, or rescores the hypotheses.
- The acoustic model can remain frozen or be accessed only through an API.



The LLM can be finetuned or frozen



(1) Train/Finetune an LLM

Input: Perform error correction on the top3 outputs generated by an Automatic Speech Recognition (ASR) system. The ASR hypotheses, listed in order of their ASR posterior score, are as follows:

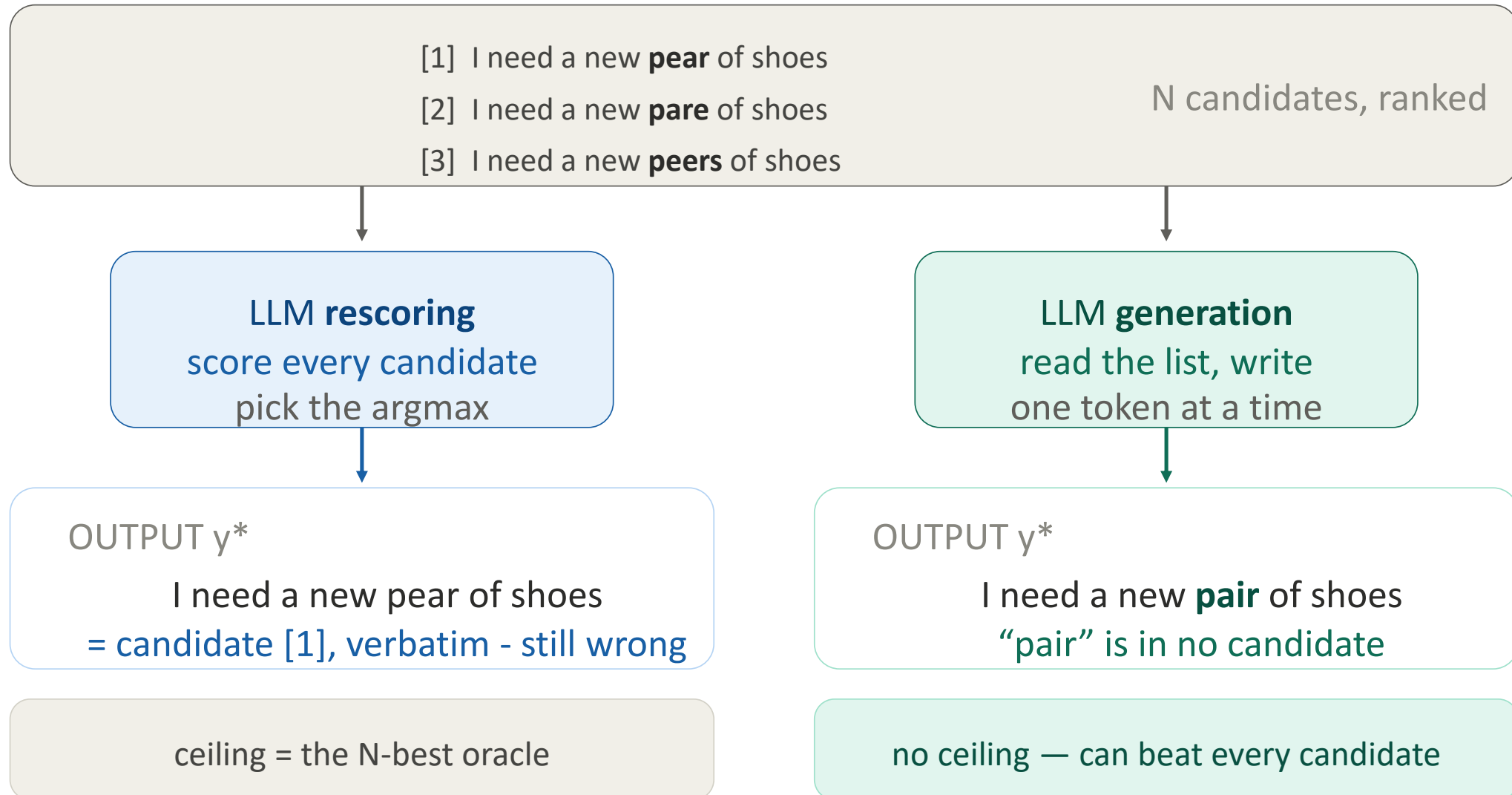
<hypothesis1> he grasped the little passing </hypothesis1>
<hypothesis2> he grasped the little parcel </hypothesis2>
<hypothesis3> he grasped the little puzzle </hypothesis3>

Please provide the corrected top1 ASR transcription of the given utterance only, do not add any explanations or other words.

ChatGPT: he grasped the little parcel

(2) Describe the task in the prompt

Two manners: rescoring or generation



Content

I. Basics of Speech Recognition and LLMs (50 min)

II. Continuous-interface based approach (50 min)

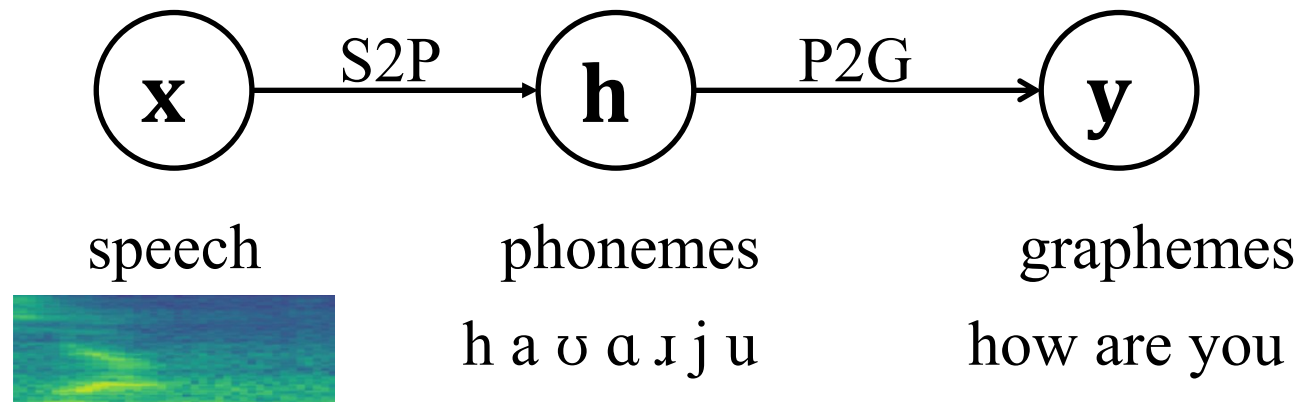
III. Discrete-interface based approach (50 min)

1. LLM-based ASR error correction with single-best and N-best recognition hypotheses
2. LLM-based phoneme-to-grapheme methods for phoneme-based speech recognition
3. Latent Softmax for Data-Efficient Phoneme-Based Multilingual ASR Across Tonal and Non-Tonal Languages
4. Pronunciation-lexicon free training via joint stochastic approximation

IV. Open questions and future directions (20 min)

Motivation & Background

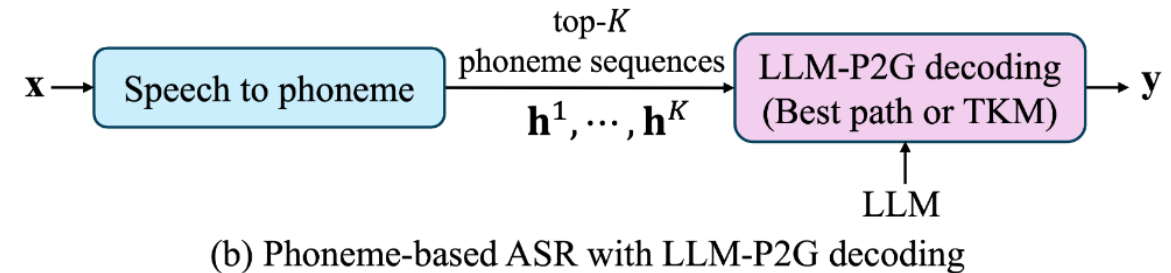
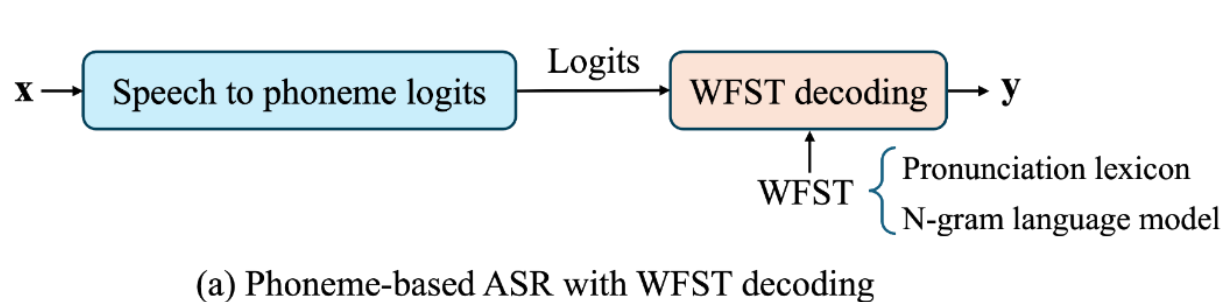
- It is found in **Whistle** ^[*] that when cross-lingual fine-tuning data is more limited, **phoneme-based supervised pre-training** achieves the most competitive results and provides high data-efficiency.
 - Presumably, this is because phoneme-based supervision enables more efficient **data sharing** than subword-based supervision.
- We speculate that **modeling phonemes as an interface between speech and text** in the ASR pipeline, serving as a **structural constraint**, significantly reduces the problem complexity.



[*] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, “Whistle:Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision,” IEEE Transactions on Audio, Speech and Language Processing, pp. 1–14, 2025.

Motivation & Background

- However, challenges remain for decoding in **phoneme-based ASR**.
- The widely used Weighted Finite State Transducer (WFST) based decoding has **two major drawbacks**:
 - It involves a complex pipeline, which needs construction of pronunciation lexicons and compiling of WFSTs;
 - It is not easy to effectively leverage the rich linguistic knowledge in large language models (LLMs).



- P2G models can be naturally trained over LLMs
- Simplifying decoding pipeline

Related Work

- **The two-step idea of phoneme-based ASR** (recognizing speech to phonemes and then to graphemes) has been studied for ASR ^[13,14]
 - with a similar motivation as ours
 - However, prior studies do not explore using LLMs for P2G for phoneme-based ASR.
- **How to integrate LLMs into ASR?**
Different interfaces between speech and languages have been studied:
 - ASR-generated text ^[16]: heavier than phonemes
 - Continuous embeddings of speech ^[17, 18, 19]

[13] “TranUSR: Phoneme-to-word transcoder based unified speech representation learning for cross-lingual speech recognition,” INTERSPEECH, 2023.

[14] “Optimizing two-pass crosslingual transfer learning: Phoneme recognition and phoneme to grapheme translation,” ASRU, 2023.

[16] “Can generative large language models perform ASR error correction?” arXiv, 2023.

[17] “On decoder-only architecture for speech-to-text and large language model integration,” ASRU 2023.

[18] “Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models,” arXiv 2023.

[19] “An embarrassingly simple approach for LLM with strong ASR capacity,” arXiv 2024.

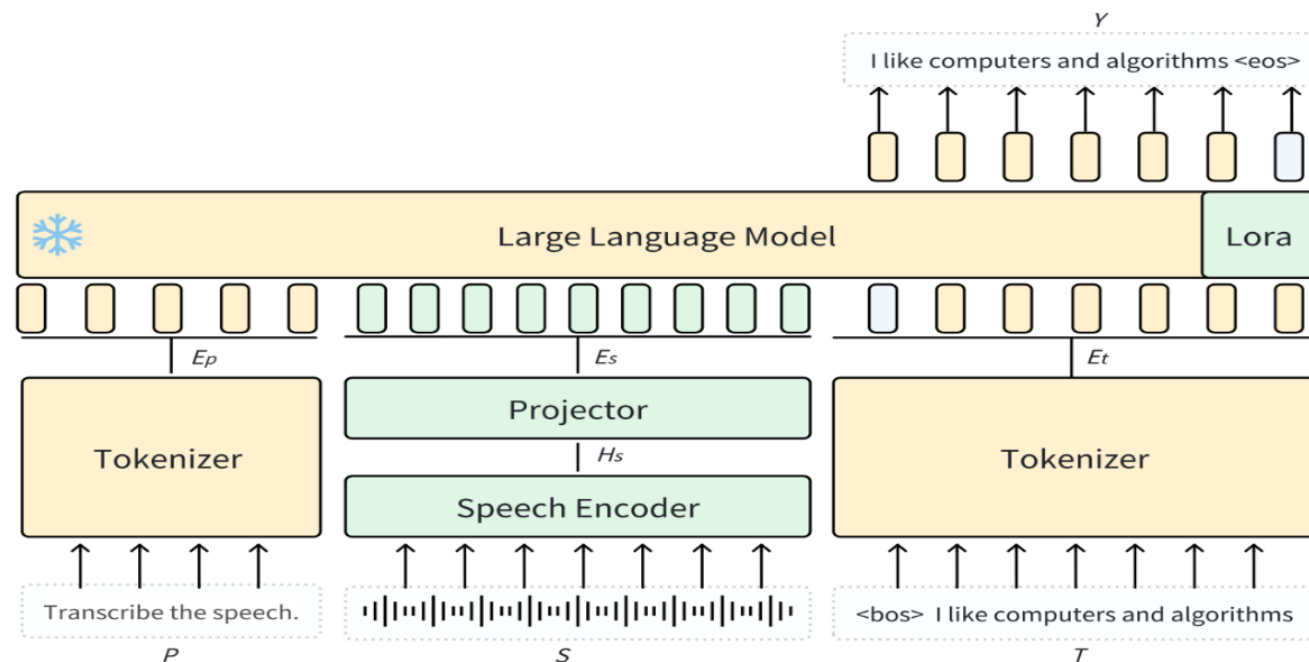
[LLM-based phoneme-to-grapheme for phoneme-based speech recognition. INTERSPEECH 2025.](#)

Related Work

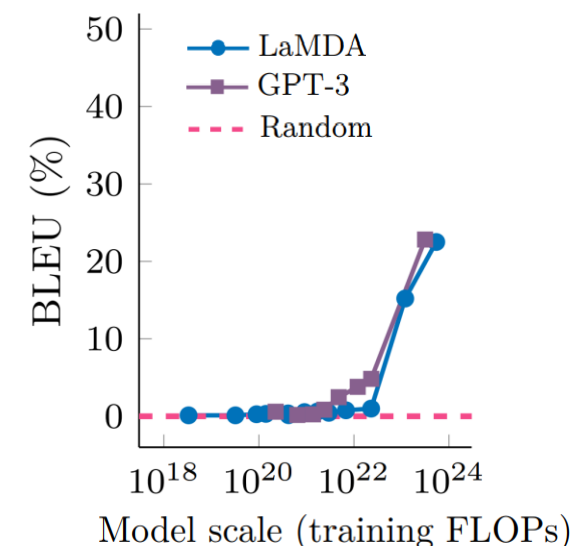
- **Speech Encoder + Projector + LLM:**
 - Black box
 - Needs additional projection, which is not easy, not natural

- **Our approach (LLM-P2G):**
 - Explainable
 - Natural integration
 - ❑ Both phonemes and subwords are discrete tokens
 - ❑ The IPA symbols all fall in the token set of LLM
 - ❑ It is found that the P2G capability (called IPA transliterate) **emerges** in LLMs^[15]

LLMs are well suited to P2G !

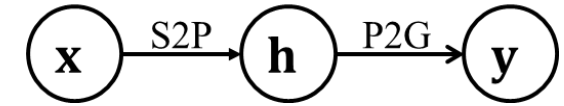


(B) IPA transliterate



[15] "Emergent abilities of large language models," Transactions on Machine Learning Research, 2022.

Method overview



A two-step ASR architecture, which we refer to as **SPG** (speech-to-phoneme-to-grapheme):

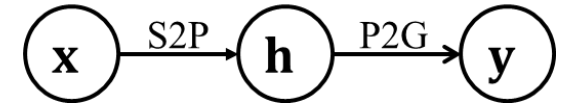
$$p(\mathbf{y} | \mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{x}) p(\mathbf{y} | \mathbf{h}) \quad (1)$$

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{h}$ denote speech, grapheme and phoneme sequence.

- **S2P (speech-to-phoneme)**
 - Modeled by $p(\mathbf{h} | \mathbf{x})$, CTC-based in our experiments
 - Obtained by fine-tuning a phoneme-based multilingual S2P backbone (Whistle) ^[7] over speech data with phoneme labels, which we refer to as *Whistle-S2P*.
- **P2G (phoneme-to-grapheme)**
 - Modeled by $p(\mathbf{y} | \mathbf{h})$
 - Obtained by fine-tuning an LLM

[7] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, “Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision,” IEEE Transactions on Audio, Speech and Language Processing, pp. 1–14, 2025.

Method overview



A challenge in building the two-step ASR

- There seems to have *information loss* in cascading S2P and P2G.
- In decoding, the hypothesized phoneme sequence output from S2P may contain errors, which may propagate to P2G.

We propose four training strategies

- Data Augmentation with Noisy Phonemes (DANP)
- Top-K Marginalized (TKM) training and decoding
- Sampling-K Marginalized (SKM) training
- Simplified Sampling-K Marginalized (S-SKM) training



Data Augmentation with Noisy Phonemes (DANP)

- **Motivation**

- If the P2G model is fine-tuned/trained using only the single annotated phoneme sequence, then there is a severe **mismatch** in training and testing for ASR.
- The input phonemes fed to P2G in ASR testing is much **noisier**.
- Therefore, a straightforward strategy to compensate for such a mismatch is to add noise to the input phonemes fed to P2G in training.

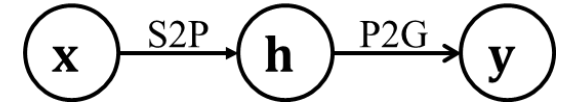
- **Training**

- For CTC-based S2P, we perform *beam search* or *sampling* to generate K hypothesized phoneme sequences
- To train P2G, the LLM is fine-tuned on K pairs of (phoneme, grapheme) sequences.

- **Decoding**

- Using the best phoneme sequence from S2P, which is fed to P2G for decoding, called *Best Path Decode*

Top-K Marginalized (TKM) training



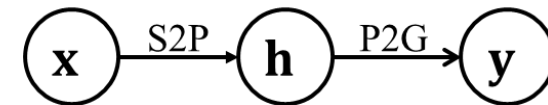
- **Motivation**

- The DANP strategy mainly address the *mismatch in training and testing* when P2G decoding is based on the 1-best phoneme sequence generated from S2P.
- Ideally, in P2G decoding, it would be better to marginalize over multiple hypothesized **h** to decode.
- The RAG-Sequence technique in Retrieval-Augmented Generation (RAG)
 - Treat the retrieved document as a latent variable that is marginalized to get *the marginal likelihood via a top-K approximation*

We maximize *the marginal likelihood $p(\mathbf{y}|\mathbf{x})$ via a top-K approximation*

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h} \in \text{top } K(p(\mathbf{h}|\mathbf{x}))} p(\mathbf{h}|\mathbf{x})p(\mathbf{y}|\mathbf{h}) = \sum_{k=1}^K p(\mathbf{h}^{(k)}|\mathbf{x}) \prod_{i=1}^L p(y_i|\mathbf{h}^{(k)}, y_{1:i-1})$$

Training Strategies



- **Data Augmentation with Noisy Phonemes (DANP)**

- Noisy S2P phoneme hypotheses (beam or random samples) to fine-tune LLM

- **Top-K Marginalized (TKM) training**

- Maximize the marginal likelihood $p(\mathbf{y}|\mathbf{x})$ via a top-K approximation^[21]

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x})p(\mathbf{y}|\mathbf{h}) \approx \sum_{\mathbf{h} \in \text{top}K(p(\mathbf{h}|\mathbf{x}))} p(\mathbf{h}|\mathbf{x})p(\mathbf{y}|\mathbf{h}) = \sum_{k=1}^K p(\mathbf{h}^{(k)}|\mathbf{x}) \prod_{i=1}^L p(y_i|\mathbf{h}^{(k)}, y_{1:i-1})$$

- **Sampling-K Marginalized (SKM) training**

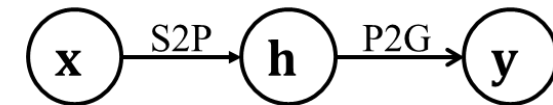
- Randomly draw CTC paths π , then collapse $\pi \rightarrow \mathbf{h}$

- **Simplified Sampling-K Marginalized (S-SKM) training**

- Simply using a Monte-Carlo estimator

$$p(\mathbf{y}|\mathbf{x}) \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{y} | \mathbf{h}^{(k)}), \mathbf{h}^{(k)} \sim p(\mathbf{h}|\mathbf{x})$$

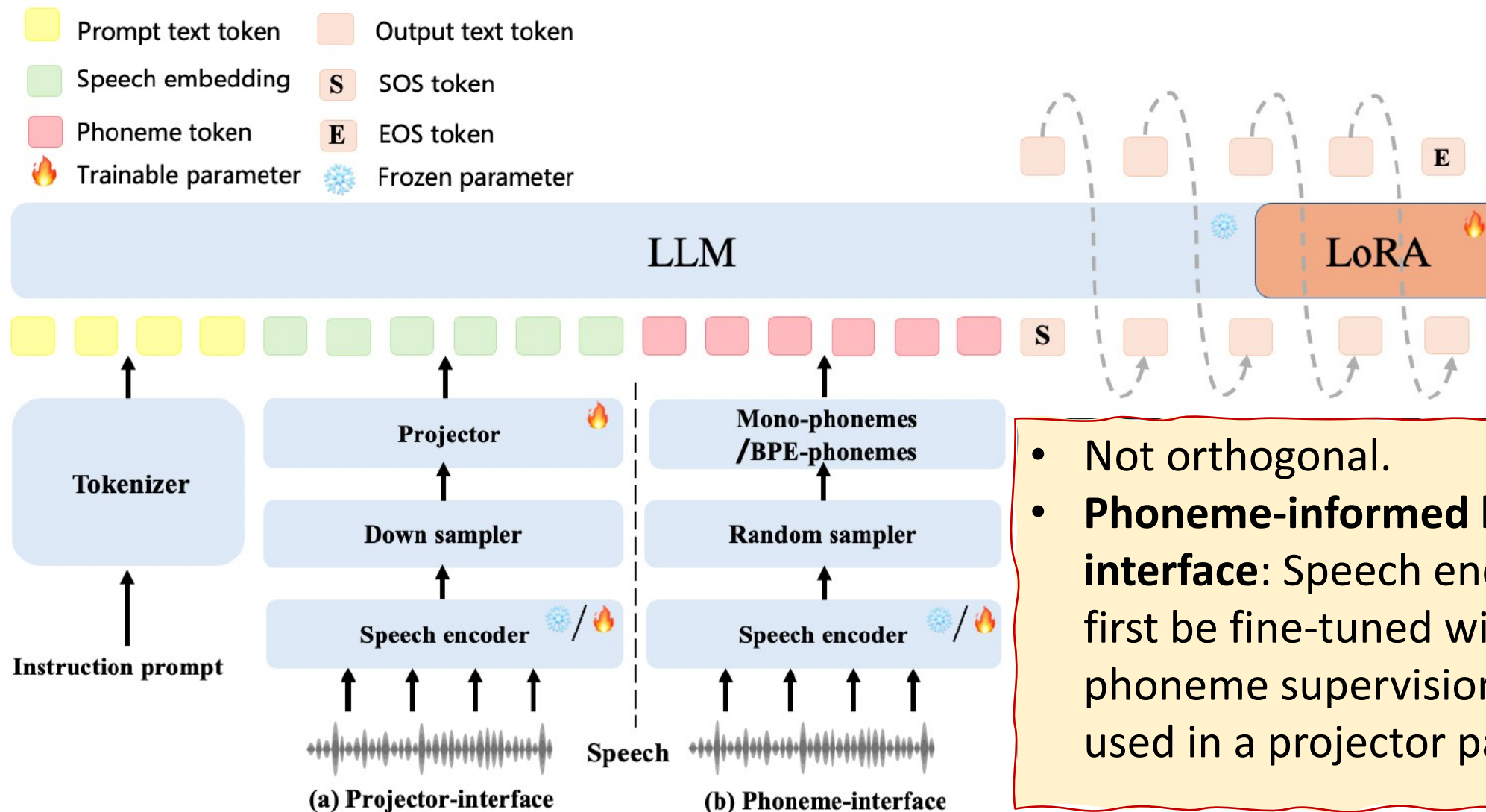
Top-K Marginalized (TKM) decoding



- We first run S2P beam search to obtain a set of top-K phoneme sequences $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(K)}$
- For each phoneme sequence candidate $\mathbf{h}^{(k)}$, the LLM-P2G generates S text predictions, producing a total of $K \times S$ grapheme sequence results.
- After de-duplication, we obtain a set of grapheme sequences $\mathbf{Y}$. Each grapheme sequence $\mathbf{y} = y_1, \dots, y_L$ from $\mathbf{Y}$ is scored as follows:
$$p(\mathbf{y}|\mathbf{x}) \approx \sum_{k=1}^K p(\mathbf{h}^{(k)}|\mathbf{x}) \prod_{i=1}^L p(y_i|\mathbf{h}^{(k)}, y_{1:i-1}) \quad (2)$$
- The top- S grapheme sequence in $\mathbf{Y}$ are obtained, which can be further re-scored, by combining Eq. (2) with in-domain language model scores.

Controlled comparison

Using the same encoder and LLM backbones



Tokenization details

Text:

stuff it into you his belly counselled him

Using Phonetisaurus based G2P;
no whitespace between adjacent mono-phonemes

Mono-phoneme:

['s', 't^h', 'ə', 'f', 'i', 't^h', 'i', 'n', 't^h', 'o', 'ʊ', 'j', 'u', 'h', 'i', 's', 'b', 'e', 'l', 'i', 'k^h',
'a', 'ʊ', 'n', 's', 'ə', 'l', 'd', 'h', 'i', 'm']

Over Librispeech, we train a SentencePiece BPE tokenizer
on phoneme sequences (keeping word boundaries).

BPE-Phoneme-100 tokenizer:

['_s', 't^h', 'ə', 'f', '_it^h', '_in', 't^h', 'oʊ', '_ju', '_hi', 's', '_b', 'e', 'li', '_k^h',
'aʊ', 'n', 's', 'əl', 'd', '_hi', 'm']

We merge BPE-phoneme units back into word-level
phoneme sequences (**phoneme-words**), and then pass
the resulting sequence to the LLM tokenizer.

Qwen3 tokenizer (150K tokens):

['st', 'h', 'ə', 'f', '_it', 'h', '_int', 'h', 'o', 'ʊ', '_ju', '_his', '_bel', 'i', '_k', 'h', 'a',
'ʊ', 'ns', 'ə', 'ld', '_him']

Training protocol

Projector-based

Pr-Stage1

- **Train:** projector only
- **Freeze:** speech encoder and LLM



Pr-Stage2

- **Freeze:** projector
- **Adapt:** LLM with **LoRA**

Phoneme-based

Ph-Stage1

- Fine-tune speech encoder with CTC
- Targets: **mono-phoneme** or **BPE-phoneme**



Ph-Stage2

- Sample **K = 8 phoneme sequences** per utterance from **frozen S2P**.
- Fine-tune LLM for **P2G** with **S-SKM** and **LoRA**.

Phoneme-informed projector: initialize the encoder from **Ph-Stage1**, then run **Pr-Stage1** and **Pr-Stage2**.

Experiment Setup

- **Speech encoders**
 - **Whistle-small (90M)**: pretrained with weakly phonetic supervision (over CV-Lang10)
 - **WavLM base (94.7M)/large (316.62M)**: pretrained with self-supervision
 - Fine-tuned with phoneme labels: Yes / No
- **LLM**
 - Qwen3-1.7B + 0.5B LoRA
 - Qwen3-8B + 1.4B LoRA
- **Dataset**
 - English (high-resource): LibriSpeech 960 hours
 - Tatar (low-resource): Commonvoice v11.0, 20 hours
- **Metric**
 - WER (Word Error Rate)

Main result: LLM-P2G based vs Projector-based

- Vanilla projector baselines (E1–E2) are weak
- Phoneme-informed hybrid interfaces (E3–E5) are substantially stronger
 - Phoneme fine-tuning makes better alignment between speech and language
- BPE-phoneme P2G gives the best result on Libri clean/other: 2.78 / 6.75 WER

ID	Speech Encoder	Encoder Params	PER(↓)		WER(↓)			
			clean	other	clean		other	
					Proj.	Phon.	Proj.	Phon.
E1	Whistle-small	90M	–	–	5.94	–	12.66	–
E2	WavLM-base	94.7M	–	–	3.08	–	7.66	–
E3	Whistle-small + Mono-phoneme-FT	90M	1.78	4.73	4.11	4.28	9.18	9.46
E4	WavLM-base + Mono-phoneme-FT	94.7M	1.10	3.21	3.03	3.37	7.02	7.56
E5	WavLM-base + BPE-phoneme-FT	94.7M	1.06	3.29	2.85	2.78	6.79	6.75

Table 1: Comparison of projector-based and phoneme-based interfaces on LibriSpeech (LLM: Qwen3-1.7B-Base + LoRA). E1–E2 are vanilla projector baselines with frozen pretrained encoders. E3–E5 use phoneme-finetuned encoders (Ph-Stage1), enabling a direct comparison between the phoneme-informed projector (Proj.) and the phoneme-based interface (Phon.). PER is reported only for phoneme-finetuned encoders; (–) indicates not applicable. For the BPE-phoneme interface we use BPE size 100, selected by the ablation in Table 4.

Result: scaling the speech encoder and the LLM

- **Scaling** improves final WER
- Reductions in PER correlate with downstream WER gains.
- Compares our best system (E8) with **representative recent LLM-ASR baselines on LibriSpeech 960h**
- Stronger than early frozen-connector systems such as Q-Former^[11] and SALM^[12]
- Competitive with SLAM-ASR^[13]

ID	Speech Encoder (BPE-phoneme-FT)	LLM	PER (↓)		WER (↓)	
			clean	other	clean	other
E5	WavLM-base	Qwen3-1.7B-Base	1.06	3.29	2.78	6.75
E6		Qwen3-8B-Base			2.43	6.00
E7	WavLM-large	Qwen3-1.7B-Base	0.70	1.71	2.13	4.13
E8		Qwen3-8B-Base			1.97	3.78

clean	other	Method	Ref.
1.97	3.78	WavLM-large + BPE-phoneme interface + Qwen3-8B LoRA	E8
2.28	5.20	Whisper-large-v2 + 80-query Q-Former connector (24.5M) + Vicuna-13B	[11]
2.30	4.80	NeMo Conformer encoder (110M) + adapter + MegatronLLM-2B LoRA + nucleus sampling	[12]
2.10	4.26	HuBERT-Large + projector (18.88M) + Vicuna-7B	[13]
2.29	5.67	HuBERT-Large + adapter + Vicuna-7B frozen (48M learnable)	[14]
1.78	3.62	HuBERT-Large LoRA + adapter + Vicuna-7B LoRA (65M learnable)	[14]
1.85	3.77	HuBERT-Large LoRA + adapter + Vicuna-7B LoRA (337M learnable)	[14]

[11] “Connecting speech encoder and large language model for ASR,” arXiv:2309.13963, 2023.

[12] “SALM: Speech-augmented language model with in-context learning for speech recognition and translation,” arXiv, 2023.

[13] “An embarrassingly simple approach for LLM with strong ASR capacity,” arXiv:2402.08846, 2024.

[14] “A comprehensive solution to connect speech encoder and large language model for ASR,” arXiv:2406.17272, 2024.

Training and inference complexity

Training complexity for the representative E5 setting on 8×A800 GPUs

The projector pipeline takes 5.14h for Pr-Stage1 (44,992 iterations) plus 4.11h for Pr-Stage2 (15,620 iterations), i.e., $5.14+4.11 = \mathbf{9.25h}$

The phoneme system with S-SKM ($K=8$) has higher per-iteration cost (2.998s vs. 0.411/0.947s), but uses only 7,000 iterations and finishes in **5.83h**

Table VIII: **Inference efficiency** on one A800 GPU.

Model	Setting	Test Set	RTF	WER
E5-Pr1	$beam_size = 4$	test-clean	0.095	3.57
		test-other	0.094	7.69
E5-Pr2	$beam_size = 4$	test-clean	0.091	2.85
		test-other	0.093	6.79
E5-Ph	$K = 8, beam_size = 4$	test-clean	0.130	2.78
		test-other	0.134	6.75
E5-Ph	$K = 1, beam_size = 4$	test-clean	0.090	2.94
		test-other	0.097	7.19

- With **K=8**, E5-Ph is slower (RTF around 0.13) but gives the best E5 WER.
- With **K=1**, E5-Ph becomes comparable to projector inference speed (0.090/0.097 RTF on test-clean/test-other, compared with 0.095/0.094 for E5-Pr1), **retaining better WER than E5-Pr1 and approaching E5-Pr2**, showing a practical speed–accuracy trade-off controlled by K.

Result: Low-Resource Evaluation on Tatar

- Using the same pretrained **frozen** Whistle-large encoder for both systems
 - The encoder is identical — the only difference between T1 and T2 is the interface.
- In this low-resource setting, the phoneme-based interface nearly halves WER

Table 7: *WER (%) on low-resource CV-tt (20h) using the same frozen pretrained Whistle-large encoder for both systems.*

ID	Interface	dev	test
T1	Projector-based	28.10	33.58
T2	Phoneme-based (Mono)	14.14	17.44

The Phoneme-based interface is more data-efficient under limited data.

Echo the conclusion as in Part I — now at the interface level in LLM-ASR

Subsection summary

3.2 LLM-based phoneme-to-grapheme generation

- **LLM-P2G** combines CTC speech-to-phoneme (S2P) recognition with LLM phoneme-to-grapheme generation.
- **Phoneme uncertainty** motivates noisy-phoneme augmentation (DANP) and marginalization over multiple hypotheses (TKM, SKM and S-SKM).
- **Phonetic supervision** also improves projector-based ASR through a phoneme-informed encoder, linking continuous and discrete interfaces.
- **Tatar 20-hour experiments** nearly halve WER with a phoneme interface using the same frozen Whistle-large encoder. This finding is specific to that setting.

Phoneme interfaces can improve data efficiency, with gains that depend on the language, data and training protocol.

Content

I. Basics of Speech Recognition and LLMs (50 min)

II. Continuous-interface based approach (50 min)

III. Discrete-interface based approach (50 min)

1. LLM-based ASR error correction with single-best and N-best recognition hypotheses
2. LLM-based phoneme-to-grapheme methods for phoneme-based speech recognition
3. Latent Softmax for Data-Efficient Phoneme-Based Multilingual ASR Across Tonal and Non-Tonal Languages
4. Pronunciation-lexicon free training via joint stochastic approximation

IV. Open questions and future directions (20 min)

Motivation & Background

1. Multilingual and cross-lingual ASR **pool** data across languages to improve training efficiency, especially in low-resource settings.
2. It is found in Whistle that **phonetic supervision** can be more data-efficient than subword supervision for speech-encoder pretraining.
3. **Phoneme-informed interface** improves over vanilla projector in LLM-ASR.

RQ: How should tones be incorporated in multilingual phoneme-based models?

- Treating a tone-marked vowel and its non-tonal base-vowel as unrelated units: weakening phonetic sharing ☹️
- Collapsing tone-marked vowel variants into the same base vowel: losing distinctions required by tonal languages ☹️

/a/

English base vowel

/ā/

/á/

/ǎ/

/à/

Mandarin tone-marked vowels



Latent Softmax

- **Challenge: supervision mismatch**

Tonal language requires a finer tone-marked-vowel subclass space

VS

Non-tonal vowel supervision provides only base-vowel major-class labels

- Non-tonal speech still exhibits pitch variation
 - This variation is linguistically non-contrastive and **unmarked** in phonetic transcription
 - But is **helpful** for speech discrimination [5]–[7]

[5] “Models of tone for tonal and non-tonal languages,” ASRU, 2013.

[6] “Switching auxiliary chains for speech recognition,” IEEE Signal Processing Letters, 2007.

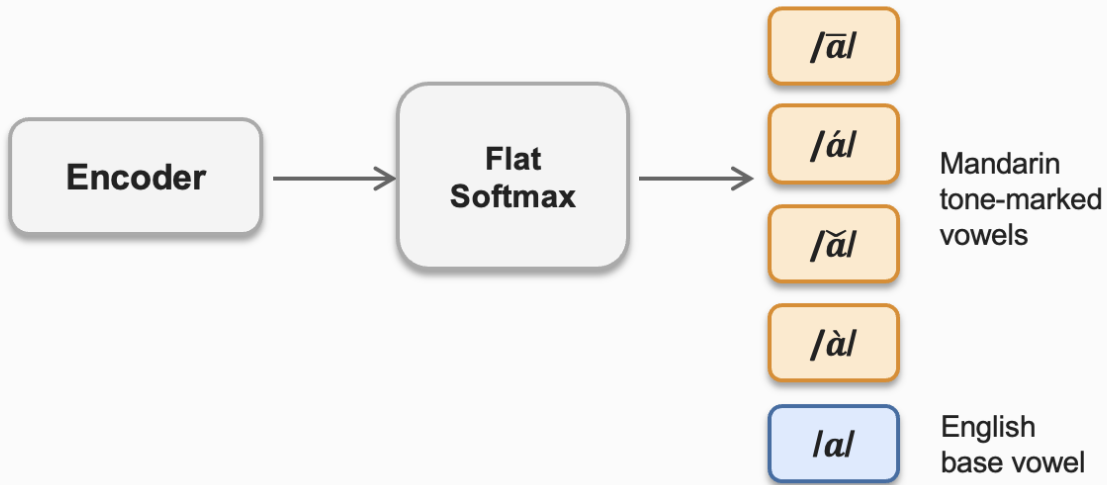
[7] “Switching auxiliary chains for speech recognition based on dynamic bayesian networks,” ICPR, 2006.

Latent Softmax

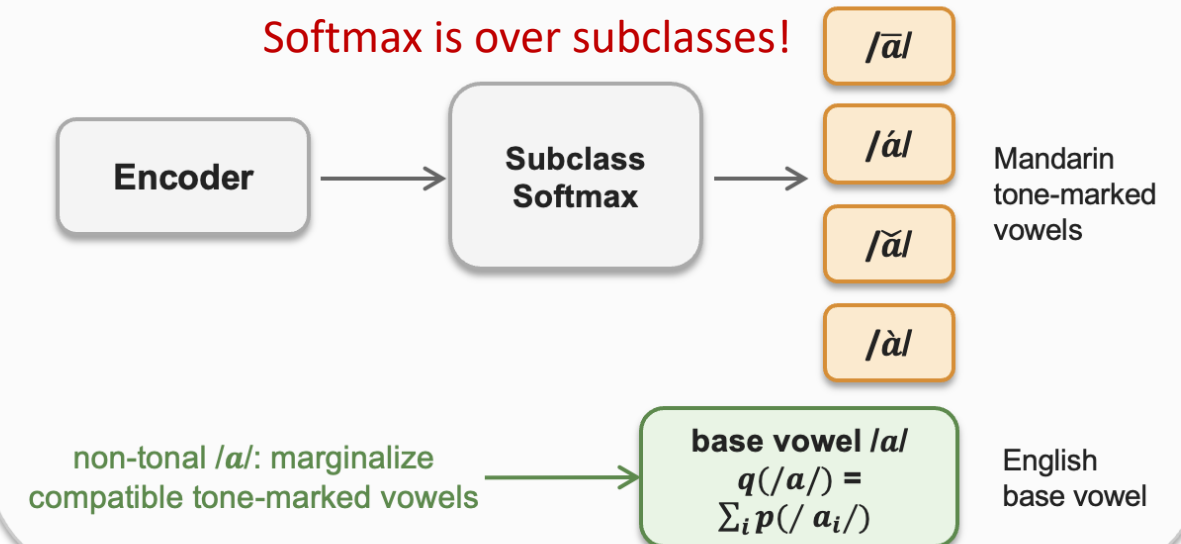
This motivates a latent view:

When only a base-vowel label (major class) is observed, the tone-marked subclass is latent and marginalized out.

Standard Softmax



Latent Softmax



Empirical label distribution on the mixed English–Mandarin data

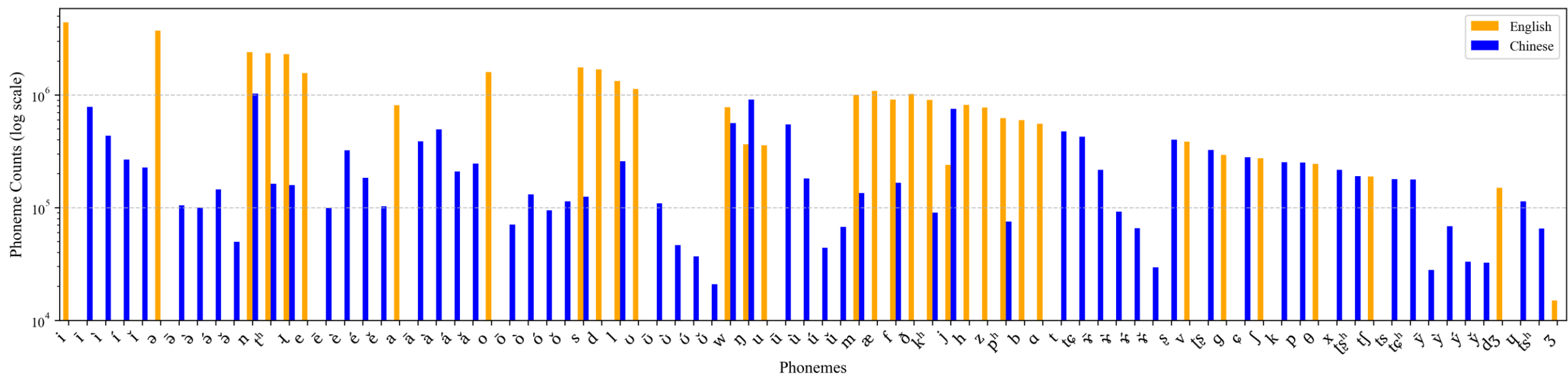


Fig. 2: Phoneme-count distribution in LibriSpeech English and AISHELL-1 Mandarin.

- The horizontal axis lists English and Mandarin phoneme labels, **ordered by descending major-class count**, and the vertical axis reports their corpus-level counts in the two datasets.
- When a phoneme appears in both English and Chinese, **two bars** are shown side-by-side.
- Base vowels and their tone-marked vowel subclasses, such as /a/, /ā/, /á/, /ǎ/ and /à/, are placed contiguously to show how non-tonal and tonal evidence is distributed across the shared inventory.

Standard softmax for CTC-based S2P

- A standard softmax layer produces logits $u_t(l)$ for each $l \in \mathcal{L}$
- Frame posterior:

$$p_t(l) = p(\pi_t = l \mid \mathbf{x}) = \frac{\exp(u_t(l))}{\sum_{l' \in \mathcal{L}} \exp(u_t(l'))}$$

The label inventory —
— **flat** (Tone-marked
vowels and base vowels
are **unrelated**)

- For target label sequence $\mathbf{y}$, the CTC likelihood is

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\pi \in \mathcal{A}(\mathbf{y})} \prod_{t=1}^T p_t(\pi_t)$$

$\mathcal{A}(\mathbf{y})$ is the set of frame-level alignments (paths) that collapse to $\mathbf{y}$

Latent Softmax for CTC-based S2P

- For $l \in \mathcal{L}$, define the compatible-subclass set:

$$B(l) = \{z \in \mathcal{Z} : z \text{ is compatible with } l\}$$

$\mathcal{Z}$ denote the subclass inventory over which softmax is calculated.

Example:

$$B(/a/) = \{/ā/, /á/, /ǎ/, /à/\}$$

$$B(/ā/) = \{/ā/\}$$

- Subclass = Tone-marked vowels + singleton-labels that require no subdivision including consonants and <blk>
- Each base vowel defines a major class.

Latent Softmax for CTC-based S2P

- For $l \in \mathcal{L}$, define the compatible-subclass set:

$$B(l) = \{z \in \mathcal{Z} : z \text{ is compatible with } l\}$$

$\mathcal{Z}$ denote the subclass inventory over which softmax is calculated.

- The network outputs subclass logits $v_t(z)$ and the **subclass posterior** is

$$p_t(z) = p(z_t = z \mid \mathbf{x}) = \frac{\exp(v_t(z))}{\sum_{z' \in \mathcal{Z}} \exp(v_t(z'))}.$$

- Frame posterior** for supervised label l :

$$q_t(l) = P(z_t \in B(l) \mid \mathbf{x}) = \sum_{z \in B(l)} p_t(z)$$

It is shown that the **Latent-Softmax CTC likelihood**:

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\pi \in \mathcal{A}(\mathbf{y})} \prod_{t=1}^T q_t(\pi_t) \quad \leftarrow \text{play the same computational role}$$

Example:

$$B(/a/) = \{/\bar{a}/, /á/, /ǎ/, /à/\}$$

$$B(/ā/) = \{/\bar{a}/\}$$

Standard-softmax frame posterior:

$$p_t(l) = \frac{\exp(u_t(l))}{\sum_{l' \in \mathcal{L}} \exp(u_t(l'))}$$

Standard-softmax CTC likelihood:

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\pi \in \mathcal{A}(\mathbf{y})} \prod_{t=1}^T p_t(\pi_t)$$

Experiment Setup

- **Model:**
 - **S2P model:** Conformer based acoustic encoder (115 M)
 - **LLM:** Qwen3-1.7B (LoRA $r=1024, \alpha=2048$). **Projector:** 2×4096 -dim linear
- **Dataset:**
 - AISHELL-1, 178 hours of Mandarin data with $3 \times$ speed perturbation.
 - LibriSpeech, 960 hours English data
 - ASRU2019 Mandarin-English code-switching challenge data, 200 hours
 - CS-Dialog, 104.02 hours Mandarin-English code-switching data

} **Pooled for bilingual experiment**
- **Metric:**
 - T-PER (Tonal Phone Error Rate), CER (Character Error Rate), WER (Word Error Rate)
 - NT-PER (non-tonal PER): compute PER after removing tone marks
 - MER (mixed error rate): for Mandarin–English code-switching evaluation

Mandarin-English Bilingual experiment

- Latent Softmax significantly reduces encoder **phoneme error rates** over standard softmax.
- Downstream speech recognition is evaluated with two speech-language interfaces:
 - Latent Softmax **consistently improves** both interfaces
 - Gains are largest for LLM-P2G, whose input directly depends on decoded S2P phonemes.

TABLE I
TONAL PERs (%), NON-TONAL PERs (%), CERS(%) AND WERS (%) ON AISHELL-1 AND LIBRISPEECH. AISHELL-1 USES TONAL PER, WHILE LIBRISPEECH USES NON-TONAL PER. PER IS MEASURED AT THE S2P OUTPUT.

System	AISHELL-1		LibriSpeech			
	Tonal PER	CER	test-clean Non-tonal PER	WER	test-other Non-tonal PER	WER
Standard Softmax S2P + LLM-P2G	2.03	5.63	1.60	4.29	5.00	10.70
Standard Softmax S2P + Projector		5.61		3.92		9.61
Latent Softmax S2P + LLM-P2G	1.86	5.26	1.32	3.91	4.37	9.89
Latent Softmax S2P + Projector		5.52		3.63		9.14

Mandarin-English Code-switching experiment

- Code-switching gains are interface-dependent
 - Projector-based interface: Latent Softmax improves the MER on both evaluated datasets
 - The LLM-P2G results do not establish a consistent advantage

TABLE II
NON-TONAL (NT) PERS (%) AND MERs (%) ON MANDARIN-ENGLISH
CODE-SWITCHING DATASETS AFTER CODE-SWITCHING ADAPTATION.

System	ASRU2019		CS-Dialogue	
	NT- PER	MER	NT- PER	MER
Standard Softmax S2P + LLM-P2G	3.23	13.69	9.66	-
Standard Softmax S2P + Projector		14.08		15.53
Latent Softmax S2P + LLM-P2G	3.83	13.83	9.02	14.06
Latent Softmax S2P + Projector		13.71		14.06

On CS-Dialogue, LLM-P2G is harder to adapt, because the data are limited and the decoded S2P phoneme sequences remain too noisy for training a robust text conversion model.

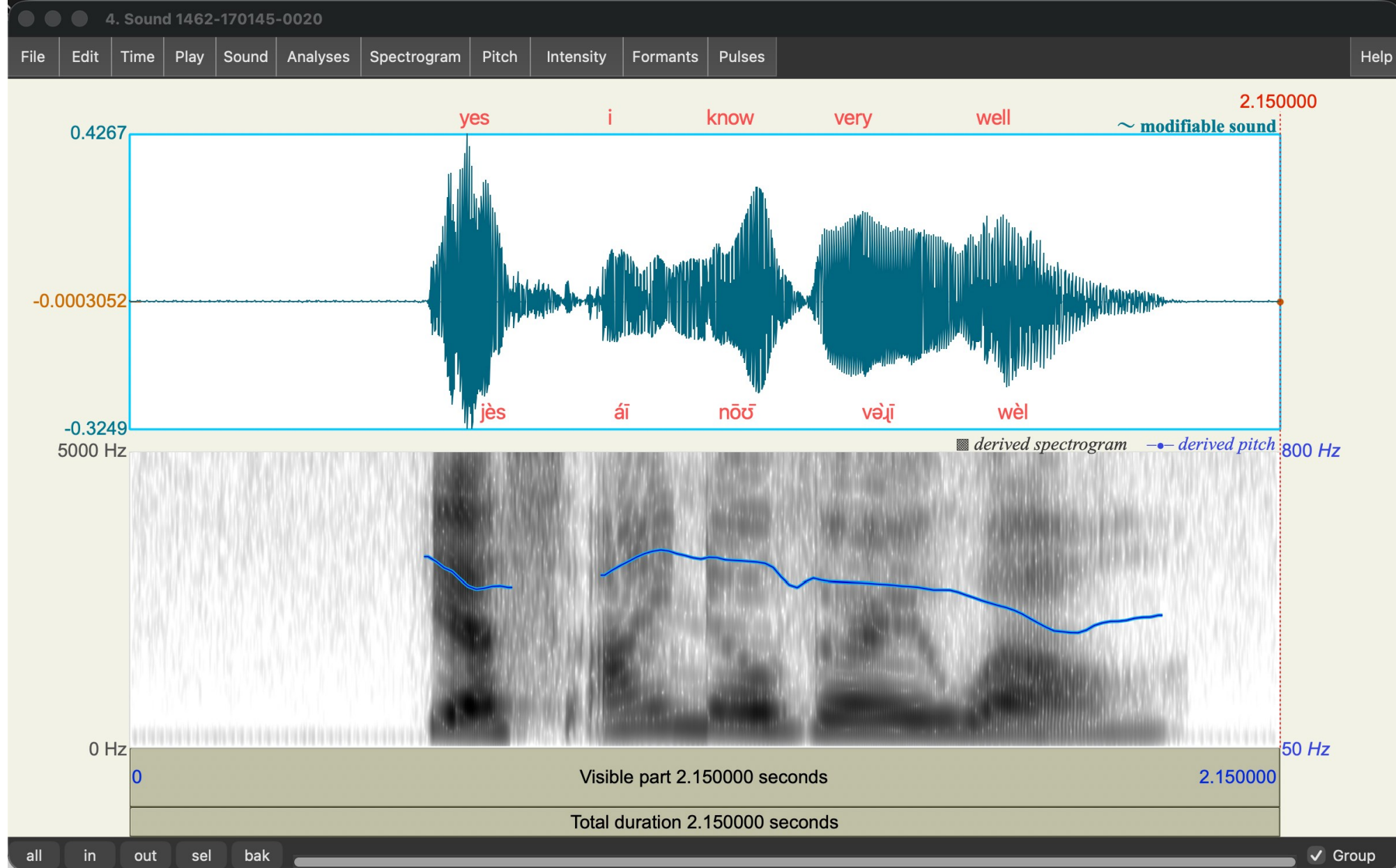
Tonal representation analysis in non-tonal languages

- Table III shows some English examples decoded by the Latent Softmax S2P model.

TABLE III
EXAMPLE SUBCLASS AND MAJOR-CLASS PHONEME DECODING ON LIBRISPEECH. SUBCLASS SEQUENCES ARE DECODED FROM SUBCLASS POSTERIORIS; MAJOR-CLASS SEQUENCES ARE DECODED AFTER MARGINALIZING SUBCLASS POSTERIOR PROBABILITIES.

text	subclass phoneme sequence	major class phoneme sequence
where is that	w è ɹ í z ð æ t ^h	w e ɹ i z ð æ t ^h
yes i know very well	j è s á í n ó ú v ə ɹ í w è l	j e s a i n o u v ə ɹ i w e l
what made the difference	w ə t ^h m è ɪ d ð ə d ì f ə ɹ ə n s	w ə t ^h m e i d ð ə d i f ə ɹ ə n s

- The examples show that, even without explicit English tone labels, **the model assigns tone-like subclass marks to English vowels**.
- We treat this observation as qualitative evidence that the latent tone-marked vowel subclasses **capture prosodic variation** useful for sharing with tonal-language data.
- It should not be interpreted as a claim that the predicted English tone marks are linguistically ground-truth tones.
- From the model’s viewpoint, an English vowel token can nevertheless **instantiate one latent tone-marked vowel subclass** inside the corresponding major class.



A preliminary check of the Table III utterances suggests **qualitative agreement between predicted subclass tone marks and F0-contour directions**. This is qualitative evidence only, not corpus-level validation.

Subclass versus major-class phoneme input in LLM-P2G

- The S2P output could be subclass sequences or major-class sequences,
- A further question: what's the effect of sending different sequences to LLM-P2G?



TABLE IV
CERs (%) ON AISHELL-1 AND WERS (%) ON LIBRISPEECH (LS) FOR
THE LLM-P2G MODEL USING SUBCLASS AND MAJOR-CLASS S2P
PHONEME SEQUENCES AS INPUT.

S2P output	AISHELL	LS clean	LS other
Major class	5.31	4.10	10.19
Subclass	5.26	3.91	9.89

- Subclass inputs yield **lower P2G WER** on all three evaluation sets.
- This suggests that the **subclass representation** contains useful fine-grained information for the P2G model, including in the non-tonal English setting.

Subsection summary

3.3 Latent-Softmax for tonal and non-tonal languages

- We studied data-efficient combined training of tonal and non-tonal languages for phoneme-based multilingual ASR.
- The proposed Latent Softmax lets non-tonal vowel data update a tone-marked vowel subclass space, yielding a more data-efficient sharing mechanism than standard softmax.
- **Mandarin-English multilingual Results:** improves S2P PER over a standard softmax and transfers these gains to both LLM-P2G and projector-based ASR interfaces.
- **Mandarin-English code-switching Results:** improves projector-based MER, the LLM-P2G results remain inconclusive.

Content

I. Basics of Speech Recognition and LLMs (50 min)

II. Continuous-interface based approach (50 min)

III. Discrete-interface based approach (50 min)

1. LLM-based ASR error correction with single-best and N-best recognition hypotheses
2. LLM-based phoneme-to-grapheme methods for phoneme-based speech recognition
3. Latent Softmax for Data-Efficient Phoneme-Based Multilingual ASR Across Tonal and Non-Tonal Languages
4. Pronunciation-lexicon free training via joint stochastic approximation

IV. Open questions and future directions (20 min)

Introduction

- Pre-trained models with phonetic supervision have demonstrated their advantages for crosslingual speech recognition in data efficiency and information sharing across languages.
- **Challenge:** phoneme labels are needed for each training utterance.
- Phoneme labels are usually obtained by using a manually-crafted pronunciation lexicon, which maps every word in the vocabulary into a phoneme sequence.
- It also can be obtained by using Grapheme-to-phoneme (G2P) tools. But G2P tools are also created based on pronunciation lexicons.

SPG (Speech-to-Phoneme-to-Grapheme) model architecture

- Two-stage ASR model
 - Speech-to-phoneme (S2P) model
 - Phoneme-to-grapheme (P2G) model
- Latent-variable model, **with phonemes being treated as discrete latent variables**
 - Grapheme-to-phoneme (G2P) model is introduced as an auxiliary inference model

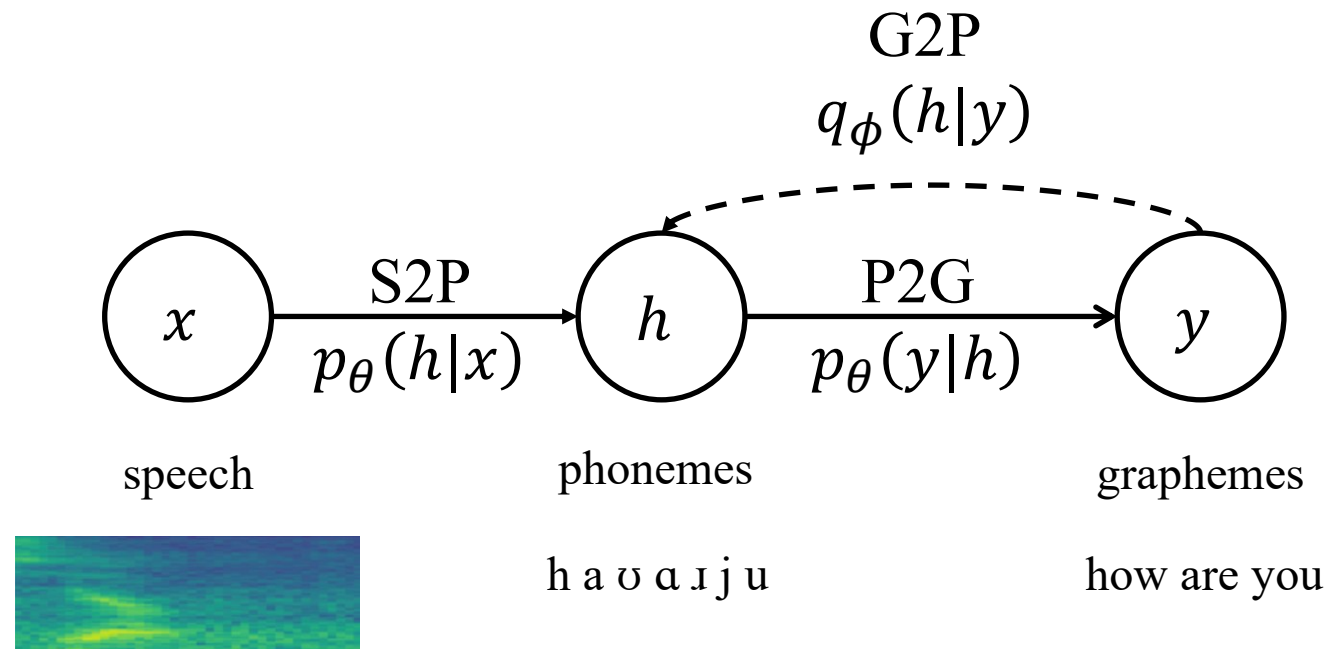


Fig. 1. Overview of the SPG model

EM (Expectation-Maximization) Algorithm

Maximum-likelihood parameter estimation for the latent-variable model $x \rightarrow h \rightarrow y$:

- Expectation of the log joint likelihood $\log P_{\theta}(h, y|x)$ under the conditional distribution $P_{\theta^{(old)}}(h|x, y)$

$$Q(\theta|\theta^{(old)}) = E_{P_{\theta^{(old)}}(h|x,y)}[\log P_{\theta}(h, y|x)] = \sum_h P_{\theta^{(old)}}(h|x, y) \log P_{\theta}(h, y|x)$$

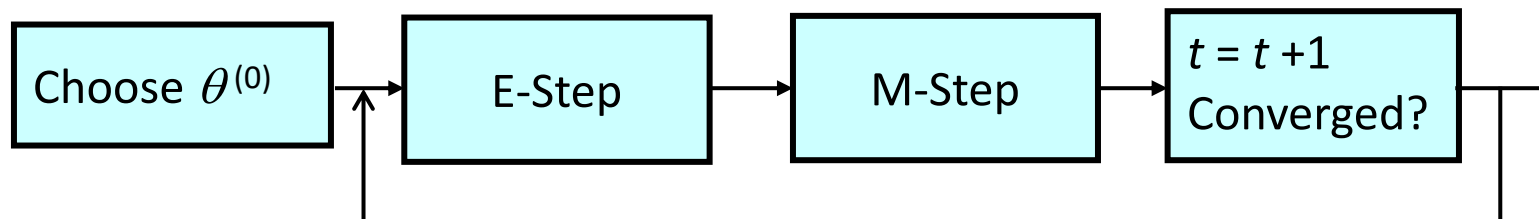
Infer h , using the current model

Supervised training with the estimated h , and update the model

- Solve $\theta^* = \operatorname{argmax}_{\theta} Q(\theta|\theta^{(old)})$

Holds $\log P_{\theta^{(old)}}(y|x) \leq \log P_{\theta^*}(y|x)$

- The EM algorithm is an iterative process



Connecting JSA to MCMC-SAEM

JSA = coupling an SA version of EM (SAEM) with an adaptive MCMC procedure (UAI 2020)

- $q_\phi(h|x)$ acts like an **adaptive proposal**, adjusted from past realizations of the Markov chain

SAEM-MCMC (Kuhn and Lavielle, 2004):

- Apply **Markov move** which admits $p_{\theta^{(t-1)}}(h|x)$ as the invariant distribution
- The proposal of the transition kernel is fixed, which may limit the MCMC efficiency

adaptive MCMC

sampling from $p_\theta(h|x)$ is intractable

SAEM (Delyon et al., 1999):

- **Monte Carlo sampling:** fill the missing values for latent variables through sampling $h' \sim p_\theta(h|x)$
- **SA updating:** perform gradient ascent over θ using $\nabla_\theta \log p_\theta(x, h')$

EM (Dempster et al., 1977):

- E-step: calculate $Q(\theta|\theta^{(t-1)}) = E_{p_{\theta^{(t-1)}}(h|x)}[\log p_\theta(x, h)]$
- M-step: update θ by $\max_\theta Q(\theta|\theta^{(t-1)})$

E is intractable

max is intractable

perform gradient ascent over θ

Technical innovation: JSA enables pronunciation-lexicon-free training for phoneme-based recognition

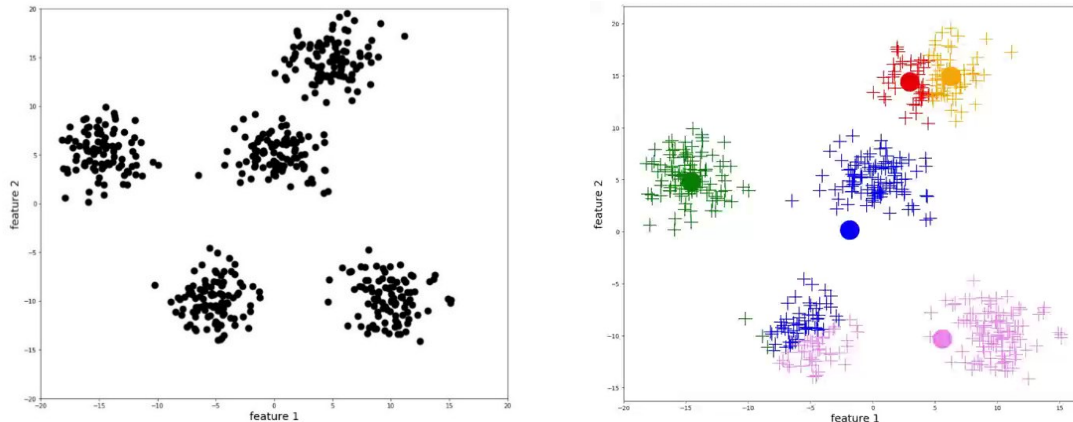
Technical limitation

- Classic methods heavily rely on supervised learning with manual annotations

Novel algorithm

- Build $x \rightarrow h \rightarrow y$ with h as a latent variable,
- Employ the **Joint Stochastic Approximation (JSA) learning algorithm** (UAI 2020)
- Train $x \rightarrow h \rightarrow y$ using only x and y
- **Overcome the efficiency bottleneck algorithmically**

Illustrative example: let x and y be one-dimensional, the EM algorithm can iteratively find h



Academic contribution of JSA:

it introduces an adaptive MCMC transition kernel to solve the **efficiency challenge of expectation calculation in high-dimensional spaces**

JSA (Ou & Song, 2020)

E-step introduces adaptive MCMC

SAEM-MCMC
(Kuhn and Lavielle, 2004)

E-step introduces Markov chain Monte Carlo

SAEM (Delyon et al., 1999)

E-step introduces Monte Carlo sampling

EM (Dempster et al., 1977)

Joint Stochastic Approximation

- Stochastic extension of EM algorithm.
- **Markov Chain Monte Carlo (MCMC) sampling (E-step in EM)**

1) Propose $h \sim q_\phi(h|y)$;

2) Accept h with probability $\min \left\{ 1, \frac{w(h)}{w(\tilde{h})} \right\}$, where

$$w(h) = \frac{p_\theta(h|x, y)}{q_\phi(h|y)} \propto \frac{p_\theta(h|x)p_\theta(y|h)}{q_\phi(h|y)}$$

- **Parameter updating (M-step in EM)**

Algorithm 1 The SPG-JSA algorithm

Input: S2P model $p_\theta(h|x)$, P2G model $p_\theta(y|h)$, G2P model $q_\phi(h|y)$, training dataset $\{(x, y)\}$

repeat

Draw a pair of speech and text (x, y) ;

Initialize $\tilde{h}$ by sampling from $p_\theta(h|x)$;

Monte Carlo sampling:

Sample h from the proposal $q_\phi(h|y)$;

Accept $\tilde{h} = h$ with probability
 $\min \left\{ 1, \frac{p_\theta(h|x)p_\theta(y|h)}{q_\phi(h|y)} / \frac{p_\theta(\tilde{h}|x)p_\theta(y|\tilde{h})}{q_\phi(\tilde{h}|y)} \right\}$;

Parameter updating:

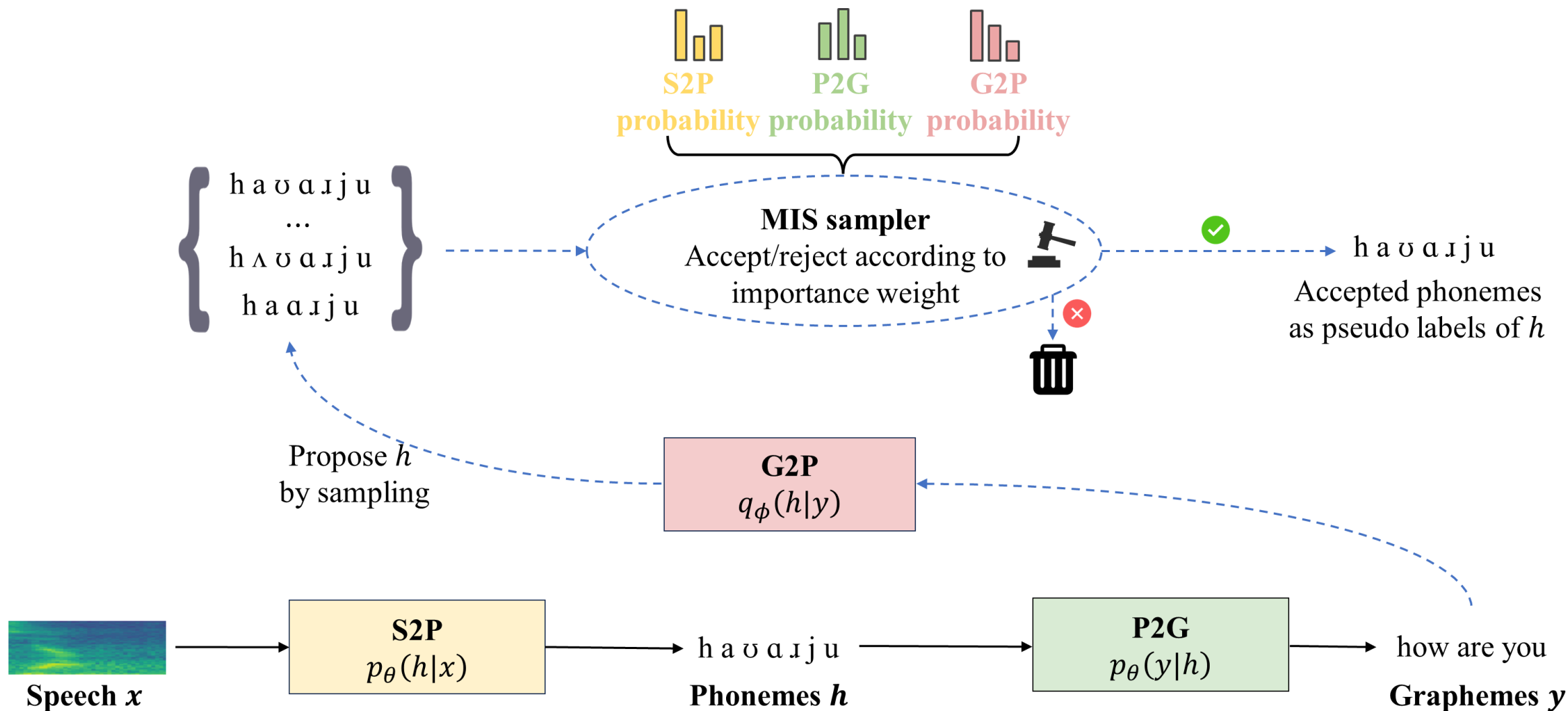
Updating θ by ascending: $\nabla_\theta [p_\theta(\tilde{h}|x)p_\theta(y|\tilde{h})]$;

Updating ϕ by ascending: $\nabla_\phi q_\phi(\tilde{h}|y)$;

until convergence

return θ and ϕ

Overview of JSA-SPG



Improving P2G via data augmentation

- During the JSA-SPG training, as the models gradually converge, the diversity of phoneme sequences sampled by MIS decreases. **The P2G model is gradually trained with less noisy input**, compared with the input fed to P2G in testing.
- In order to improve the robustness of the P2G model, we augment the P2G model **after the JSA-SPG training**.
- Particularly, we decode 128 best phoneme sequences by S2P beam search decoding over the training speech data and pair them with text labels, which serve as augmented data to **further train the P2G model**.

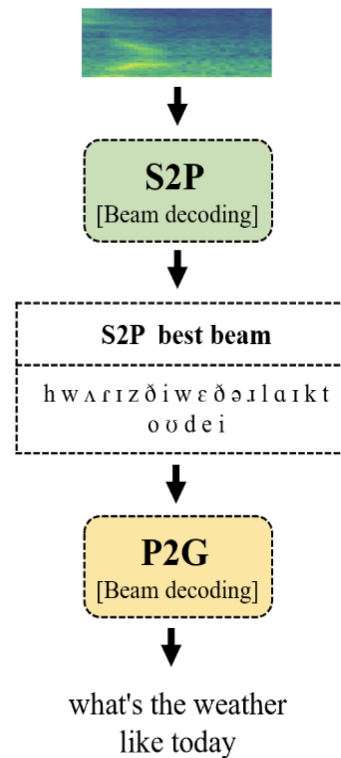
Marginal Likelihood Scoring (MLS)

Three ways of decoding:

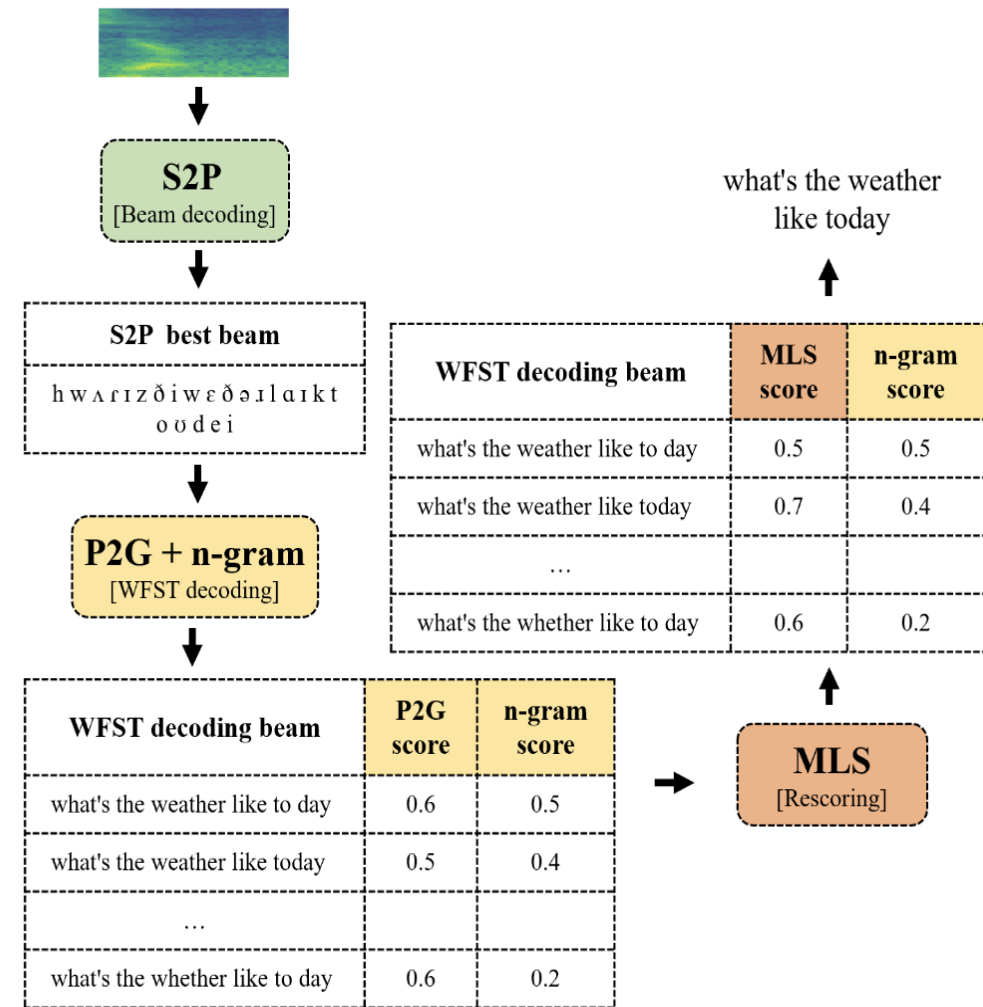
- Vanilla decoding without LM
- Vanilla decoding with LM
- MLS decoding
 - Generates an n-best list of candidates using WFST decoding
 - Rescored using marginal

$$y^* = \arg \max_{\hat{y}} \log \sum_{i=1}^l \frac{p_{\theta}(h_i|x)p_{\theta}(\hat{y}|h_i)}{q_{\phi}(h_i|\hat{y})} + \lambda \log P_{LM}(\hat{y})$$

Estimate the marginal likelihood with importance weights to align inference with the training objective



(a) Two-stage ASR decoding without LM



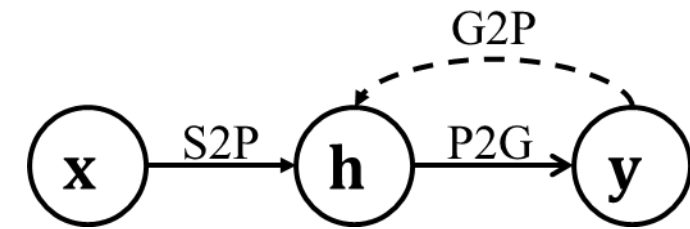
(b) Two-stage ASR decoding with MLS rescoring

Experiment Result

- Only 10 minutes of phoneme annotations (~100 utterances).
- JSA-SPG outperforms best cross-lingual fine-tuning approach using subword or full phoneme supervision.

Exp.	Polish				Indonesian			
	PER	w/o LM	w LM	<i>MLS</i>	PER	w/o LM	w LM	<i>MLS</i>
Monolingual phoneme [†]	2.82	NA	4.97	NA	5.74	NA	3.28	NA
Monolingual subword [†]	NA	19.38	7.12	NA	NA	31.96	10.85	NA
Whistle phoneme FT [†]	1.97	NA	4.30	NA	4.79	NA	2.43	NA
Whistle subword FT [†]	NA	5.84	3.82	NA	NA	12.48	2.92	NA
SPG with full phoneme labels + P2G augmentation	1.97	10.16	4.77	4.64	4.79	14.28	4.34	2.99
	1.97	5.67	4.86	4.63	4.79	7.29	3.47	2.69
SPG init from G2P + P2G augmentation	17.72	8.73	4.68	5.91	21.85	10.15	3.81	3.09
	17.72	5.93	4.97	5.88	21.85	6.34	3.44	2.91
JSA-SPG + P2G augmentation	17.35	8.19	4.65	3.93	20.66	9.04	3.26	2.47
	17.35	4.64	4.37	3.64	20.66	4.55	2.92	2.31

Language Domain Adaptation (LDA)



- In cross-domain ASR, a common approach is to train external language models for language domain adaptation.

Facilitate the construction of an ASR system for **a new domain** when only text y is available

- **LDA training**

- Use the G2P model to generate phoneme labels on cross-domain text data;
 - Further adapting the P2G model.
- LDA training performance **far exceeds** that of traditional language domain adaptation method by 9% error rate reduction (22.58 vs 20.57; 12.39 vs 11.23).

Exp.	Polish			Indonesian		
	w/o LM	w LM	<i>MLS</i>	w/o LM	w LM	<i>MLS</i>
Whistle subword FT on CV	33.46	22.58	NA	43.69	12.39	NA
JSA-SPG on CV	35.18	29.04	26.79	39.19	16.93	14.28
+ LDA training	28.87	23.84	20.57	30.69	12.68	11.23

Table 2. WER% of cross-domain language domain adaptation (LDA) experiments.

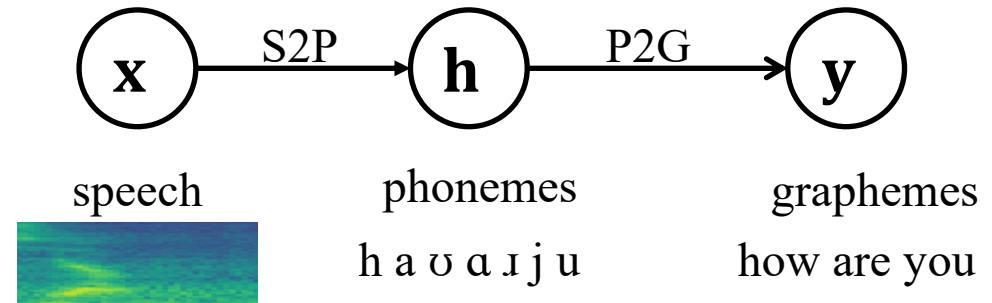
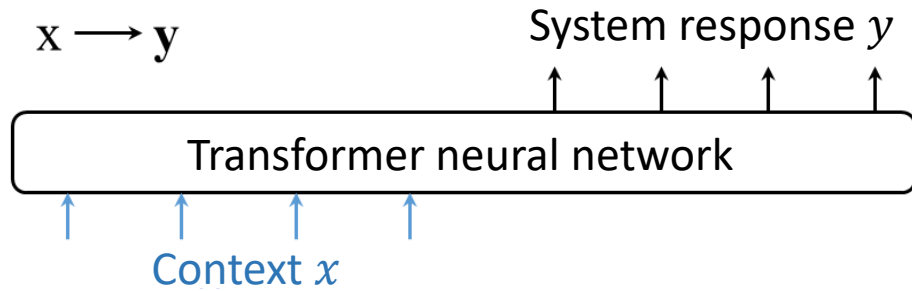
Experiment with different amounts of supervised phoneme data

- Only 2 minutes or even zero-shot, JSA-SPG obtains impressive results.
- As phoneme-annotated data increases, both PER and WER show monotonic reductions

Amount of supervised data	Polish				Indonesian			
	PER	w/o LM	w LM	<i>MLS</i>	PER	w/o LM	w LM	<i>MLS</i>
unsupervised	50.24	13.43	6.20	5.05	32.71	11.09	3.74	2.80
+ P2G augmentation	50.24	6.08	5.28	4.48	32.71	5.33	3.28	2.47
20 sentences (about 2 minutes)	27.35	8.25	5.17	4.25	27.37	9.01	3.39	2.66
+ P2G augmentation	27.35	5.31	4.78	3.96	27.37	5.45	3.04	2.47
100 sentences (about 10 minutes)	17.35	8.19	4.65	3.93	20.66	9.04	3.26	2.47
+ P2G augmentation	17.35	4.64	4.37	3.64	20.66	4.55	2.92	2.31

Table 3. SPG-JSA training result with different amounts of supervised phoneme data.

Principled unsupervised learning (rethinking)



GPT: realizes principled unsupervised learning of the probabilistic model $x \rightarrow y$ when only $\{x\}$ is available, **for single-modality problem (x, y are from the same modality)**

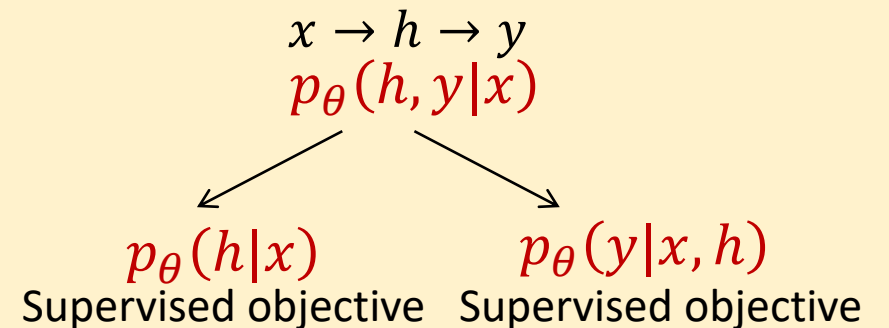
New problem (cross-modality): how can we principally solve unsupervised learning for the probabilistic model $x \rightarrow h \rightarrow y$ when only $\{(x, y)\}$ is available?



- Given only $\{(x, y)\}$, how should we train $x \rightarrow h \rightarrow y$?
- For a latent-variable model $p_{\theta}(h, y|x)$, estimate θ by maximizing the marginal likelihood $p_{\theta}(y|x)$.

$$p_{\theta}(y|x) = \sum_h p_{\theta}(h|x) p_{\theta}(y|x, h)$$

The supervised objective is part of unsupervised optimization, achieving principled unsupervised learning.



Subsection summary

- We aim to achieve phoneme-based speech recognition, **without pronunciation lexicons**.
- Introduce a new method called **JSA-SPG**, treating phonemes as discrete latent variables and using JSA for model training.
- Joint stochastic approximation (JSA) uses adaptive MCMC with a G2P proposal to jointly learn the S2P, P2G and G2P models.
- In crosslingual experiments with two languages (Polish and Indonesian), JSA-SPG demonstrates remarkable performance **compared to the best cross-lingual finetuning approach using subword or full phoneme supervision**.

Content

- I. Basics of Speech Recognition and LLMs (50 min)
- II. Continuous-interface based approach (50 min)
- III. Discrete-interface based approach (50 min)
- IV. Open questions and future directions (20 min)
 - 1. Open questions: data-efficient ASR with LLMs
 - 2. Q&A and discussion

Open questions: data-efficient ASR with LLMs

1. Cross-lingual and code-switching knowledge transfer

- Extremely low-resource languages → even those without a written tradition
- Written form not shared with high-resource ones → e.g., English pre-training to Lao

2. How good synthetic speech can do for data augmentation

- Synthetic data from different domains to enhance ASR robustness
- Synthetic data of low-resource languages or code-switching

3. Test-time adaptation to target domain/languages

- Low-resource domains or languages
- For fair or empathetic applications

Q&A and Discussion

Main References

1. Saierdaer Yusuyin, Te Ma, Hao Huang, Wenbo Zhao, Zhijian Ou. Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision. IEEE Transactions on Audio, Speech and Language Processing (TASLP), 2025. <https://github.com/thu-spmi/CAT/tree/master/egs/cv-lang10>
2. Saierdaer Yusuyin, Te Ma, Hao Huang, Zhijian Ou. Pronunciation-lexicon free training for phoneme-based crosslingual ASR via joint stochastic approximation. IEEE Transactions on Audio, Speech and Language Processing (TASLP), 2026. <https://github.com/thu-spmi/CAT/tree/master/egs/JSA-SPG>
3. Ilya Sutskever, Rafal Jozefowicz, Karol Gregor, Danilo Rezende, Tim Lillicrap, Oriol Vinyals. Towards Principled Unsupervised Learning. arXiv:1511.06440, 2015.
4. Te Ma, Min Bi, Saierdaer Yusuyin, Hao Huang, Zhijian Ou. LLM-based phoneme-to-grapheme for phoneme-based speech recognition. Interspeech, 2025. <https://github.com/thu-spmi/CAT/blob/master/egs/llm-p2g/README.md>
5. Ziwei Li, Lukuang Dong, Saierdaer Yusuyin, Xianyu Zhao, Zhijian Ou. Phonemes vs. projectors: An investigation of speech-language interfaces for LLM-based ASR. arXiv:2604.09332, 2026.
6. Saierdaer Yusuyin, Nanling Jiang, Hao Huang, Zhijian Ou. Latent Softmax for data-efficient phoneme-based multilingual ASR across tonal and non-tonal languages. IEEE Spoken Language Technology Workshop (SLT), 2026.
7. Zhijian Ou, Yunfu Song. Joint stochastic approximation and its application to learning discrete latent variable models. Conference on Uncertainty in Artificial Intelligence (UAI), 2020.
8. Graves, et al., “Connectionist Temporal Classification: Labelling unsegmented sequence data with RNNs”, ICML 2006.
9. Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems (NeurIPS), 2020.
10. S. Fong, et al., “Speech LLMs in Low-Resource Scenarios: Data Volume Requirements and the Impact of Pretraining on High-Resource Languages,” in Proc. Interspeech 2025, 2003-2007.
11. V. T. Pham et al, “A comprehensive solution to connect speech encoder and large language model for ASR”, arXiv preprint arXiv: 2406.17272, 2024.
12. Y. Xu, et al, “Comparing Discrete and Continuous Space LLMs for Speech Recognition”, in Proc. Interspeech, 2024, pp. 2509-2513
13. C. Chen, et al, “Hyporadise: An open baseline for generative speech recognition with large language models,” in Proc. Adv. Neural Inf. Process. Syst., vol. 36, 2023, pp. 31 665–31 688.

Thanks for your attention !

Tutorial website:

<https://oa.ee.tsinghua.edu.cn/~ouzhijian/INTERSPEECH2026/>