Syllable-Text Interleavingに基づく音声言語モデルの効率的スケーリング

日本音響学会九州支部第3回オンラインセミナー

小松 亮太（東京科学大学）

Institute of SCIENCE TOKYO

2026/02/27

自己紹介

- 経歴
  - 2021年~2023年 東京工業大学 修士課程
  - 2023年~2025年 日立製作所 研究開発グループ
  - 2025年~ 東京科学大学 博士課程

- 研究分野
  - 音声表現学習
    - 音声エンコーダから抽出した潜在音声表現の量子化による**離散音声トークン**の学習
  - 音声言語モデル
    - 学術機関の限られた計算資源のもとで**効率的に**高性能な基盤モデルを構築

講演の構成

- **チュートリアル**
  - End-to-end音声言語モデルの学習効率に関する課題
  - Speech-text interleavingによるテキストから音声への知識転移
  - 大規模学習を可能にする実装技術

- **我々の取り組み**
  - 話者分離音節トークンの学習
  - Syllable-text interleavingに基づく音声言語モデルの効率的スケーリング

- **今後の課題**
  - 音声・テキスト間での表現分離

End-to-End音声言語モデルの学習効率に関する課題

音声領域におけるFew-shot学習能力の創発

- 2025年末から，GPT3で示されたスケーリングによるFew-shot学習能力が動画・音声でも確認されつつある

NeurIPS'20

Language Models are Few-Shot Learners

Google DeepMind 2025-10-1

Video models are zero-shot learners and reasoners

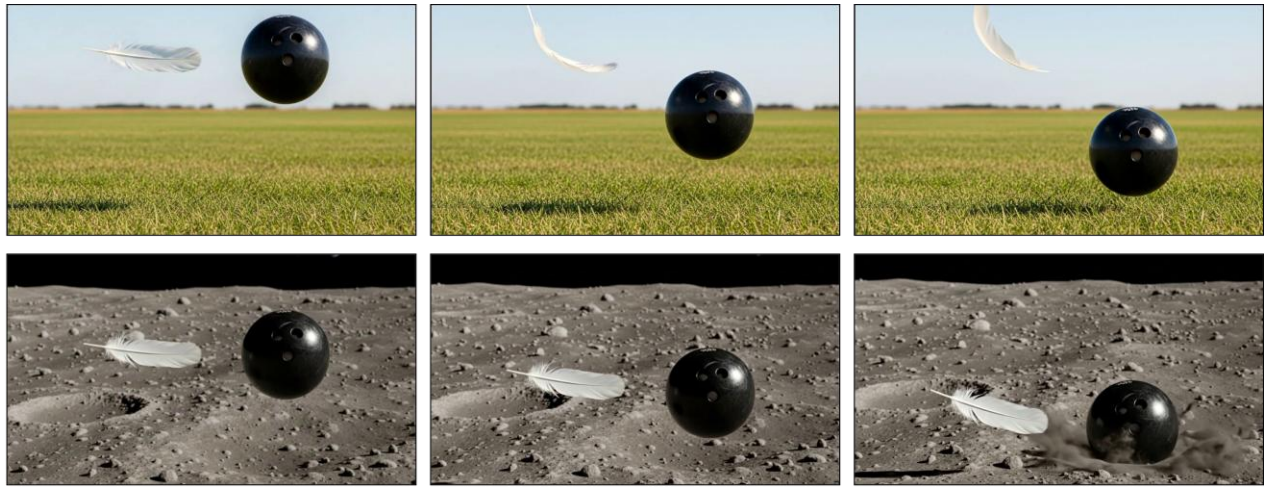


Figure 23 | **Gravity and air resistance. On earth** (top). Prompt: *"The objects fall due to gravity. Static camera, no pan, no zoom, no dolly."* Success rate: 0.5. **On the moon** (bottom). Prompt: *"The objects fall down on the moon due to gravity. Static camera, no pan, no zoom, no dolly."* Success rate: 0.5.

2025-12-29

MiMo-Audio: Audio Language Models are Few-Shot Learners

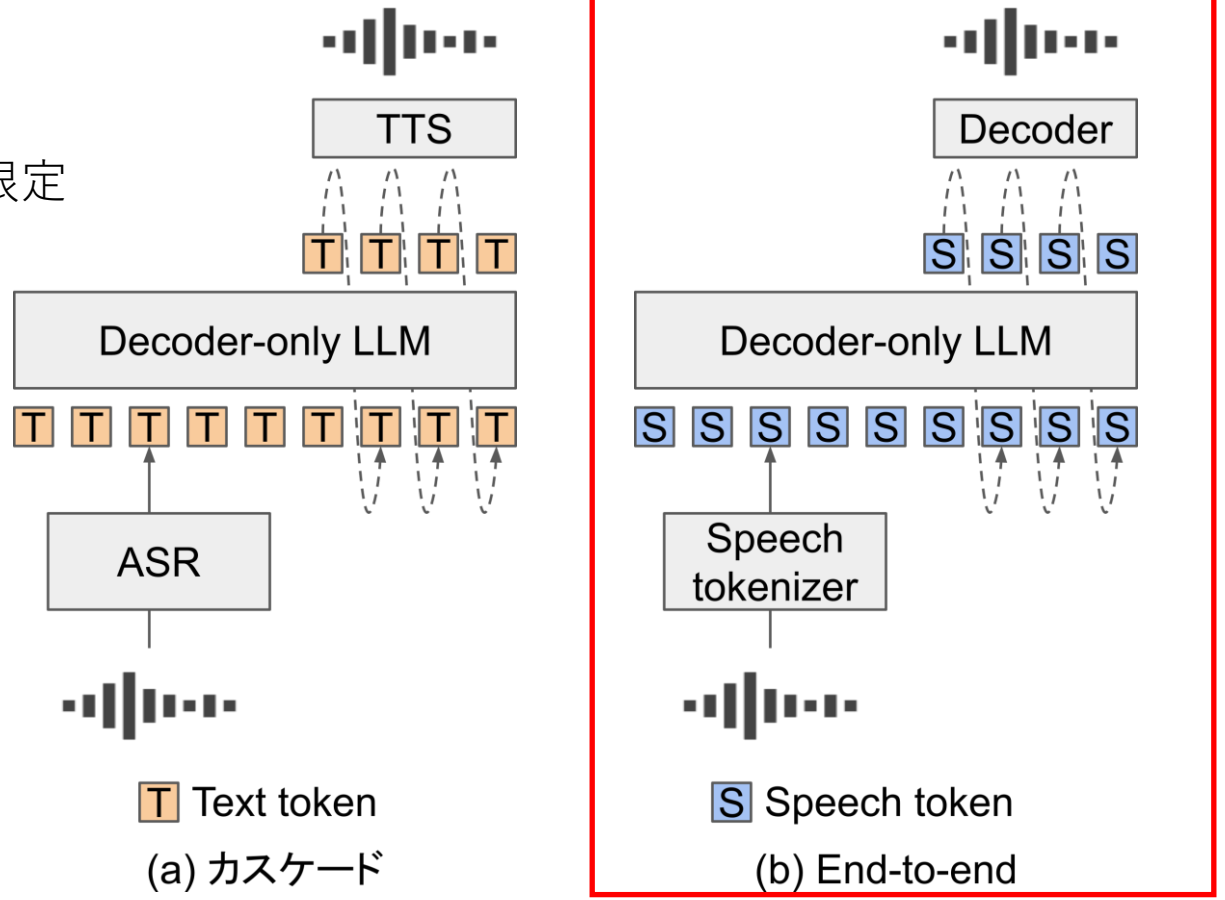
LLM-Core Xiaomi

感情変換（喜び→悲しみ）　　環境音付加（雷）

End-to-End音声言語モデル

- カスケード
  - 音声認識（ASR）→LLM→音声合成（TTS）
  - モジュール化に優れる
  - 処理対象はテキストで書き起こし可能な情報に限定
- End-to-end
  - LLMで音声トークンを直接理解・生成
  - 感情・話者情報も統一的に理解

本講演で扱うモデル

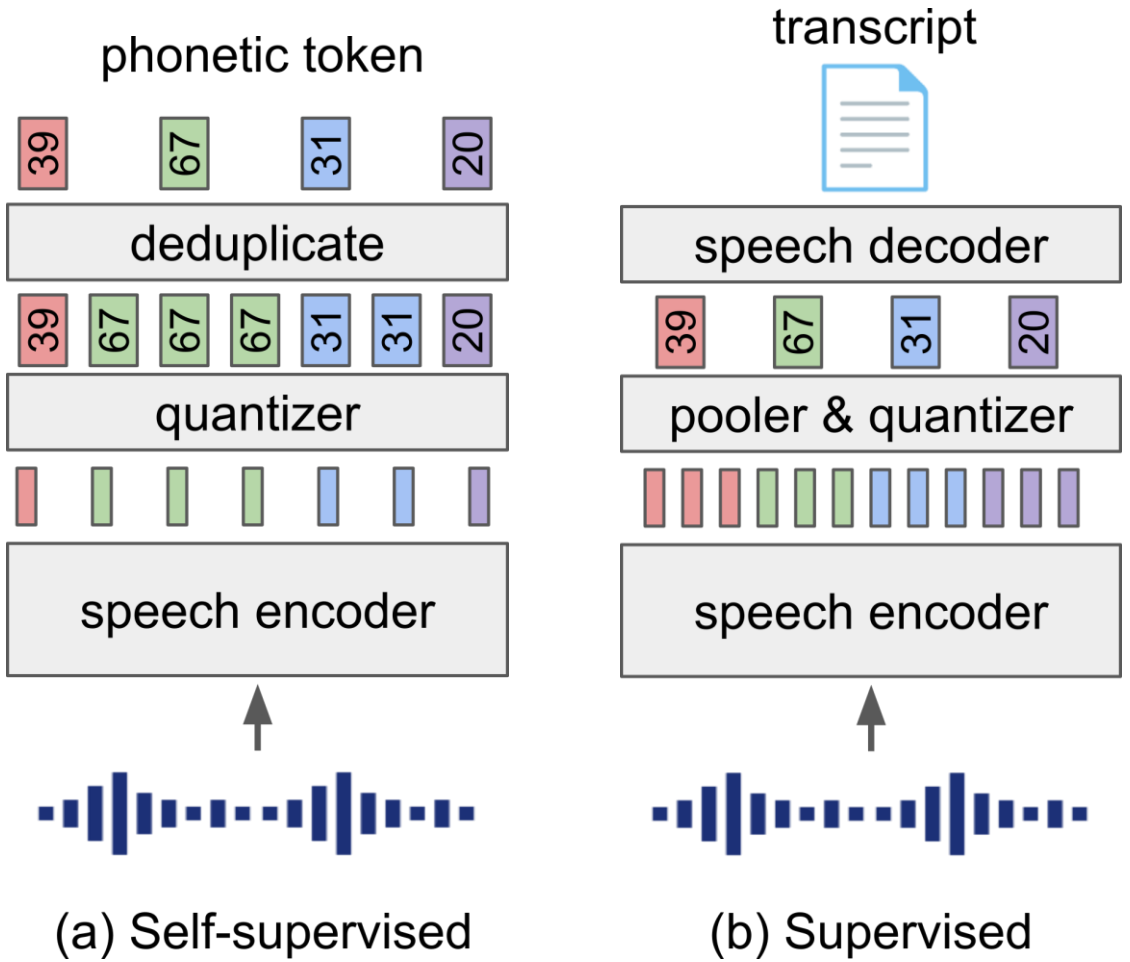


Phonetic token

- 音声エンコーダで抽出した潜在フレーム表現を量子化したクラスタID

- **Self-supervised**: HuBERT [Hsu+, TASLP'21]
  - オフラインK-meansクラスタリング
  - 連続する同一IDから重複除去

- **Supervised**: GLM-4-Voice [Zeng+, ICLR'25]
  - 明示的に言語情報を抽出
  - 音声エンコーダにベクトル量子化層を挿入
  - ASRでコードブックをオンライン学習



音声トークンの粒度

- 事前学習済みHuBERTの特徴量をそのまま量子化した25-50Hzの音素レベルのトークンが広く用いられる
- ビットレートと保持する情報量はトレードオフがあるため，用途に応じて使い分け

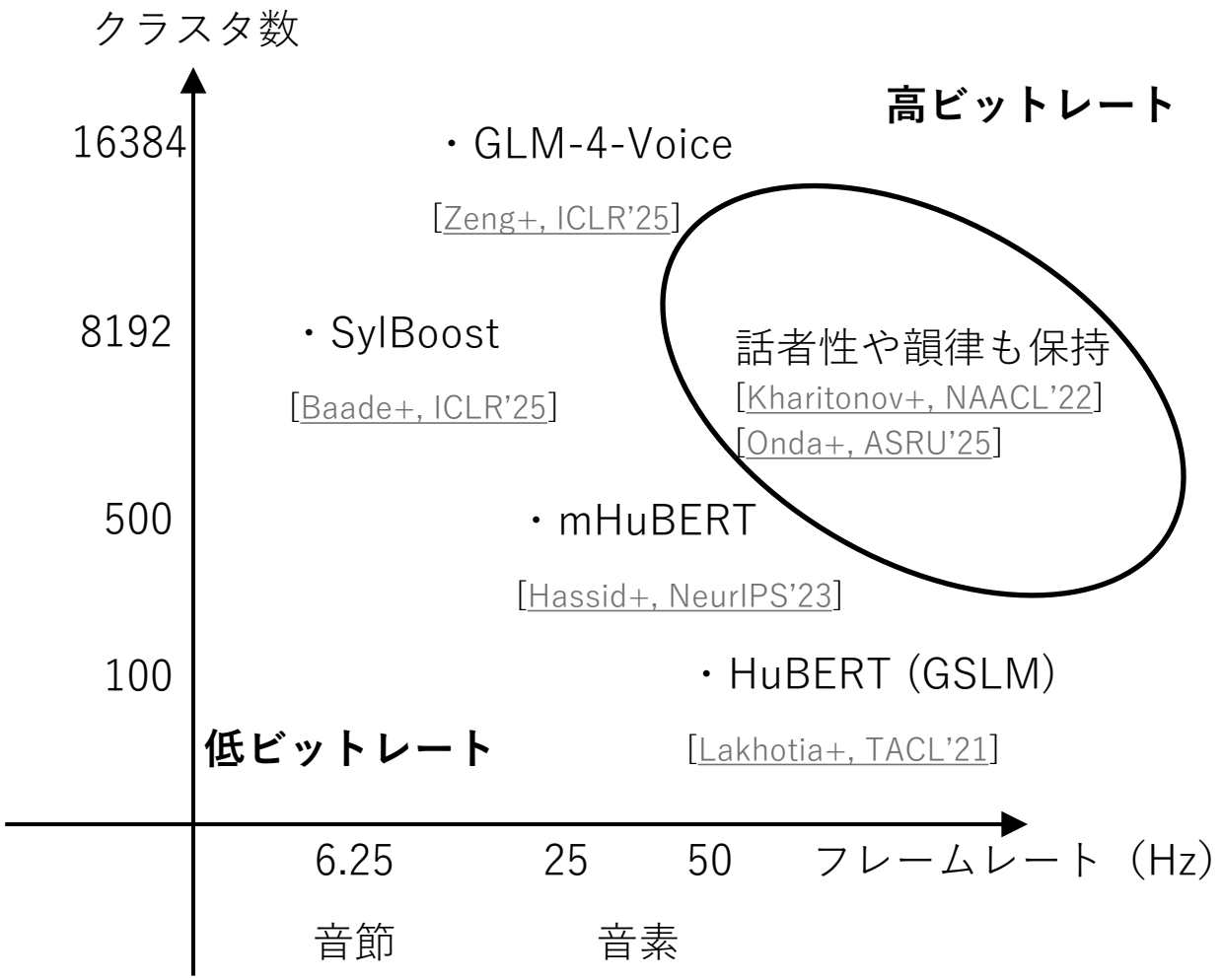


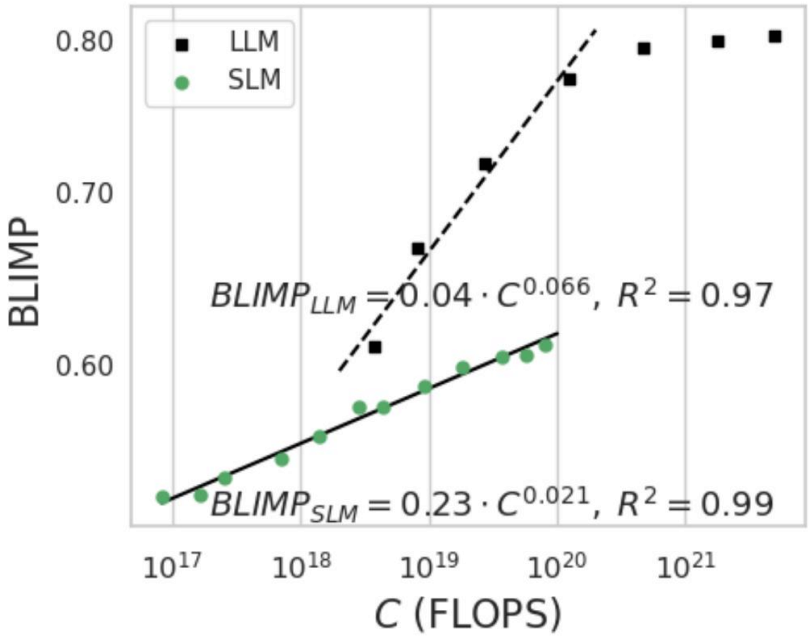
Table 1. Units Interpretation results. For phonemes, higher is better. While for the speaker and gender, a lower score indicates that the units success 'ignores' this information.

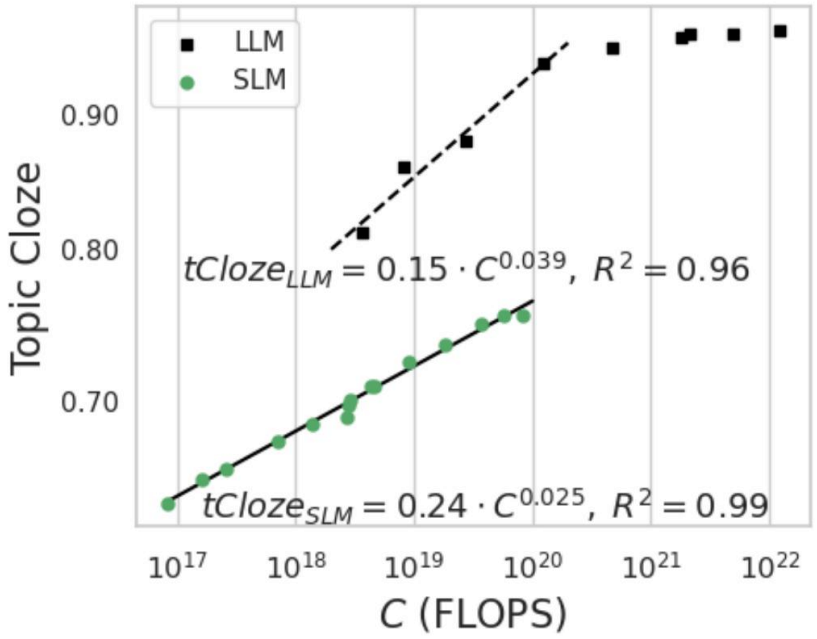
Model	Size	Speaker	Gender	Phoneme	
CPC	50	1.35	0.66	47.30	
	100	2.35	0.54	**48.45**	
	200	3.70	1.62	47.74	↓ 劣化
	2000	10.39	4.14	44.06	
HuBERT	50	0.73	0.03	42.49	
	100	1.41	0.17	45.48	
	200	1.95	0.21	46.64	↓ 劣化
	2000	5.15	0.65	43.32	

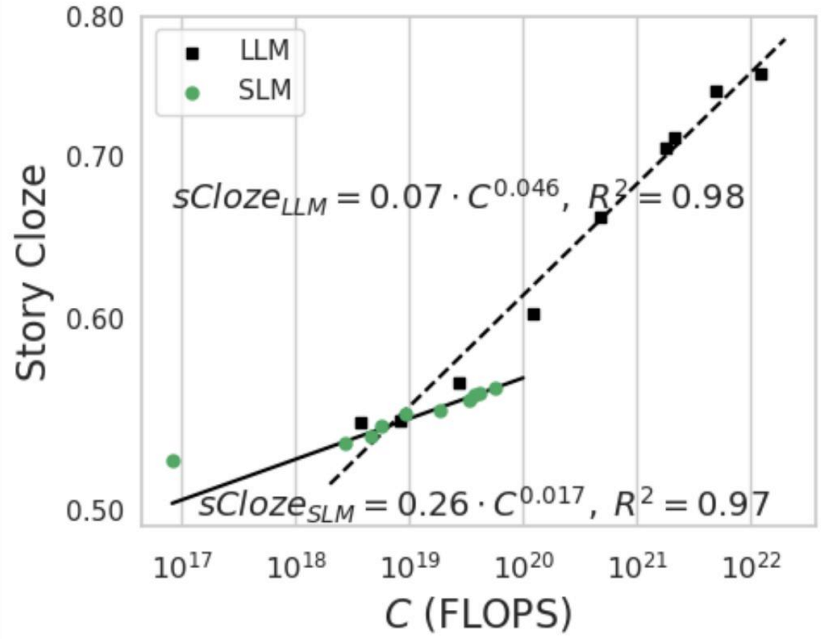
[Sicherman+, ICASSP'23]

Speech-onlyモデル（GSLM）の問題

- 乱数初期化された言語モデルを音声トークンのみで学習 [Lakhotia+, TACL'21]
  - テキストラベルが不要であるため，データセット構築は比較的容易
  - LLMの言語知識を活用不可
    - ➢ テキストLLMと比較して言語性能の**スケールは最大10^3のオーダーで遅い** [Cuervo+, EMNLP'24]
    - ➢ このままモデルとデータサイズをスケールさせるのは筋が悪い







BLIMPタスク：言語モデルに2つの音声トークン系列を入力したとき，文法が正しい方の尤度が大きければ正解

Topic/Story Clozeタスク：物語の2つのエンディングのうち，正しい方の尤度が大きければ正解

End-to-End音声言語モデル特有の課題

- Speech-only End-to-End音声言語モデルの学習効率が悪い本質的要因
  - LLMが処理する言語単位における情報密度

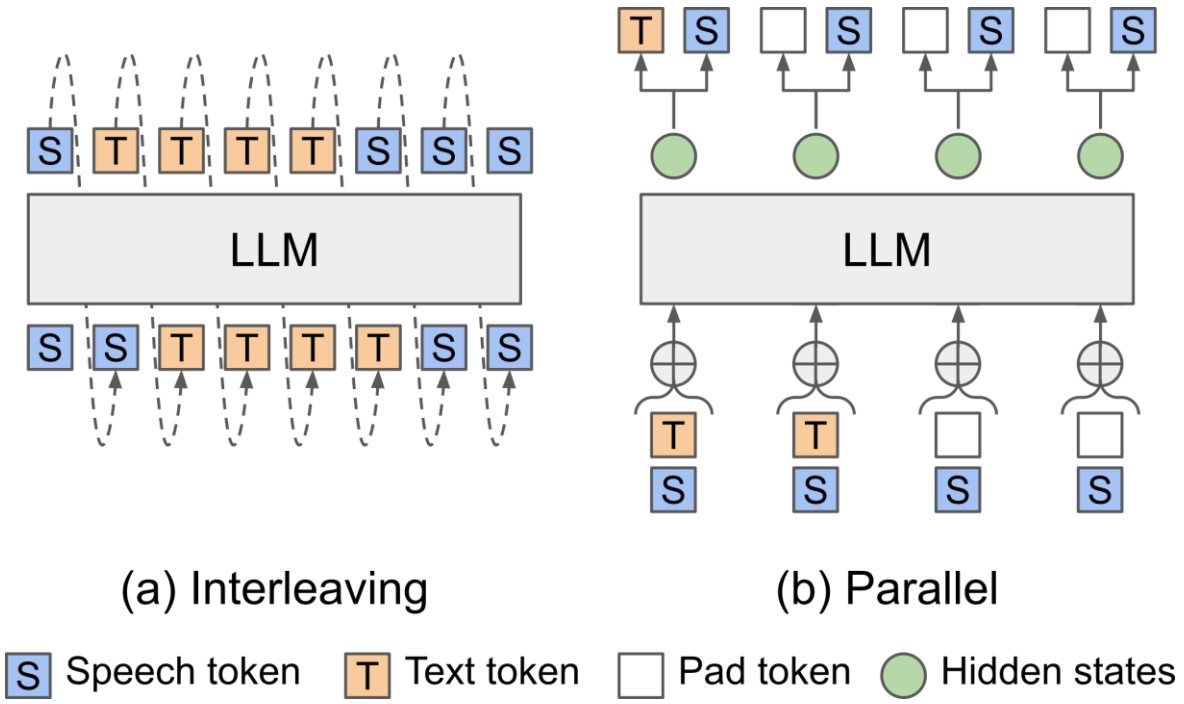
テキスト	音声
離散	連続
意味単位と強く対応	同一意味に対して多様な音響的実現値
高密度情報	時間的冗長性が高い

- カスケードにはないEnd-to-End特有の課題が生じる
  1. LLMの推論能力を保持しつつ音声へ知識転移するスケーラブルな学習戦略
     - Speech-text interleaving（前半の内容）
  2. 情報密度の高い音声トークンの学習
     - Syllabic token（後半の内容）

Speech-text interleavingによるテキストから音声への知識転移

テキストからの知識転移による学習効率化

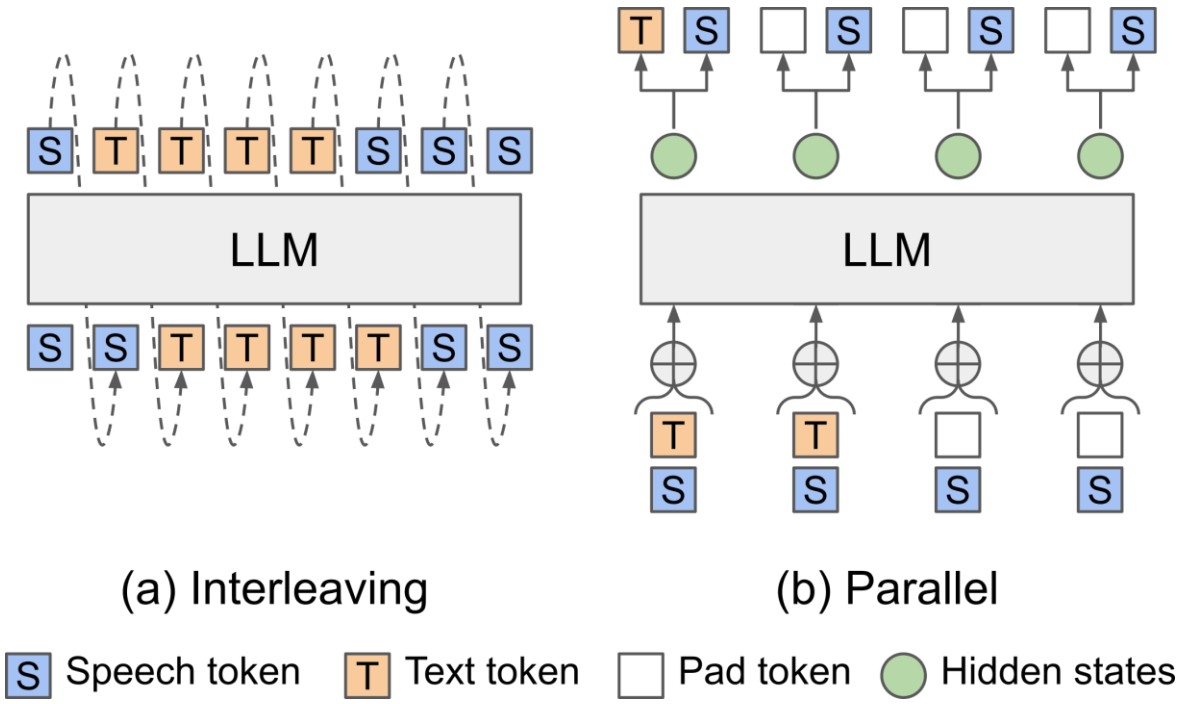
- **Speech-text interleaving**: SpiRit-LM [Nguyen+, TACL'25]
  - 事前学習済みテキストLLMを，**音声とテキストを交互に入れ替え**た文で学習
  - 例えば，“The capital of Japan is <6>”という文では，音声トークン<6>がTokyoだと**知識転移**できる
- **Parallel**: Moshi [Défossez+, arXiv'24]
  - LLM最終層の潜在状態に対して，アライメントした音声と書き起こしを**それぞれのヘッドで並列に次トークン予測**
  - テキストによるガイダンス（Inner monologue）で応答の質を保ちつつ，並列予測によってリアルタイム生成を実現



精度面ではInterleavingが優れる[Wu+, ASRU'25]

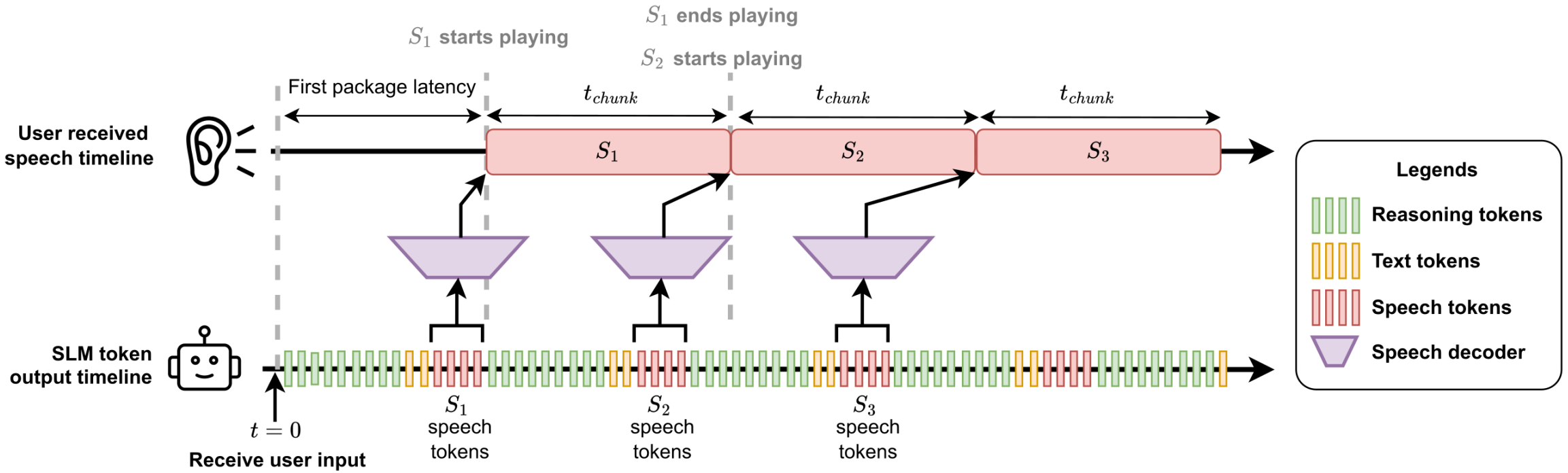
- 並列型では，入力において**音声とテキストの埋め込みを加算**
  - ➢ **モダリティが干渉**し，ベース言語モデルのテキスト推論能力が損なわれる恐れがある

Pattern	LLaMA Questions			Trivia-QA			Web Questions			
	Text(%) ↑	Speech(%) ↑	Rel. ↑	Text(%) ↑	Speech(%) ↑	Rel. ↑	Text(%) ↑	Speech(%) ↑	Rel. ↑	WER(%) ↓
Interleave	**64.67**	**57.33**	0.89	24.76	23.20	0.94	27.29	26.20	0.96	5.11
Parallel	60.67	46.67	0.77	22.51	19.59	0.87	26.99	19.67	0.73	17.94



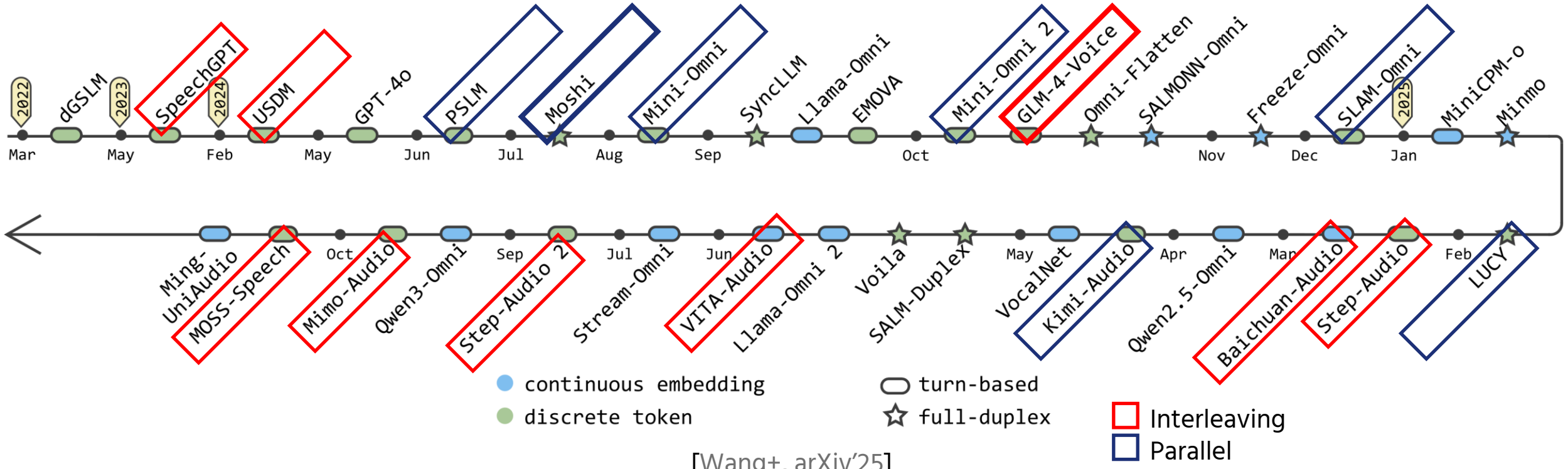
STITCH: 推論時における低遅延Interleaving [Chiang+, ICLR'26]

- Interleavingは推論時にもChain-of-thought（CoT）のような形で適用可能
  - 音声翻訳では，ソース音声→その書き起こし→翻訳音声の順に生成することで精度向上 [Hu+, ICASSP'25]
- テキスト生成により音声出力が**遅延する問題**がある
  - STITCHでは，テキストと約2秒の音声チャンクを交互に生成
  - **音声チャンクを再生中に次チャンクに対するCoTを並列に実行**することで遅延を低減



時系列でみるデコード方式の変遷

- 2024年から2025年初頭までは，MoshiによるFull-duplex化の流れからParallel型が注目された
- GLM-4-Voice [Zeng+, ICLR'25]がInterleaving方式のスケーラビリティを示して以降，多くの最先端モデルがInterleaving方式を採用

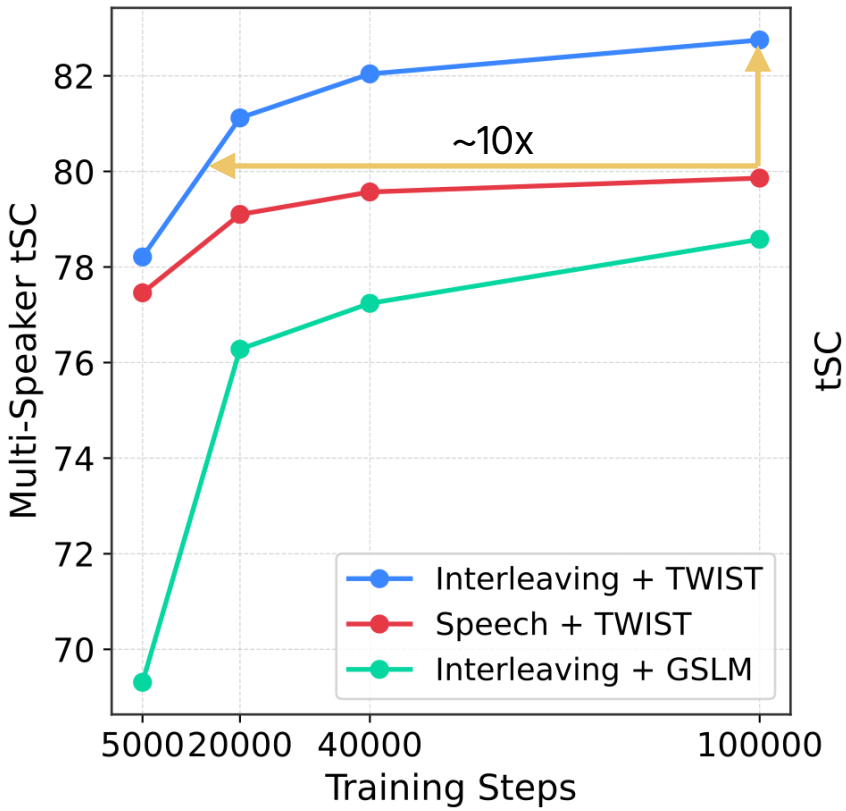


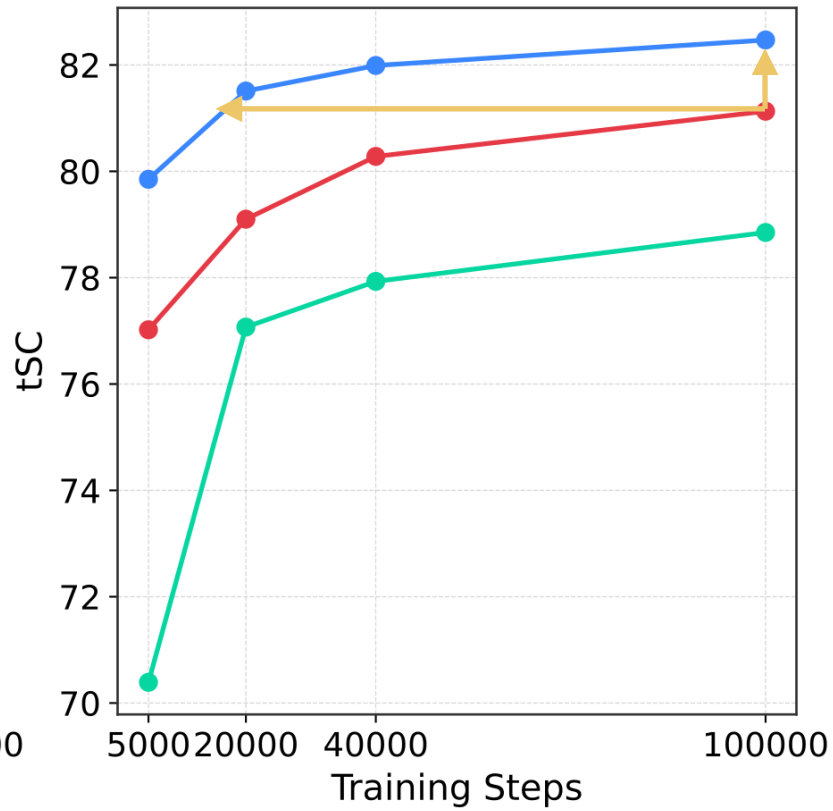
[Wang+, arXiv'25]

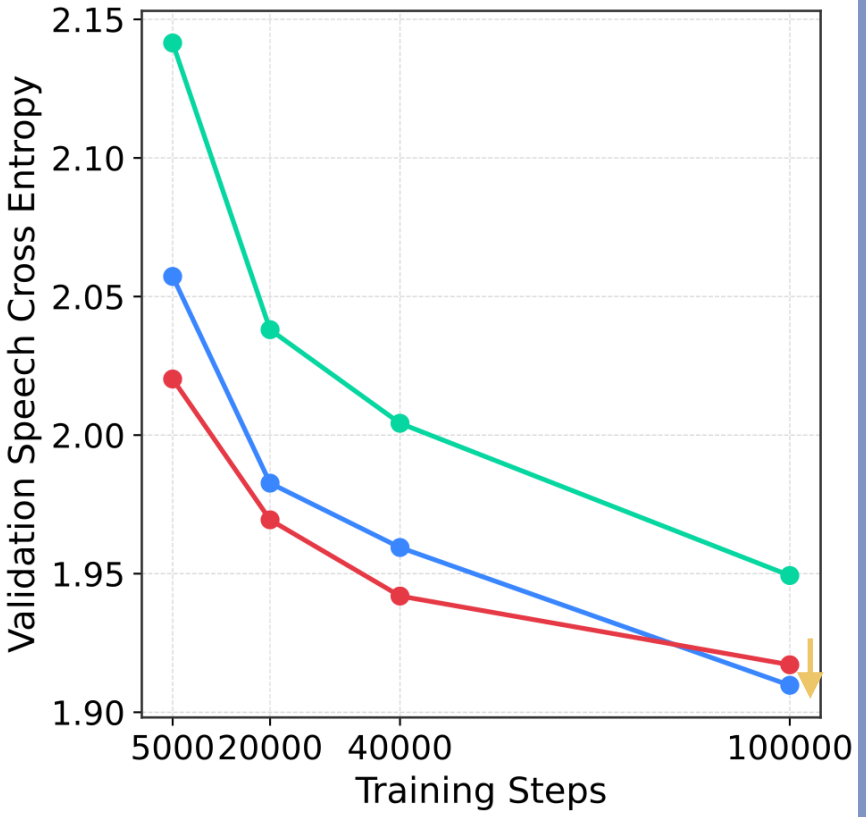
Speech-text interleavingのスケーリングへ向けて

Interleavingの効果 [Maimon+, COLM'25]

- 事前学習済みテキスト言語モデルで初期化する場合（TWIST[Hassid+, NeurIPS'23]），音声のみより **interleavingで学習した方が音声言語の意味理解精度は向上する**
- 学習効率も約10倍向上
- 音声トークンのみでのクロスエントロピーはタスク精度と相関するとは限らない

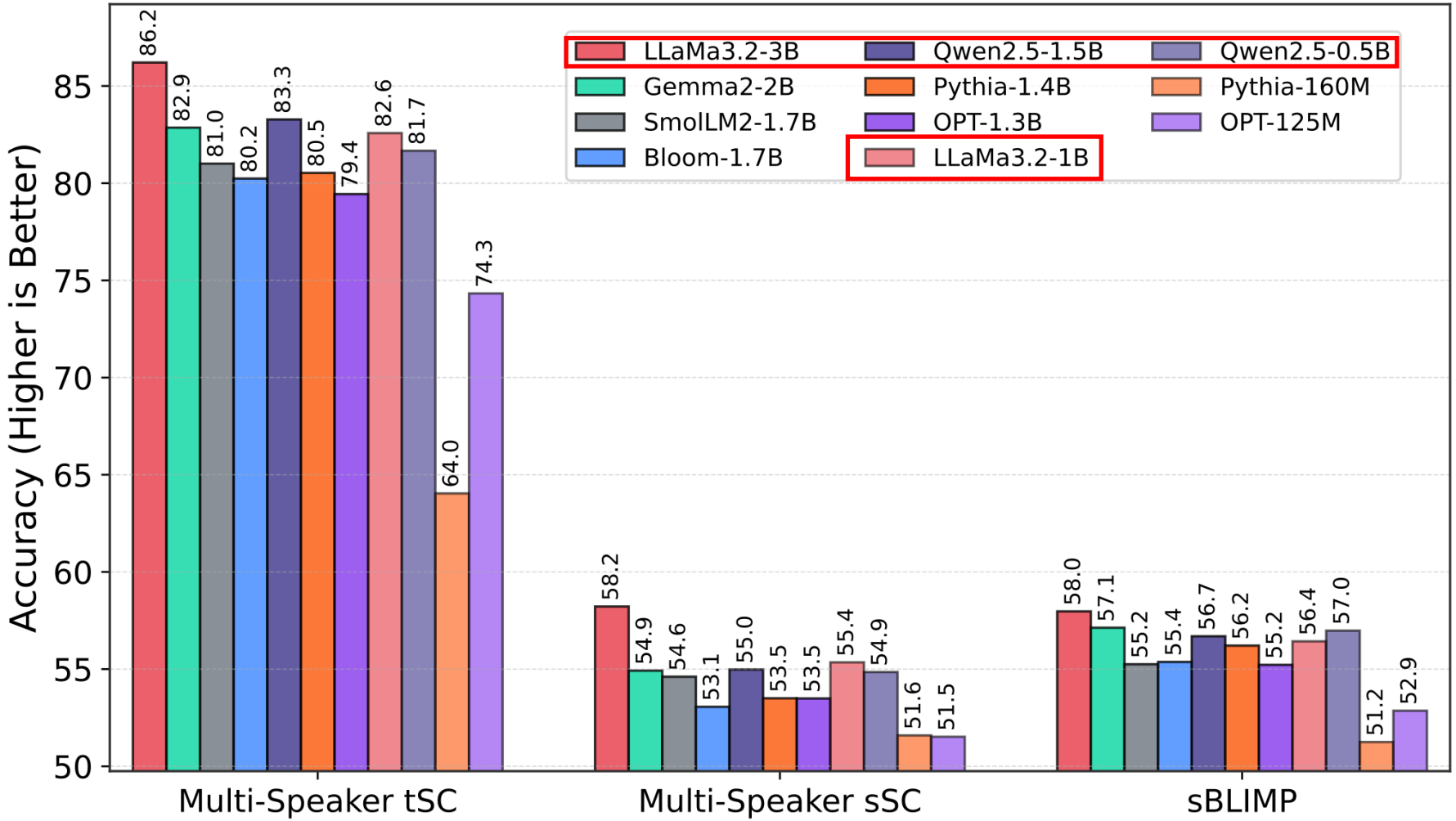






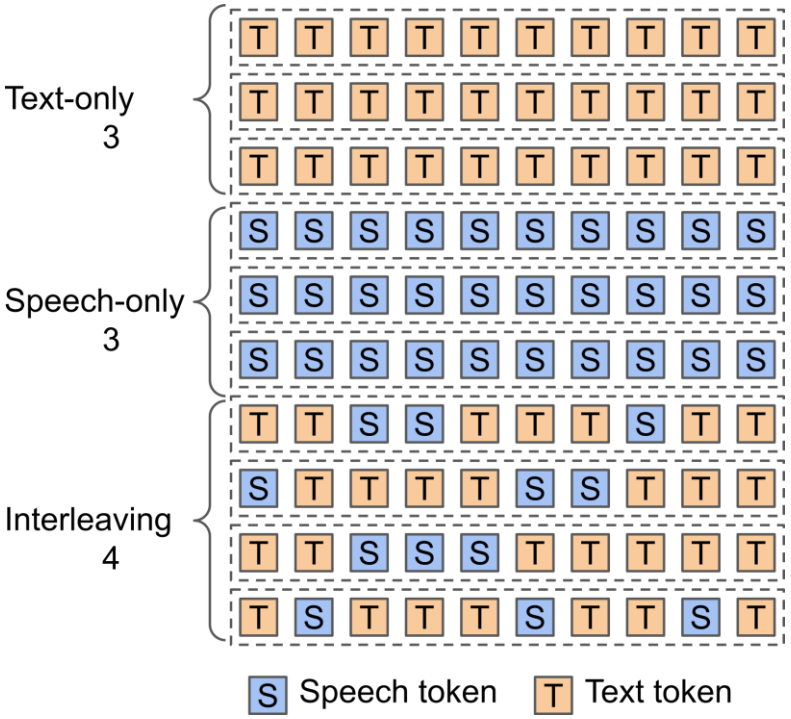
初期化に適した事前学習済みLLM [Maimon+, COLM'25]

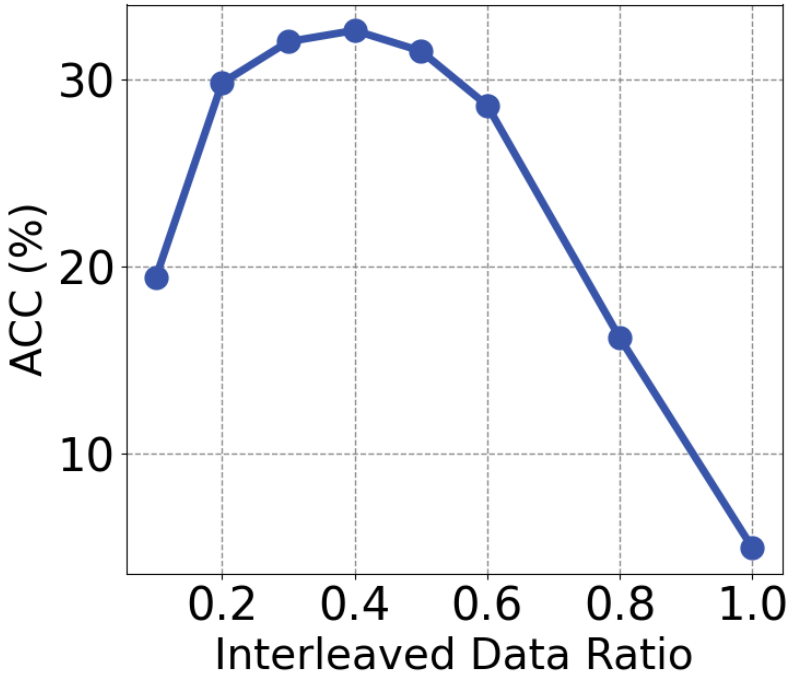
- 同じモデルサイズの中では，**LlamaあるいはQwen**ファミリが高い音声言語理解性能を示す
- 音声言語理解性能はベースLLMのテキストでの性能と**相関するとは限らない**

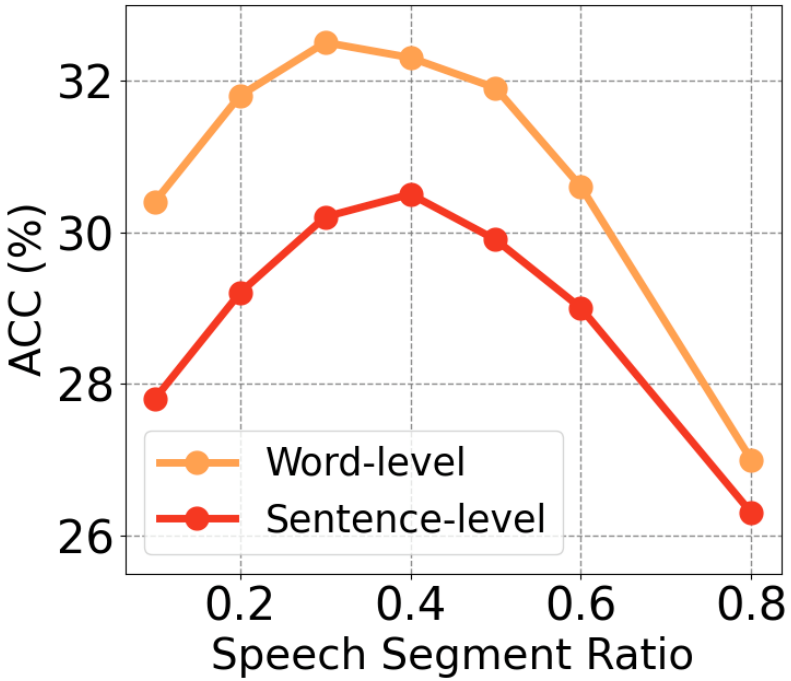


Interleavingに最適なモダリティ混合比 [Wang+, ACL'25]

- 破滅的忘却を防ぐため，テキストトークンのみの系列，音声トークンのみの系列，Interleaving系列を混合
- 単語レベルでのInterleaving系列内では，音声3：テキスト7の比率が最適（ [Zeng+, ICLR'25] とも一貫）
- Interleavingの単位として，単語か文どちらが最適かは結果が分かれる[Luo+, AAAI Workshop'26], [Udandarao+, ICLR'26 (AQ)]

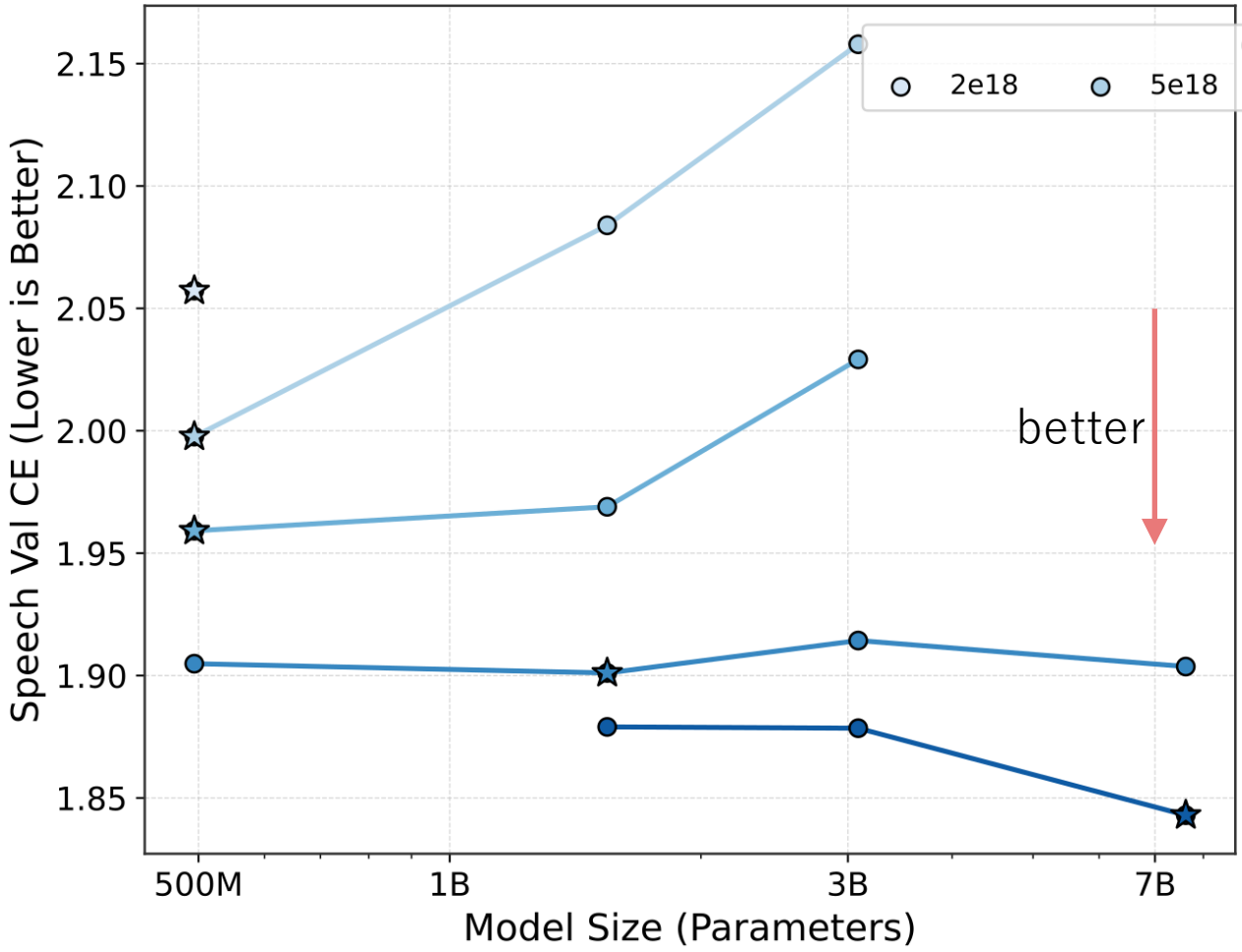


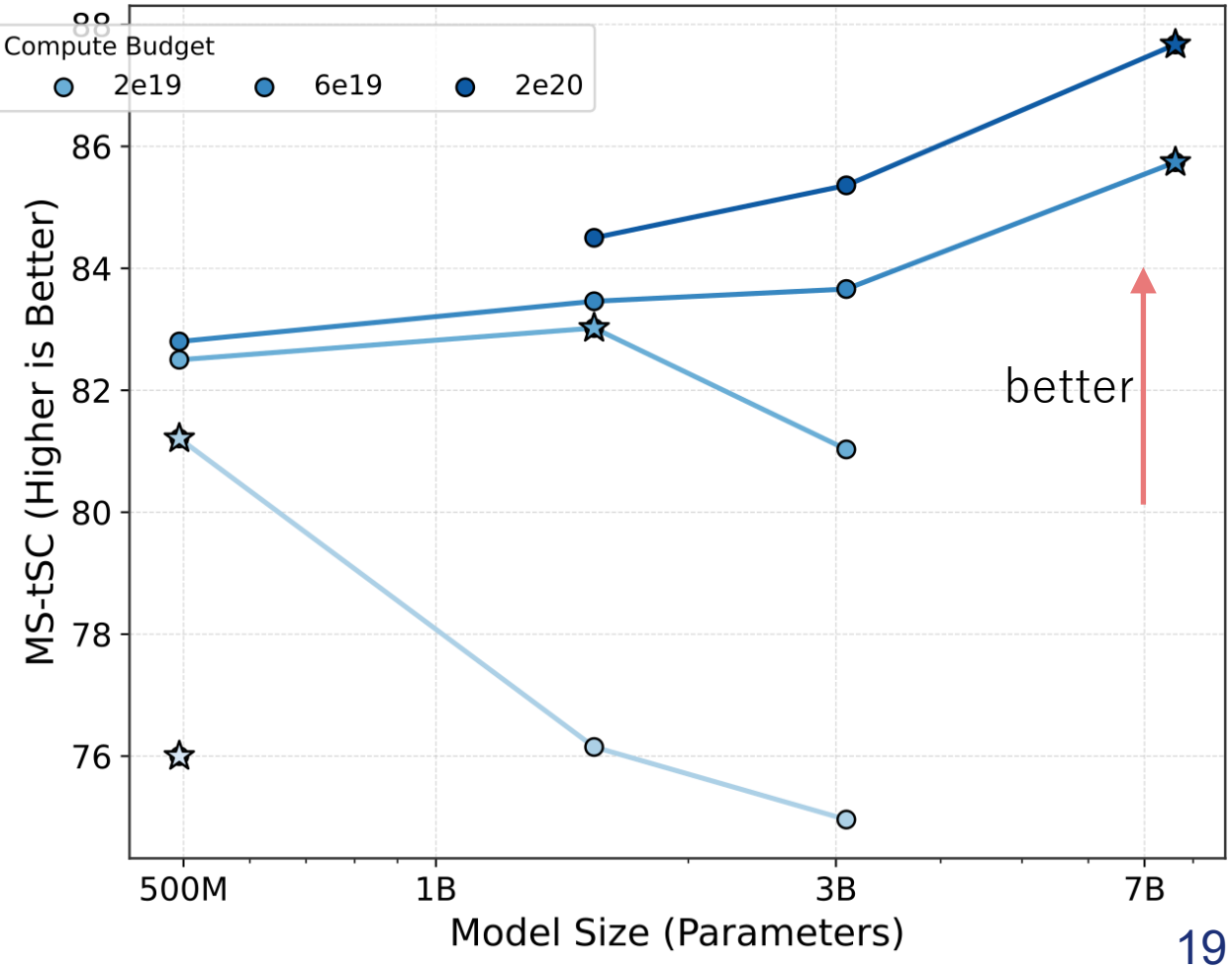




モデルサイズに関するIsoFLOPs分析 [Maimon+, COLM'25]

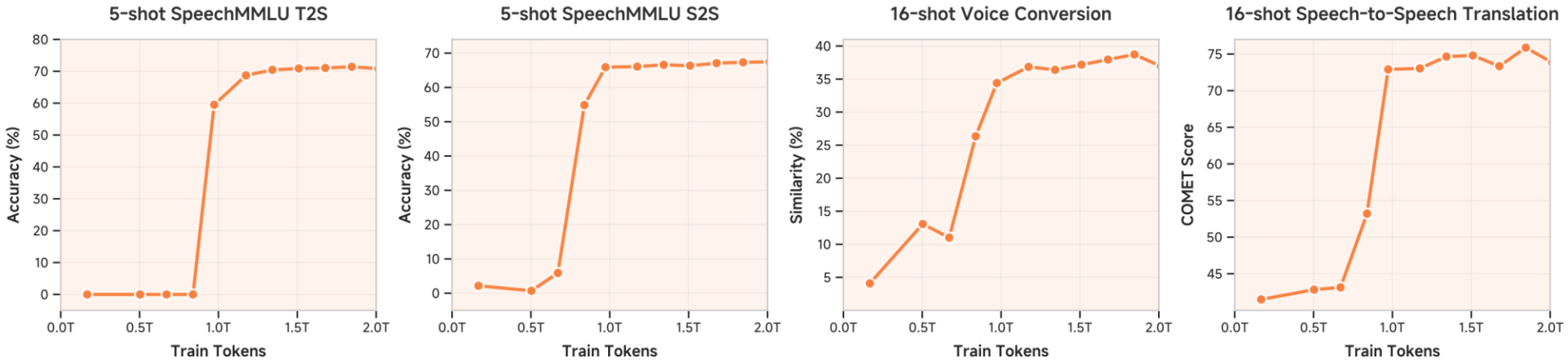
- 6e19 FLOPs以上の計算資源がある場合，500Mモデルを学習するより，トークン数を減らしてでも7Bモデルを学習した方がよい
- ご参考：TSUBAME 4.0で6e19 FLOPsを換算すると，1ノード（4 H100 GPUs）× 14.4時間

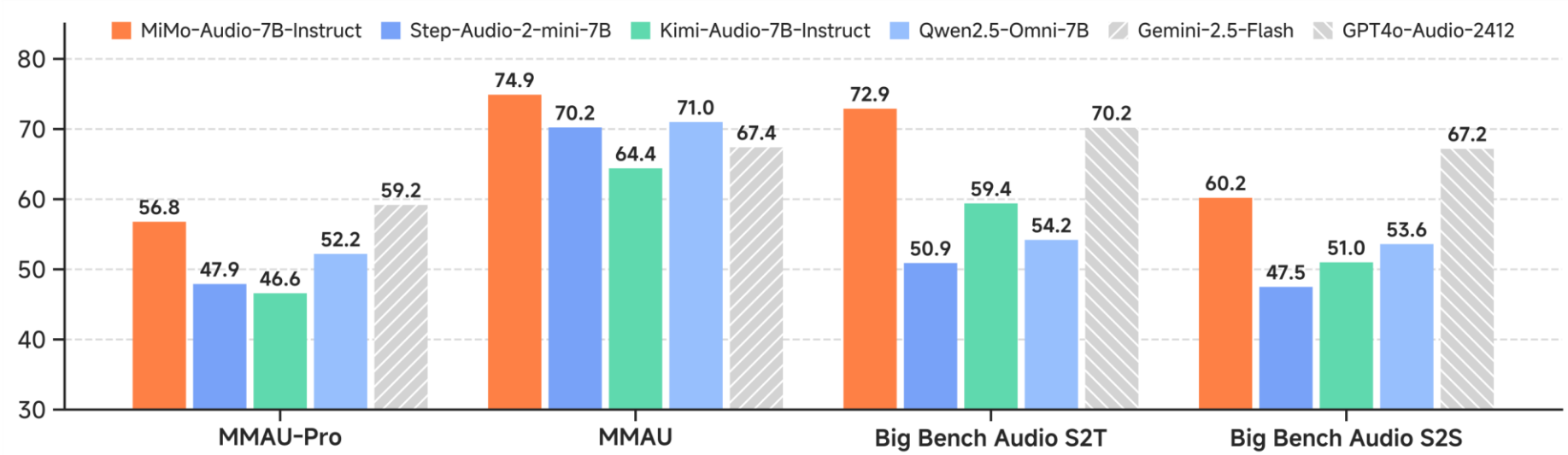




Mimo-Audio: Audio Language Models are Few-Shot Learners [Xiaomi, arXiv'25]

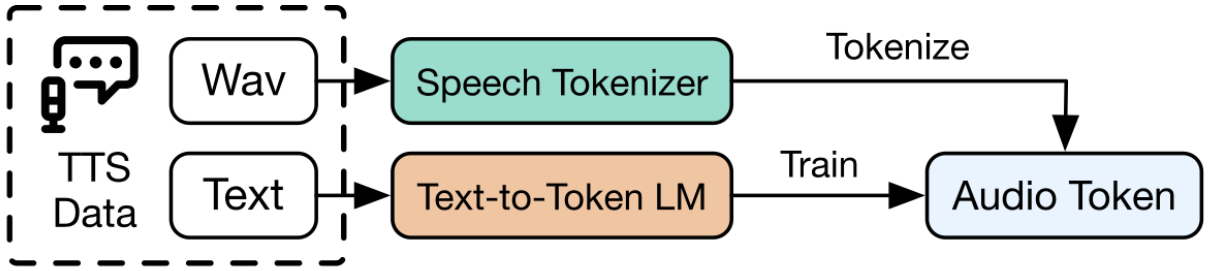
- 7Bモデルを**1億時間**の音声でInterleavingすると，0.8Tトークン付近で急速にFew-shot精度が向上する**相転移**が現れる
- 明示的に事前学習していない音声変換や音声翻訳もある程度解けるようになる
- オープンソースでありながら，GPT4oに匹敵する性能を示す



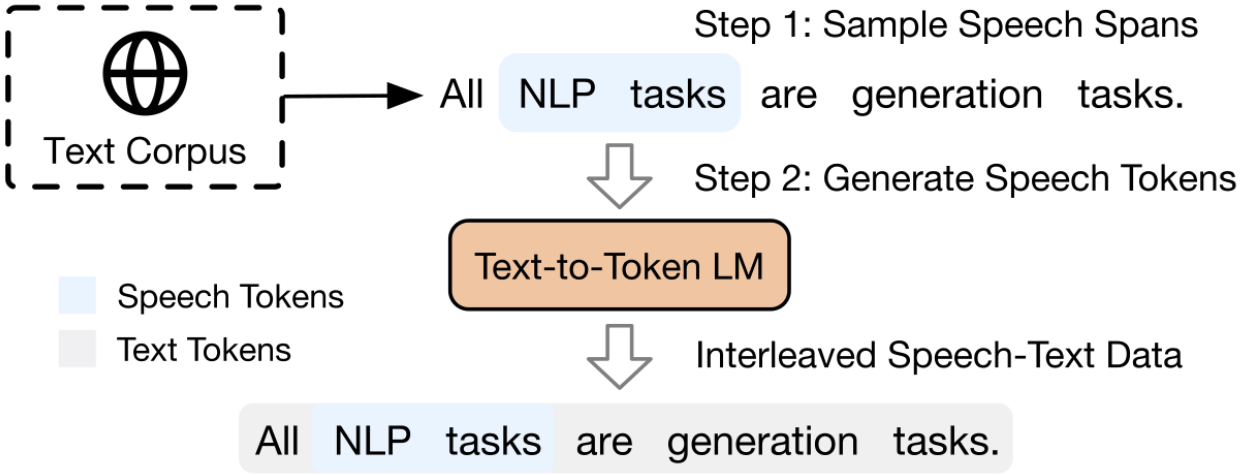


Interleavingデータの合成 [Zeng+, ICLR'25]

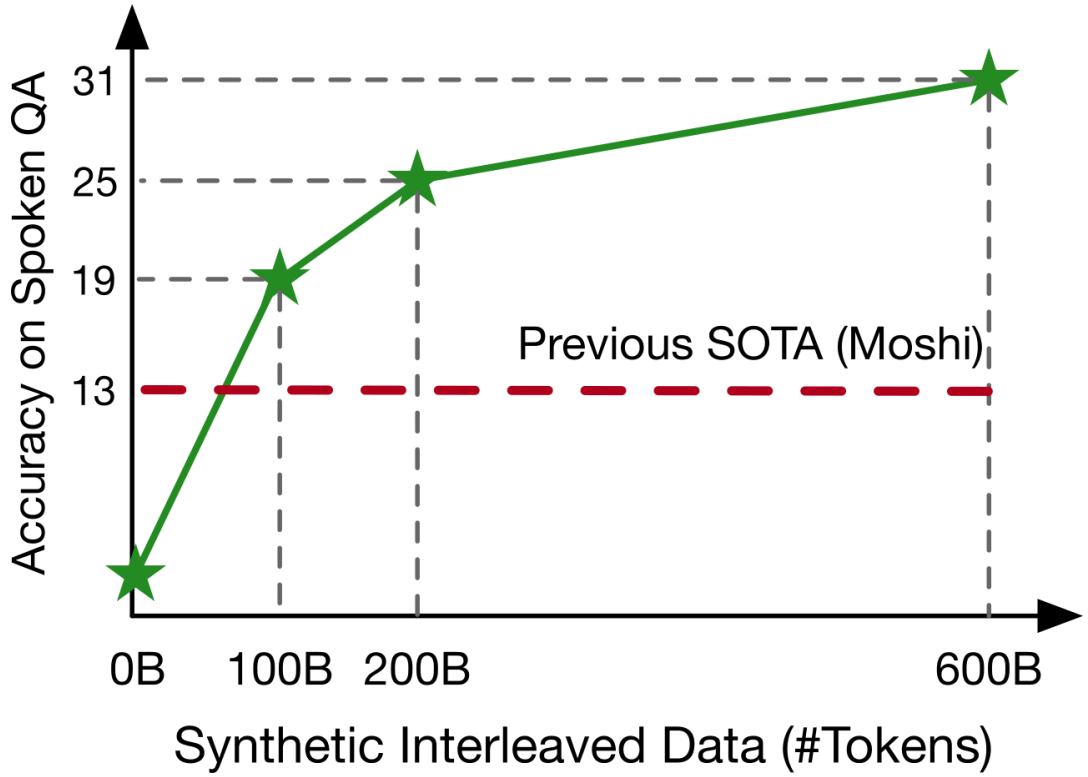
- 1億時間の音声を保存するには数PBのストレージが必要で，データ収集も大変
- TTSデータでText-to-Token LMを学習し，**テキストコーパスから直接Interleavedデータを生成**することで効率的にデータサイズをスケール



(a) Train a Text-to-Token Model using TTS data



(b) Construct Interleaved Speech-Text Data From Text Corpus



大規模学習を可能にする実装技術

単語レベルInterleavingのための強制アライメント

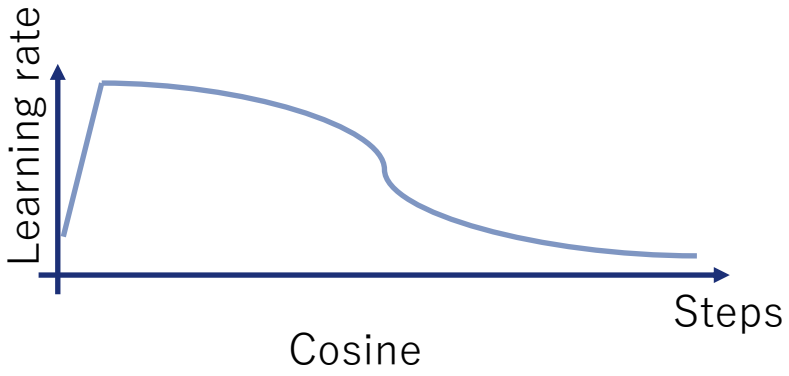
- 単語レベルでInterleavingするためには，各単語のタイムスタンプが必要
- 著名な強制アライメントツール
  - Montreal Forced Aligner (MFA) [McAuliffe+, Interspeech'17]
    - 正確だが，スケーラビリティに難あり
  - NeMo Forced Aligner (NFA) [Rastorgueva+, Interspeech'23]
  - WhisperX [Bain+, Interspeech'23]
  - MMS [Pratap+, JMLR'24]
  - Qwen3-ForcedAligner [Shi+, arXiv'26]
    - 日本語を含む多言語対応

	Monotonic-Aligner	NFA	WhisperX	Qwen3-ForcedAligner-0.6B
MFA-Labeled Raw				
Chinese	161.1	109.8	-	**33.1**
English	-	107.5	92.1	**37.5**
French	-	100.7	145.3	**41.7**
German	-	122.7	165.1	**46.5**
Italian	-	142.7	155.5	**75.5**
Japanese	-	-	-	**42.4**
Korean	-	-	-	**37.2**
Portuguese	-	-	-	**38.4**
Russian	-	200.7	-	**40.2**
Spanish	-	124.7	108.0	**36.8**
Avg.	161.1	129.8	133.2	**42.9**

[Shi+, arXiv'26]

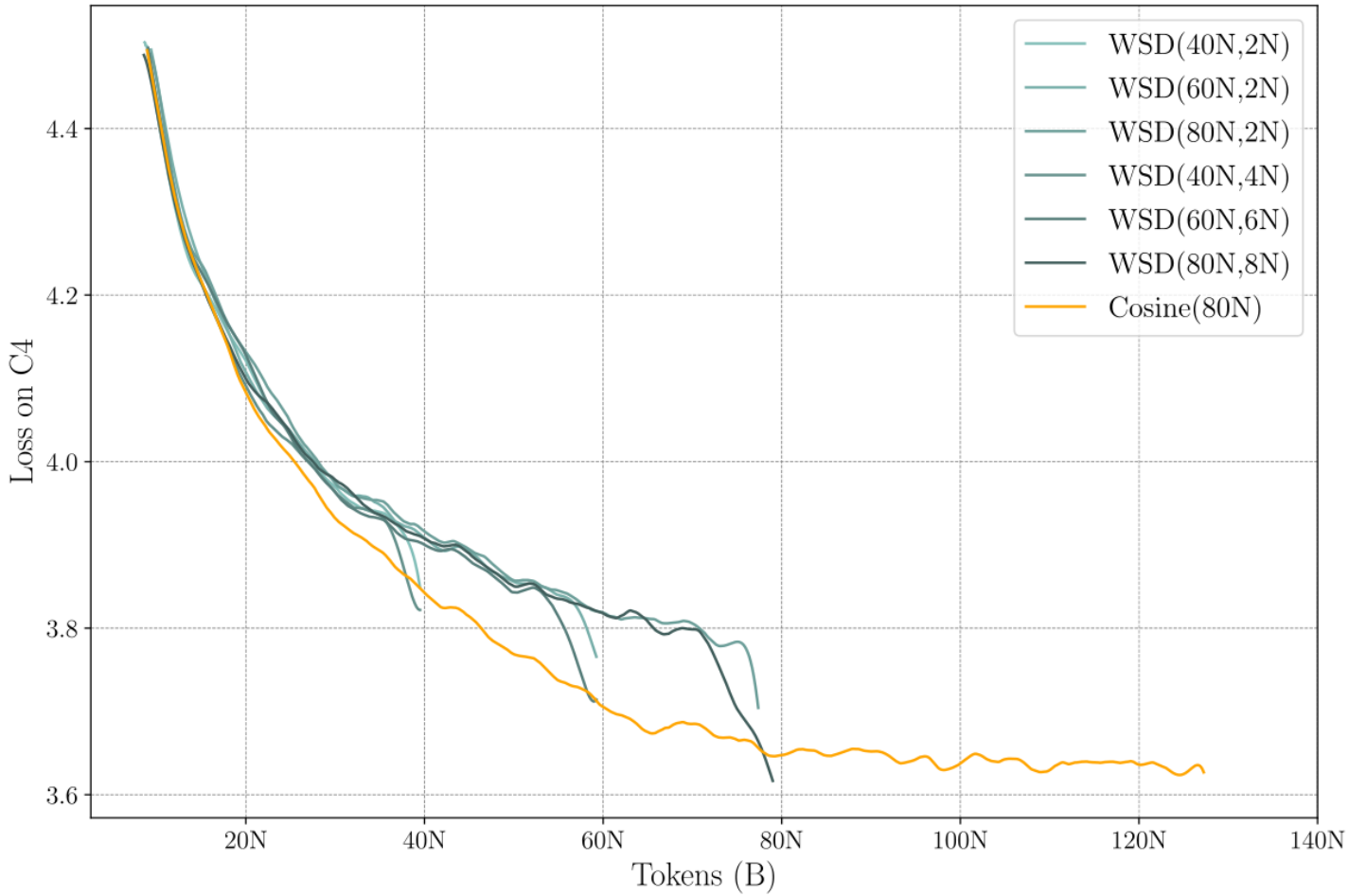
Warmup Stable Decay（WSD）学習率スケジューラ [Hu+, COLM'24]

- 大規模学習では，資金面から，学習率とステップ数の調整なしに1回の実験で済ませたい
- WSDでは，Stableフェーズでは学習率を一定に保ち，Lossがサチってきたら学習率を減衰することで，Cosineスケジューラと同等レベルまでLossが急激に減少
- 経験的には，減衰フェーズのステップ数は全体の10%で十分



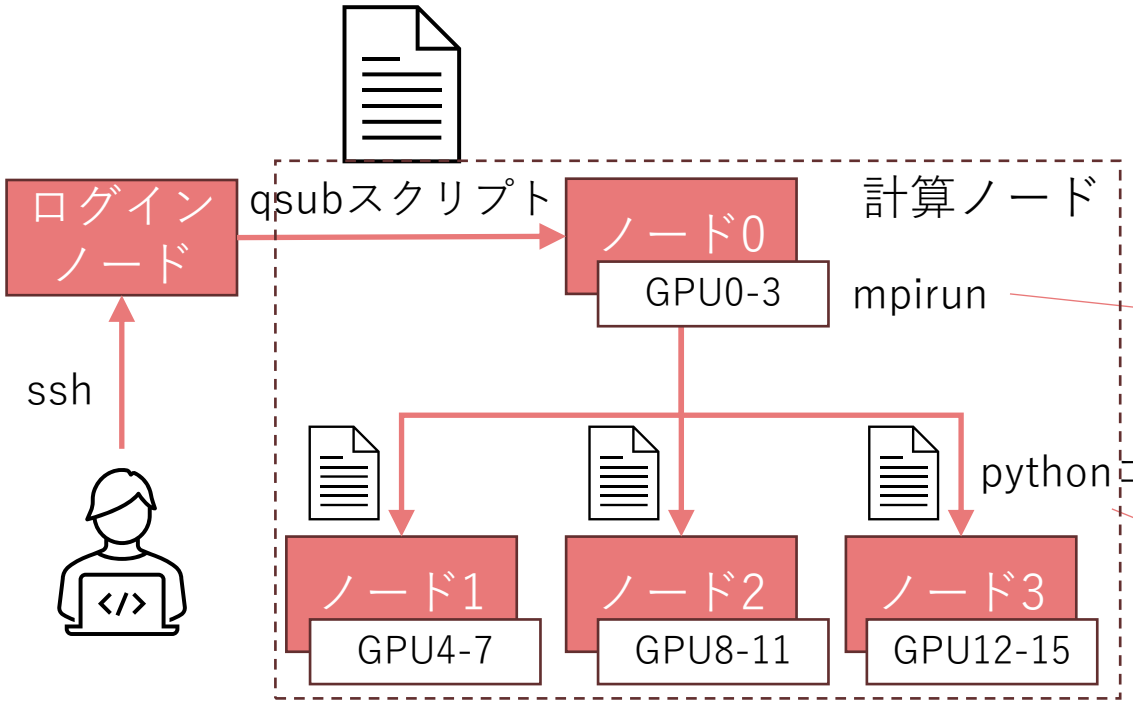
Cosine

Warmup Stable Decay (trapozoidal)



マルチノード分散学習

- 7BクラスのLLMを学習するためには産総研ABCIをはじめとするスーパーコンピュータが必要
- 学習時間を短縮するには，マルチノード学習が有効
- qsubでマスターノード（ノード0）へジョブ投入
- マスターノードでのmpirun経由で，各ノードでpythonコードを実行



4ノード16GPUの場合

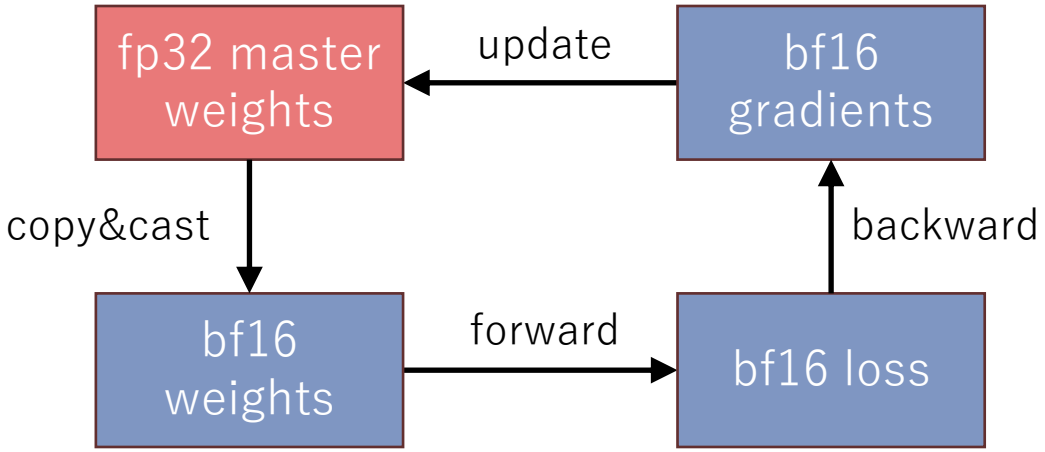
```bash
#!/bin/bash

#$ -l node_f=4      ## Number of node
#$ -l h_rt=24:00:00 ## Running job time

module load openmpi/5.0.7-nvhpc
module load cudnn/9.0.0
module load nccl/2.20.5
module load miniconda

MASTER_ADDR=$(head -n 1 $PE_HOSTFILE | awk '{print $1}')
awk '{print $1 " slots=4"}' "$PE_HOSTFILE" > hostfile

export MASTER_ADDR=$MASTER_ADDR
export MASTER_PORT=29500

mpirun \
    --hostfile hostfile \
    -npernode 4 \
    -n 16 \
    --bind-to none \
    -x MASTER_ADDR=$MASTER_ADDR \
    -x MASTER_PORT=$MASTER_PORT \
    bash -c '
        conda activate py310
        python train.py
    '
```

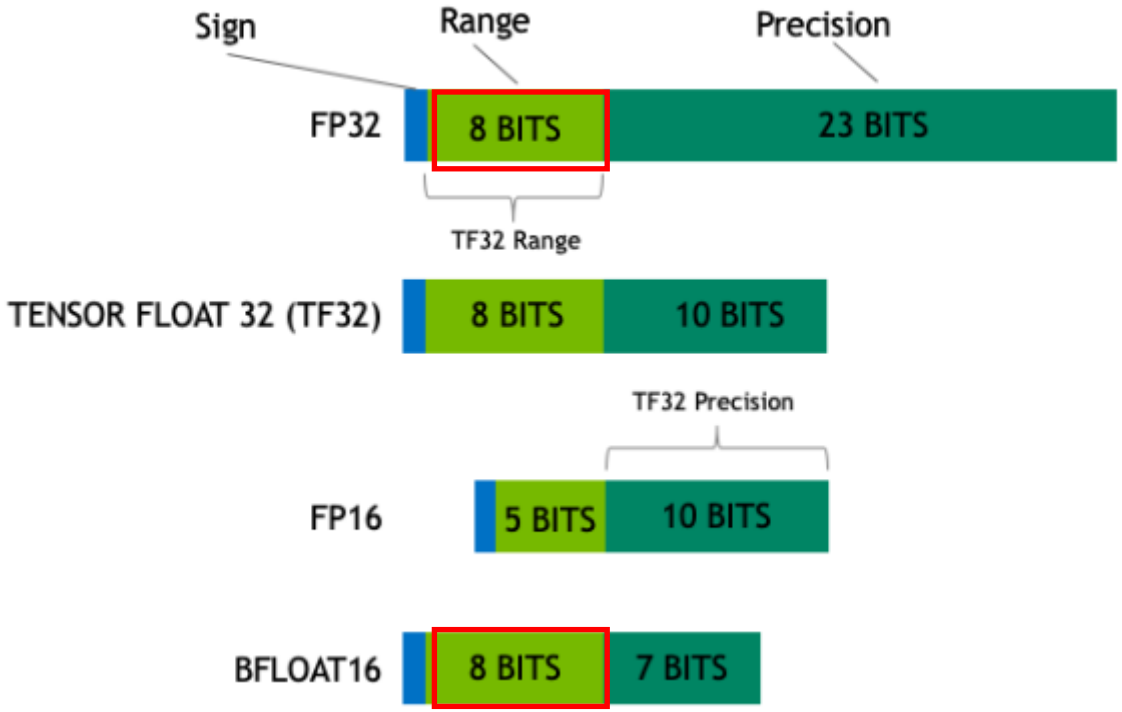
qsubスクリプト例（通常のshell script）

https://github.com/ryota-komatsu/speaker_disentangled_hubert/blob/main/scripts/run_speechlm_deepspeed.bash

Automatic Mixed Precision (torch.amp)

- **Forward/Backward計算をbfloat16で実行**することでメモリ削減
- モデルの**マスターパラメータをfp32で保持**することで，重み更新時の数値安定生保確保
- Tensor Core搭載のNVIDIA GPUの場合，bfloat16演算がハードウェア最適化されており，**学習も高速化**
- Bfloat16はfp16より精度は低いが，指数部がfp32と同じであり，オーバーフローが起こりにくいため，LLMで標準となっている



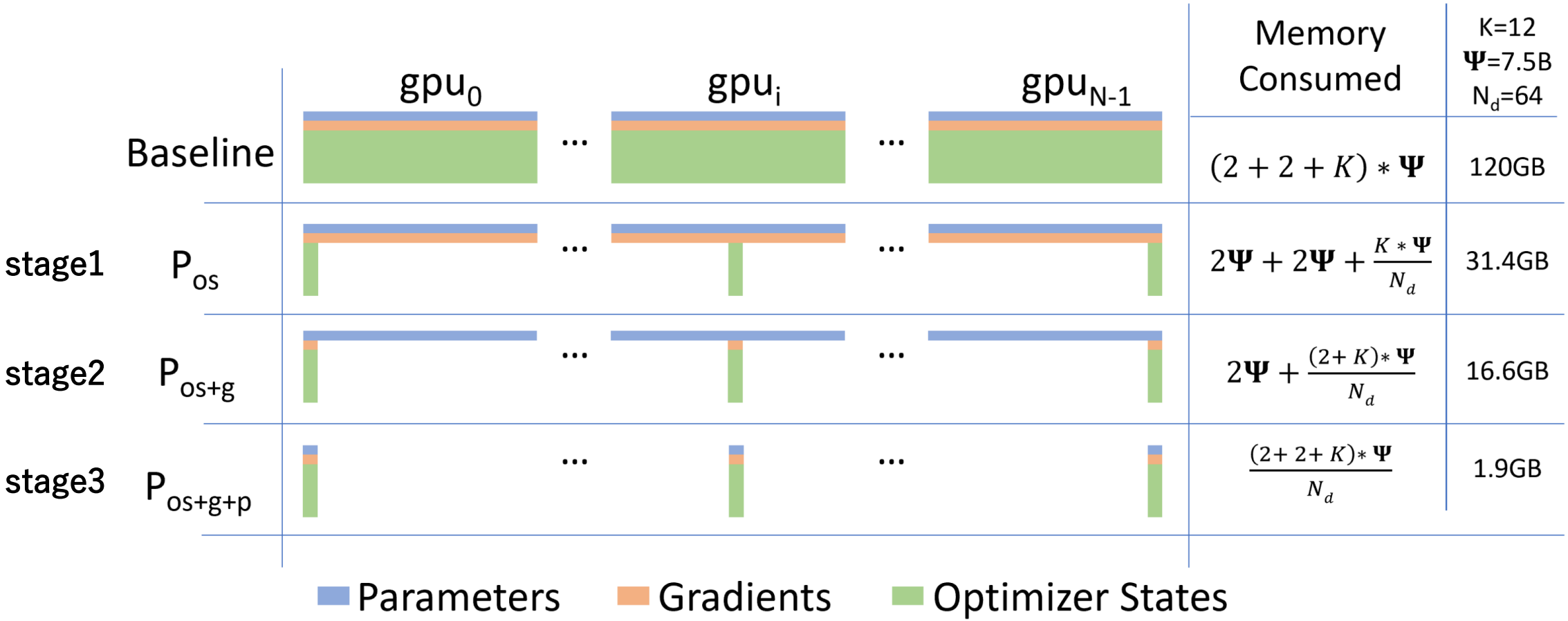


https://docs.nvidia.com/deeplearning/tensorrt/latest/inference-library/accuracy-considerations.html

DeepSpeed ZeRO [Rajbhandari+, SC'20]

Institute of SCIENCE TOKYO

- Adamオプティマイザでは勾配の平均と勾配の2乗の平均を状態として保持
- 7.5Bモデルの場合，モデルパラメータ+勾配+オプティマイザ状態で各GPUにつき最低120GBのVRAMが必要
- それらを**複数GPU間で重複のないshardへ分割**することで，必要メモリをGPU数で割ることができる
- 分割対象を3段階で選択可能



Transformers

- Automatic mixed precisionとDeepSpeed ZeROはtransformers.Trainerの設定のみで簡単に利用可能

```python
import deepspeed
from datasets import load_dataset
from transformers import AutoModelForCausalLM, AutoTokenizer, Trainer, TrainingArguments

def train():
    deepspeed.init_distributed()

    model = AutoModelForCausalLM.from_pretrained("Qwen/Qwen2.5-7B")
    tokenizer = AutoTokenizer.from_pretrained("Qwen/Qwen2.5-7B")
    train_dataset = load_dataset(
        "HuggingFaceTB/smollm-corpus", "fineweb-edu-dedup", split="train"
    )

    training_args = TrainingArguments(
        bf16=True,
        deepspeed="ds_zero2_bf16.json",
    )

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=train_dataset,
        processing_class=tokenizer,
    )
    trainer.train()
```

```json
{
    "bf16": {
        "enabled": "auto"
    },
    "zero_optimization": {
        "stage": 2,
        "allgather_partitions": true,
        "overlap_comm": true,
        "reduce_scatter": true,
        "contiguous_gradients": true
    },
}
```

https://github.com/ryota-komatsu/speaker_disentangled_hubert/blob/main/src/speechlm/train.py
https://github.com/ryota-komatsu/speaker_disentangled_hubert/blob/main/configs/speechlm/ds_zero2_bf16.json

まとめ

- End-to-End音声言語モデルの事前学習を効率的にスケールさせるには，Speech-text interleavingが有効

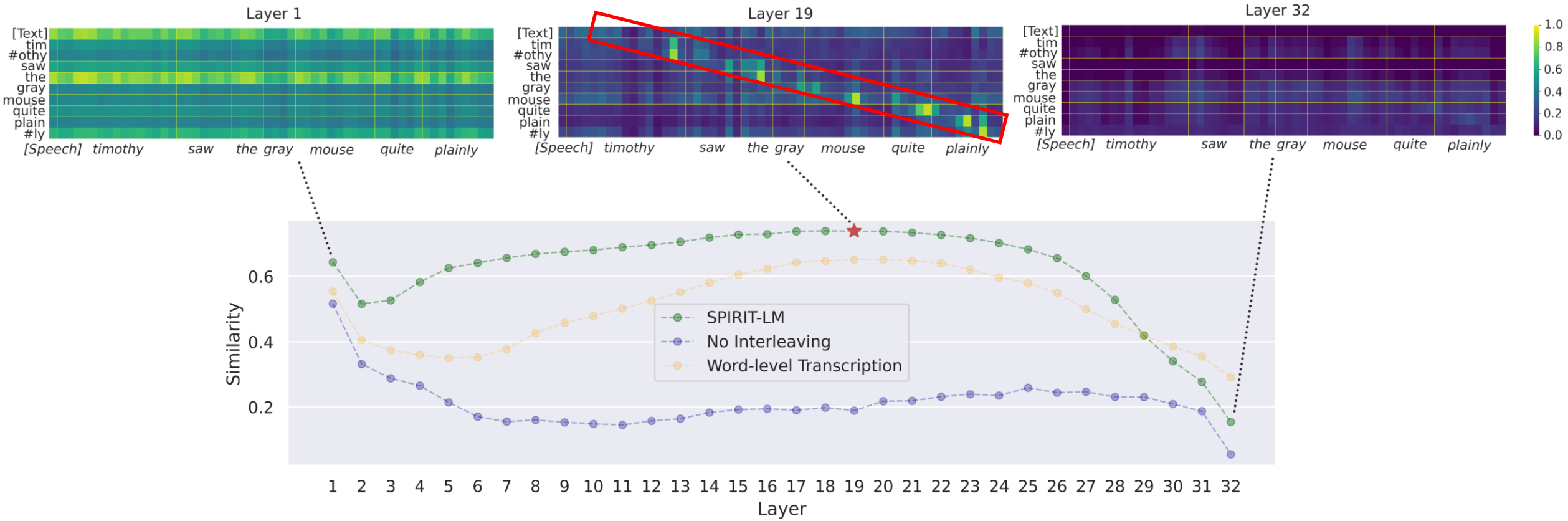
- 今後の課題
  - Interleavingによる知識転移の**仕組みの解明**
    - Interleavingモデルの音声言語理解性能とベースLLMのテキスト理解性能が相関するとは限らない

  - 音声言語モデルのための**Webデータ収集パイプライン**の作成
    - YODAS [Li+, ASRU'23]やEmilia [He+, TASLP'25]など数十万時間規模のASR・TTSデータセットが整備されつつあるが，書き起こし誤り，code-switching，不適切な単語，言い直し，フィラーなどが含まれている
    - 発話長も10秒未満であることが多く，複数文にまたがる意味理解には短いのではないか？

今後の課題：音声・テキスト間での表現分離

Layer-wise speech-text similarity [Nguyen+, TACL'25]

- Interleavingによって，対応する音声とテキスト表現のコサイン類似度が大きくなる（知識転移）
- **出力層付近では，類似度が大幅に低下**



Modality-based layer split [Zhao+, ICLR'26]

- **最終4層を各モダリティで分離**することで，双方のモダリティにおいて精度向上
- ウォームアップとして，**テキストで事前学習されたモジュールを凍結**して，ランダム初期化する音声モジュールのみ更新した後，全体を学習すると破滅的忘却を軽減可能

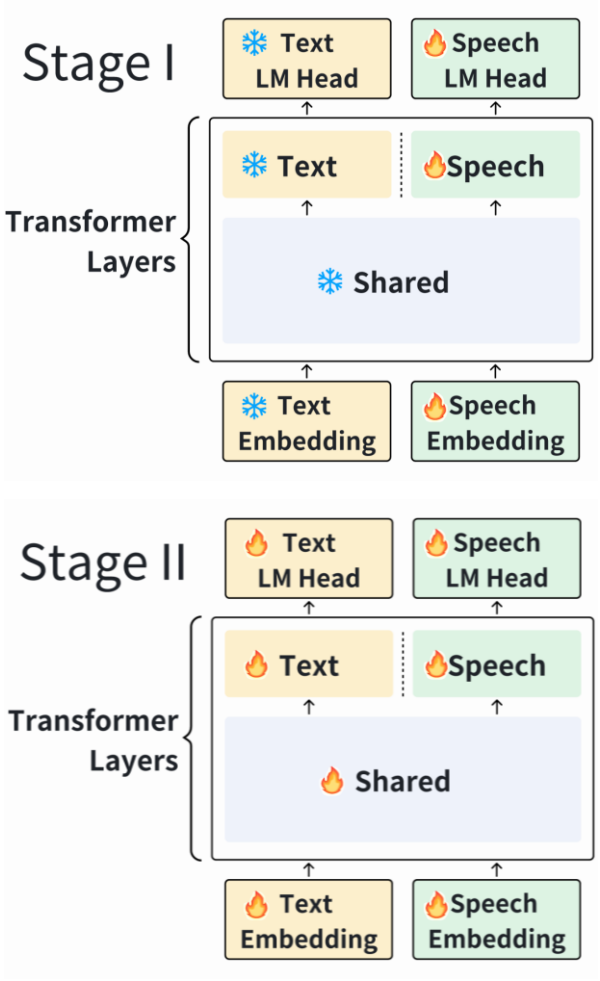
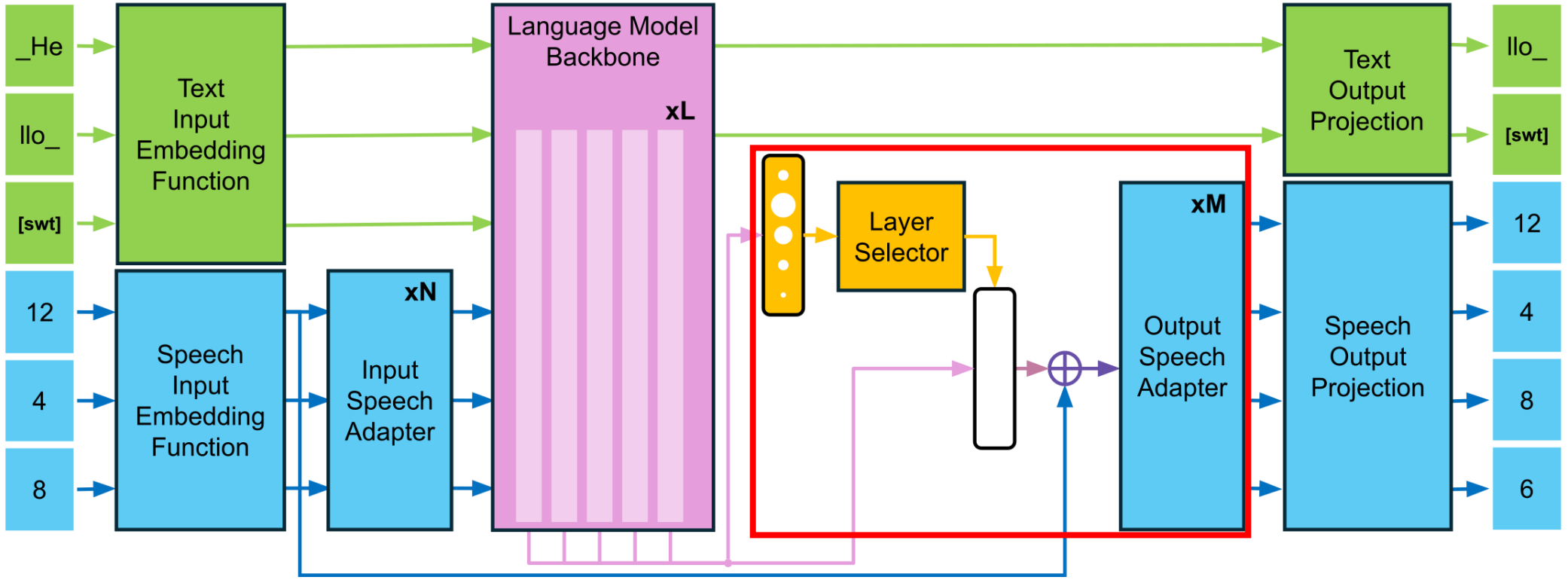


Table 6: **Ablation study on pre-training strategy.** FP: Frozen Pretrain (text parameters frozen during pretrain). **FP–Full**: all parameters unfrozen after Frozen Pretrain. **FP–Layerwise**: shared layers gradually unfrozen from last to first. **FP–Shared**: only speech–text shared layers unfrozen, text-specific remain frozen. **NF**: No Frozen Pretrain (all parameters trained directly). **NF–NoSplit**: NF without *Modality-Based Layer Split*, i.e., speech tokens added directly into text vocab without modality-specific layers. All models are trained for around 2 epochs on the pre-training dataset.

Model	Split Layers	Speech				Text	
		tS.C.	sS.C.	zh-tS.C.	zh-sS.C.	MMLU	CMMLU
FP–Full	4	**85.20**	63.12	**90.21**	72.10	66.50	69.15
FP–Layerwise	4	84.77	62.64	90.11	71.51	**68.82**	69.26
FP–Shared	4	83.27	**63.50**	90.11	**72.69**	67.26	**69.27**
NF	8	78.73	56.33	88.88	69.16	62.92	63.84
NF	6	79.05	56.49	89.10	67.93	63.27	63.79
NF	4	77.66	56.60	88.51	67.56	62.11	64.11
NF	2	78.09	56.87	88.62	68.31	62.92	63.74
NF–NoSplit	0	77.12	55.80	88.72	67.02	60.97	63.73
Qwen3-8B	-	-	-	-	-	76.60	77.35

Multi-Level Fission [Cuervo+, arXiv'25]

- **仮説1**：粗いサブワードテキスト予測に最適化された最終層で**細かい音素トークンを予測するには分解機構が必要**
  - ➢ 出力音声アダプタで表現を分解
- **仮説2**：音素はテキストより時間解像度が高いため，次単語予測のための**最終層表現が必要になる頻度は稀**
  - 単語境界では文法など言語的に抽象度の高い情報が必要であるが，単語内では抽象度の低いスペル予測で済む
  - ➢ 各層の隠れ状態を重み付けして次トークン予測することで，**音声トークンごとに異なる階層に着目**



Multi-Level Fission [Cuervo+, arXiv'25]

- 実際に，単語境界で最終層の重みが大きい傾向