



FOUNDATIONAL SPEECH MODELS AND THEIR EFFICIENT TRAINING WITH NVIDIA NEMO

Piotr Żelasko, Jan 23rd, 2025, Mila Conversational AI Reading Group





BACKGROUND

CONTRIBUTORS TO THIS WORK

Boris Ginsburg

Piotr Żelasko Subhankar Ghosh
Yifan Peng Somshubra Majumdar

Kevin Hu Krishna Puvvada

Vahid Noroozi Elena Rastorgueva

Daniel Galvez Nithin Rao Koluguri Mengru Wang

Vitaly Lavrukhin Jagadeesh Balam

Steve Huang Andrei Andrusenko Zhehuai Chen

Oleksii Hrinchuk

Vladimir Bataev Kunal Dhawan

Shuoyang Ding

NVIDIA NEMO ASR MODELS



The 😊 Open ASR Leaderboard ranks and evaluates speech recognition models on the Hugging Face Hub.

We report the Average [WER](#) (⬇️ lower the better) and [RTFx](#) (⬆️ higher the better). Models are ranked based on their Average WER, from lowest to highest. Check the ✎ Metrics tab to understand how the models are evaluated.

If you want results for a model that is not listed here, you can submit a request for it to be included 📩✨.

The leaderboard currently focuses on English speech recognition, and will be expanded to multilingual evaluation in later versions.

[Leaderboard](#) [Metrics](#) [✉️✨ Request a model here!](#)

model	Average WER ⬇️	RTFx ⬆️	AMI	Earnings22	Gigaspeech	LS Clean	LS Other
nvidia/canary-1b	6.5	235.34	13.9	12.19	10.12	1.48	2.93
nyrahealth/CrisperWhisper	6.67	84.05	8.71	12.89	10.24	1.82	4
nvidia/parakeet-tdt-1.1b	7.01	2390.61	15.87	14.49	9.52	1.4	2.6
nvidia/parakeet-rnnt-1.1b	7.12	2053.15	17.01	13.94	9.89	1.45	2.5
nvidia/parakeet-ctc-1.1b	7.4	2728.52	15.67	13.75	10.28	1.83	3.51
openai/wisper-large-v3	7.44	145.51	15.95	11.29	10.02	2.01	3.91

ACCELERATING LEADERBOARD-TOPPING ASR MODELS 10X WITH NVIDIA NEMO

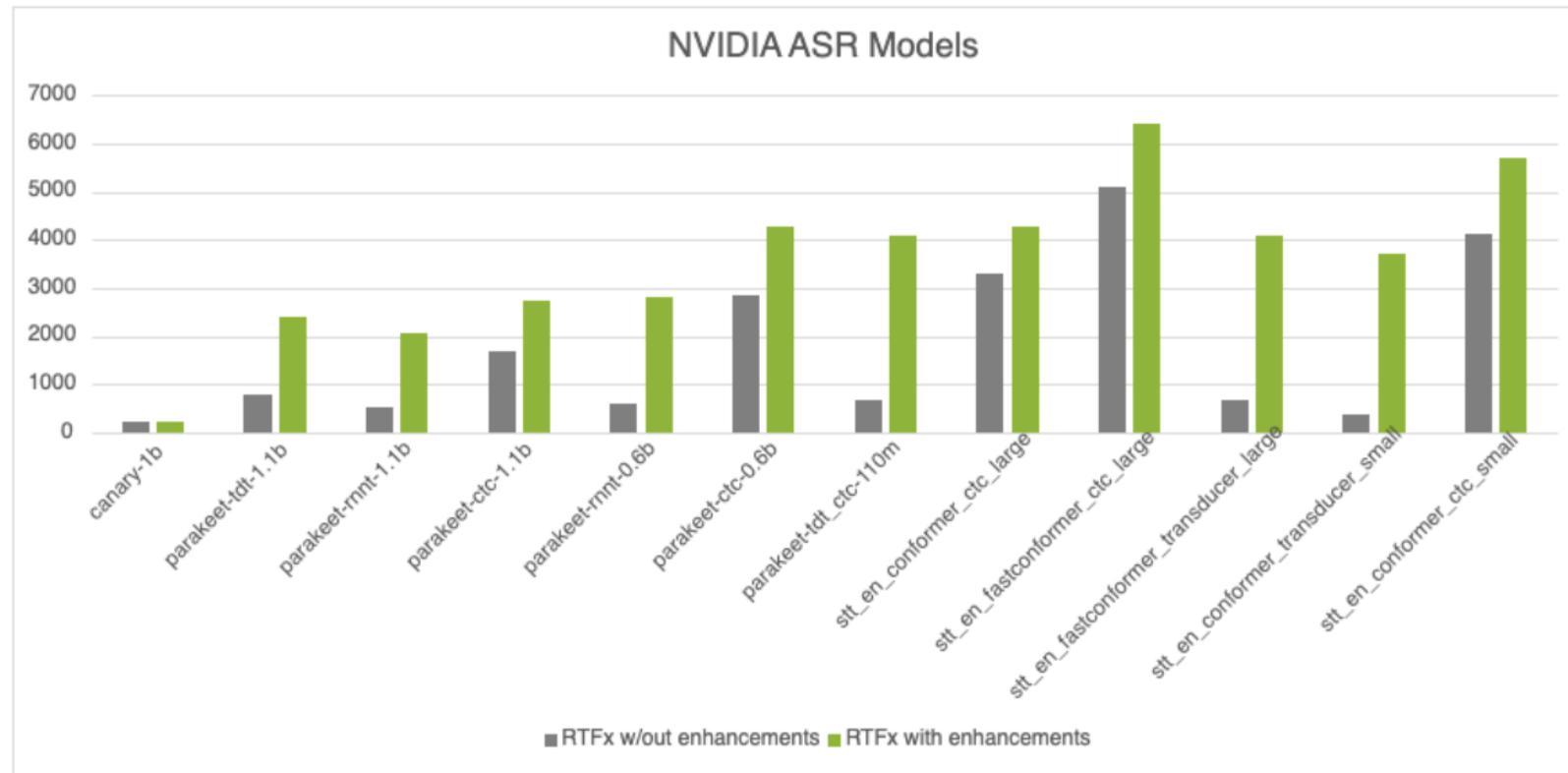


Figure 1. NVIDIA NeMo ASR models achieve up to 10x speed improvement in inverse real-time factor (RTFx) through recent optimizations

THIS TALK AT A GLANCE

Canary-1B

Training
efficiency

Multimodal
Speech LLM

THIS TALK AT A GLANCE

Canary-1B

Foundation Speech Models

Training
efficiency

Multimodal
Speech LLM

Data Sampling & Curation



CANARY-1B

Overview

Features

- ✓ The first multilingual and multitask NeMo model.
 - Speech Recognition and Translation
 - English, Spanish, German and French
- ✓ Supports Punctuation and Capitalization.

Training efficiency

- ✓ On-the fly manifest loading and tokenization.
 - >75% reduction in training start time.
- ✓ Dynamic bucketing and dataset mixing.
- ✓ Canary-1B model trains under 48h with 128 GPUs.

Metrics

- ✓ Outperforms Seamless-V1 (1.3B) on languages tested in both ASR and AST.
- ✓ Outperforms all Seamless and Whisper models for non-En languages in ASR.
- ✓ Comparable with Whisper-large-V3 for En.
- ✓ Achieves 1st place (WER) on HuggingFace Open ASR leaderboard.

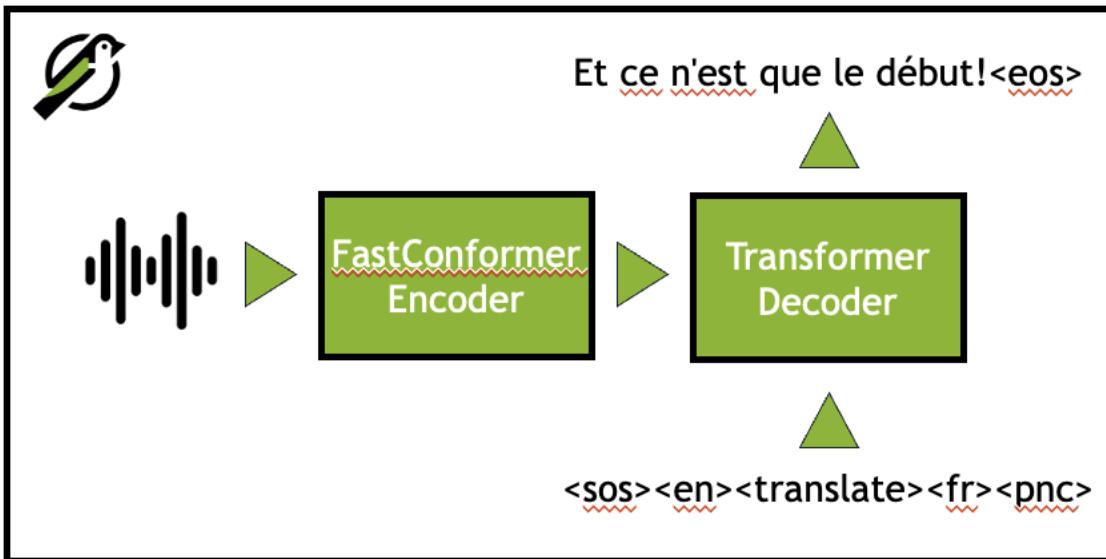
Data

- Seamless V1: 406k hours (labeled + pseudo-labeled)
- Seamless V2: 520k hours
- Whisper V3: 1M hours weakly labeled + 4M hours pseudo-labeled
- Canary: 86k hours of labeled data, replicated x3 with synthetic translations.

Architecture

Fast-Conformer Encoder Transformer Decoder

Canary-1B architecture diagram.



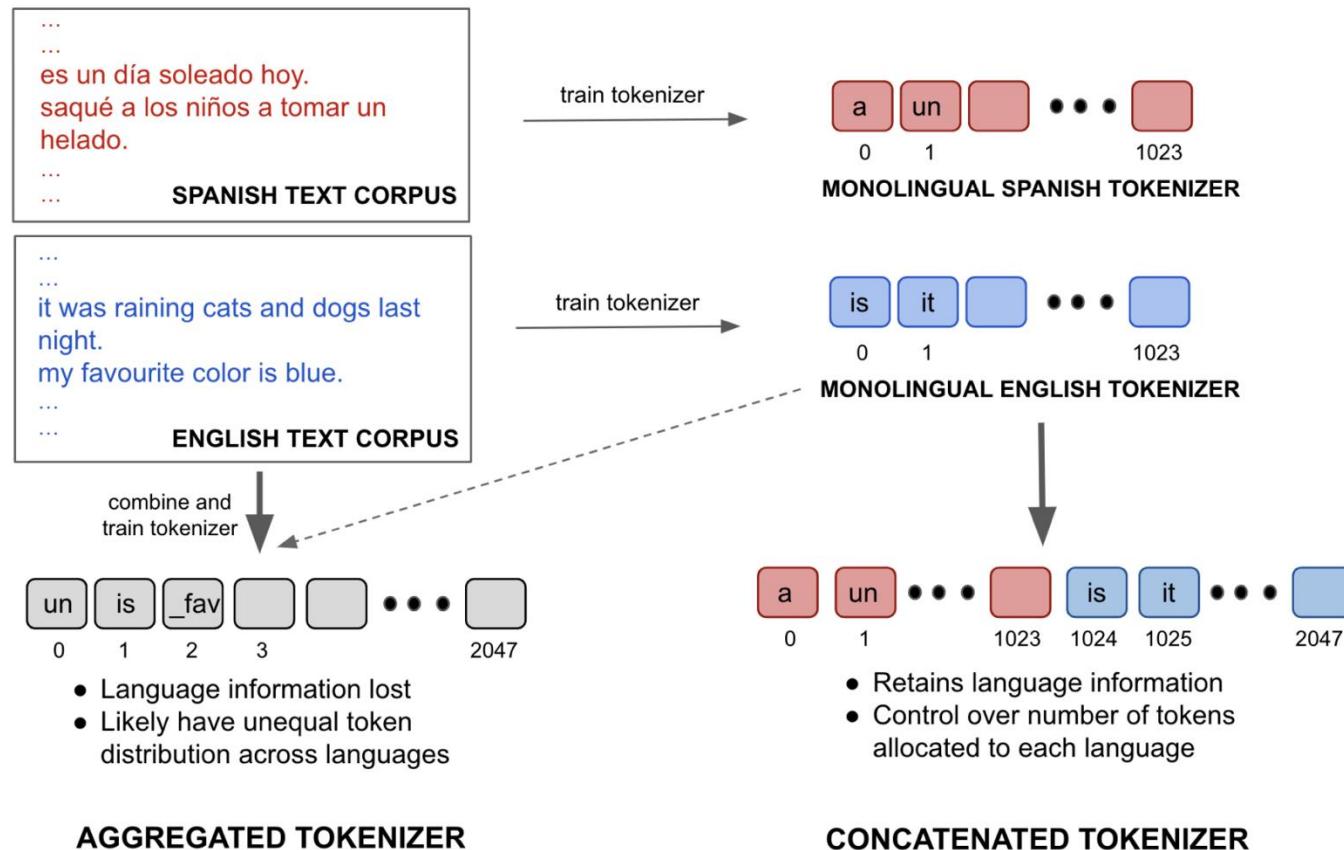
Key contributions

- Canary-1B, an open-source ASR and AST in English, French, Spanish, and German.
- Highly accurate speech translation using exclusively synthetic translation data in training.
- Fast training (<48 hours) using 128 NVIDIA A100 80GB with a total of 225K weight updates through:
 - Pre-trained encoder initialization
 - Dynamic bucketing / variable sequence lengths
 - Dynamic, weighted, and balanced data blending
- Reduced hallucination via training on noise data

Model architecture

- Attention Encoder-Decoder
- FastConformer-XL encoder: 24 layers, 0.6B parameters
- Transformer decoder: 24 layers, 0.4B parameters
- 128 log-Mel filter bank features
- Training code available in NVIDIA NeMo

Concatenated Tokenizer



Data

Table 1: Training data statistics with breakdown per language and availability of punctuation and capitalization (PnC).

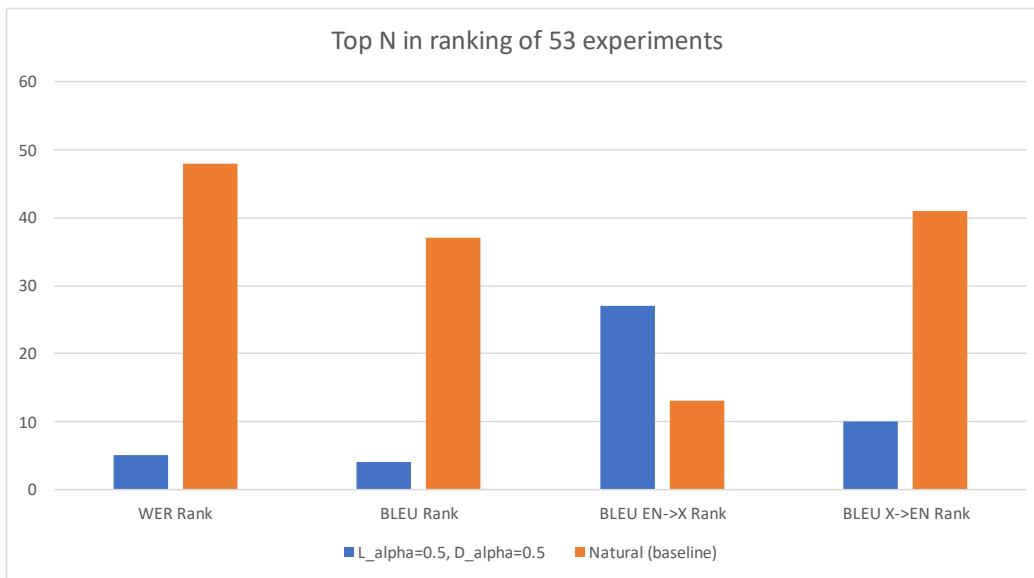
Language	Public + In-house [K hours]	PnC [K hours]	Dur. [s] [Min, Max]	# Utterances [M]
English	25.5 + 37.9	38.5	[1, 40]	24
German	2.5 + 3.6	6.1	[1, 20]	2.4
Spanish	1.4 + 5.2	1.4	[1, 20]	3.8
French	1.8 + 3.3	1.8	[0.5, 40]	2.5
Non-speech	0.3 + 0.0	NA	[0.47, 10]	0.1
Total	31.5 + 50	47.8	NA	32.8

- Data sampling and loading using Lhotse library
- Batch size: 360s/GPU, 31 buckets, quadratic duration penalty (20s)
- Synthetic translation generation with Megatron Multilingual NMT 500M
- **Data blending: two levels of temperature-based re-weighting**
 - Natural weight: the number of hours of speech
 - 1st level of re-weighting: aggregate by language, apply temperature
 - 2nd level of re-weighting: within each language, aggregate by dataset and apply temperature
 - Temperature alpha=0.5 for both levels worked best, where n_s/N are natural weights for each item/group

$$p_s \sim \left(\frac{n_s}{N} \right)^\alpha$$

Data Balancing via Temperature Re-weighting

- High-level idea: upsample data from under-represented categories (task, language, dataset)
- Method: obtain natural weights (number of hours) per category, and smooth the sampling distribution
- The approach yielding best results:
 - Effective dataset weight = language weight * inner (within language) dataset weight
 - Both language and inner dataset weight is smoothed using temperature, best alpha = 0.5 (out of 1.0, 0.75, 0.5)



Exp.	WER (%)	BLEU (avg)	BLEU EN->X	BLEU X->EN
L_alpha=0.5, D_alpha=0.5	10.41	28.88	28.87	28.88
Natural (baseline)	11.15	28.25	29.22	26.54

Tab. Averaged results from Canary data blending ablations.

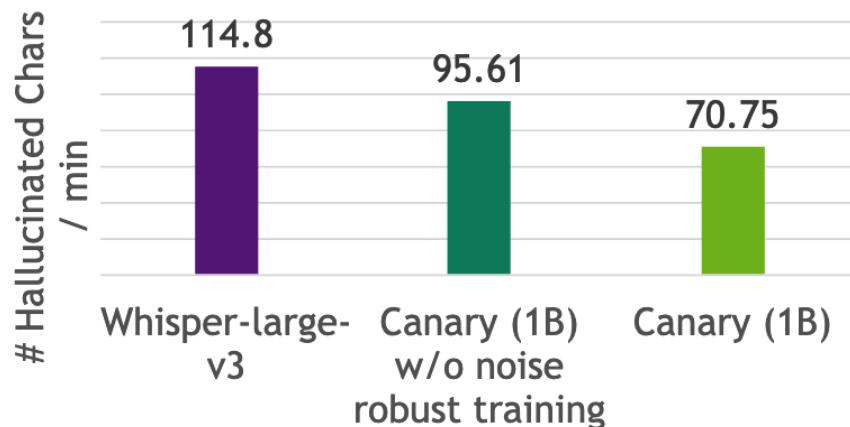
Fig. Ranking spots achieved by the baseline and the best overall experiment.

Results

Table 2. Canary on HuggingFace Open ASR Leaderboard.

Model (WER ↓)	AMI	Earnings22	GigaSpeech	LS Clean	LS Other	SPGISpeech	Tedlium	Avg. WER
Whisper-large-v3	16.01	11.3	10.02	2.03	3.91	2.95	3.9	7.16
Parakeet-RNNT-1.1B	17.1	15.15	9.96	1.46	2.48	3.11	3.92	7.60
Parakeet-TDT-1.1B	15.9	14.65	9.55	1.39	2.62	3.42	3.56	7.30
Canary (1B)	13.53	12.05	10.07	1.47	2.86	2.02	3.53	6.50

Figure 2. Hallucinated characters / minute on MUSAN.



Results

Table 3. WER (%) results on multilingual ASR benchmarks.

Model (WER ↓)	En			De			Es			Fr		
	MCV-16.1	MLS	VoxPopuli	MCV-16.1	MLS	VoxPopuli	MCV-16.1	MLS	VoxPopuli	MCV-16.1	MLS	VoxPopuli
OWSM-v3.1 (1.02B)	11.87	5.37	7.04	9.24	10.49	16.25	9.59	8.84	10.17	13.69	11.75	12.61
SeamlessM4T-medium (1.2B)	10.25	7.05	6.06	9.32	8.12	12.95	7.25	5.25	7.31	11.07	7.32	8.77
SeamlessM4T-large-v2 (2.3B)	7.47	4.14	4.68	5.82	6.08	10.68	4.82	4.14	6.76	7.75	5.38	6.82
Whisper-large-v3 (1.5B)	9.92	3.53	6.23	6.17	5.83	16.50	4.94	4.42	8.01	11.18	5.38	7.52
Canary (1B)	7.83	3.03	4.42	4.49	4.09	10.70	3.88	3.12	5.02	6.37	4.06	5.48

Table 4. BLEU scores on speech translation (AST) benchmarks.

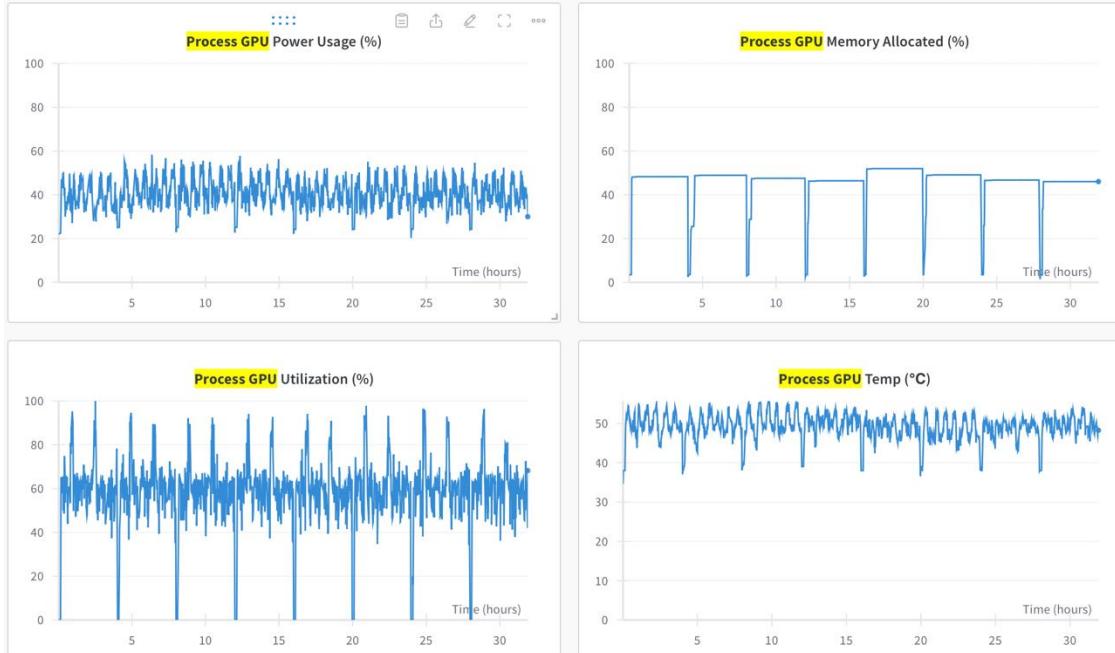
Model (BLEU ↑)	FLEURS (En → X)			mExpresso (En → X)			FLEURS (X → En)			COVOST-v2 (X → En)		
	En → De	En → Es	En → Fr	En → De	En → Es	En → Fr	De → En	Es → En	Fr → En	De → En	Es → En	Fr → En
OWSM-v3.1 (1.02B)	24.37	11.39	16.39	19.29	10.98	8.59	13.22	9.35	12.38	18.05	23.90	24.47
SeamlessM4T-medium (1.2B)	28.30	21.05	37.36	9.65	16.23	8.64	33.39	21.68	30.94	35.60	39.18	39.27
SeamlessM4T-large-v2 (2.3B)	33.17	23.72	43.05	21.48	34.89	26.04	37.06	25.41	33.70	39.96	42.91	42.12
Whisper-large-v3 (1.5B)	N/A	N/A	N/A	N/A	N/A	N/A	33.40	22.70	31.02	34.22	39.20	35.49
Canary (1B)	32.13	22.06	39.50	24.42	35.76	27.96	33.70	22.06	31.57	37.92	40.79	40.58



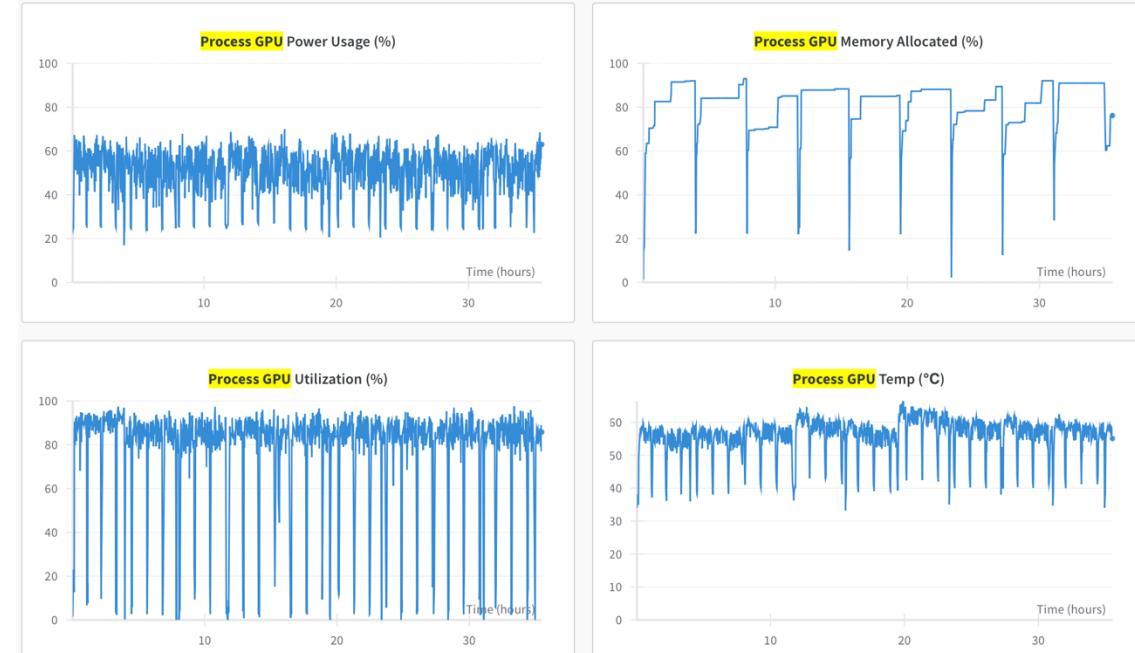
TRAINING EFFICIENCY

ARE WE AT THE “SPEED OF LIGHT”?

Training A



Training B



“SIMPLE” BOTTLENECKS

FEATURE NORM & SPECAUG: VECTORIZATION

Feature normalization

```
if normalize_type == "per_feature":  
    x_mean = torch.zeros((seq_len.shape[0], x.shape[1]), dtype=x.dtype, device=x.device)  
    x_std = torch.zeros((seq_len.shape[0], x.shape[1]), dtype=x.dtype, device=x.device)  
    for i in range(x.shape[0]):  
        if x[i, :, : seq_len[i]].shape[1] == 1:  
            raise ValueError(  
                "normalize_batch with `per_feature` normalize_type received a  
                tensor with 1 feature dimension. This is not supported.  
                In torch.std() returning nan. Make sure your audio length has  
                at least 2 features (ex. at least `hop_length` for Mel Spectrograms).")  
        x_mean[i, :] = x[i, :, : seq_len[i]].mean(dim=1)  
        x_std[i, :] = x[i, :, : seq_len[i]].std(dim=1)
```

Plain Python for loops
over tensor dimensions

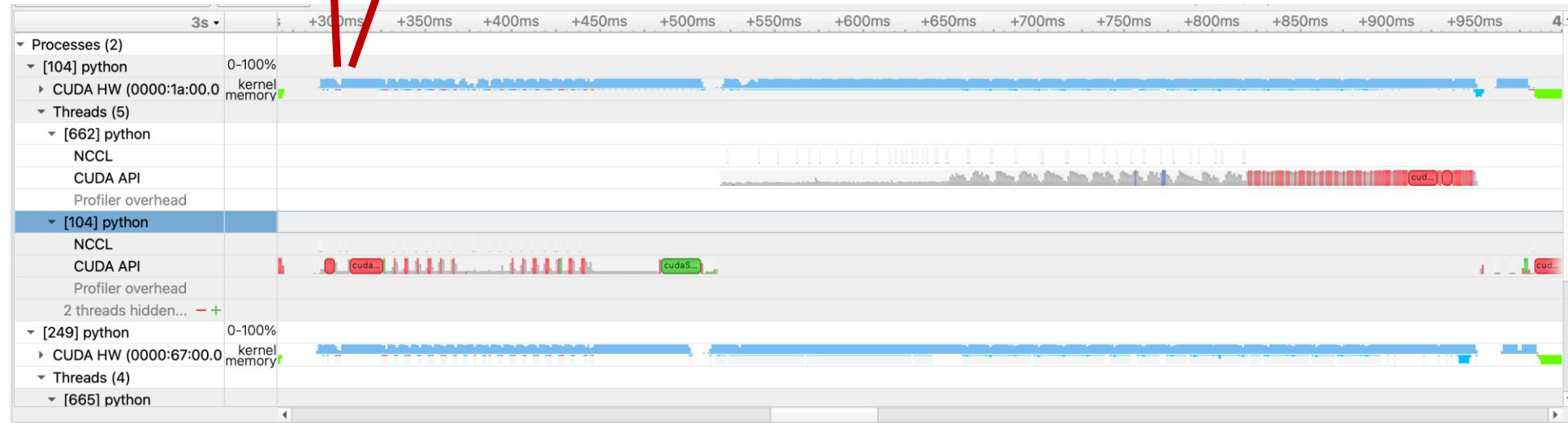
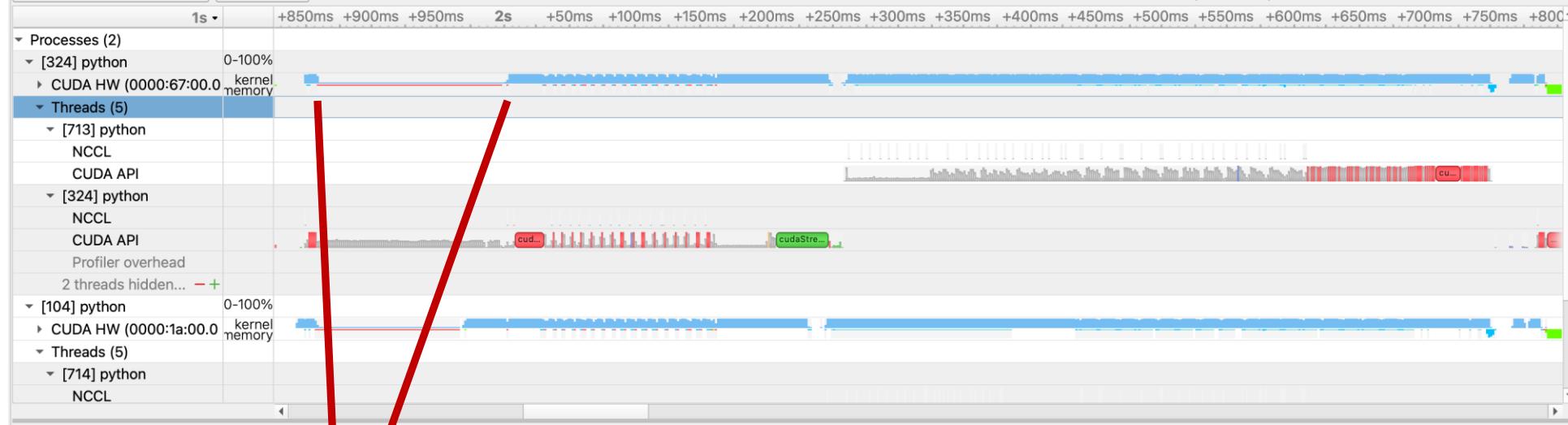
SpecAugment

```
for idx in range(batch_size):  
    # Set freq masking  
    for _ in range(self.freq_masks):  
        start = self._rng.randint(0, freq_start_upper_bound)  
        width = self._rng.randint(0, self.freq_width)  
        fill_mask[idx, start : start + width, :] = True  
  
    # Derive time width, sometimes based percentage of input length.  
    if self.adaptive_temporal_width:  
        time_max_width = max(1, int(lengths_cpu[idx] * self.time_width))  
    else:  
        time_max_width = self.time_width  
    time_start_upper_bound = max(1, lengths_cpu[idx] - time_max_width)  
  
    # Set time masking  
    for _ in range(self.time_masks):  
        start = self._rng.randint(0, time_start_upper_bound)  
        width = self._rng.randint(0, time_max_width)  
        fill_mask[idx, :, start : start + width] = True
```

PRs with enhancements (for reference):

<https://github.com/NVIDIA/NeMo/pull/8964>
<https://github.com/NVIDIA/NeMo/pull/9155>

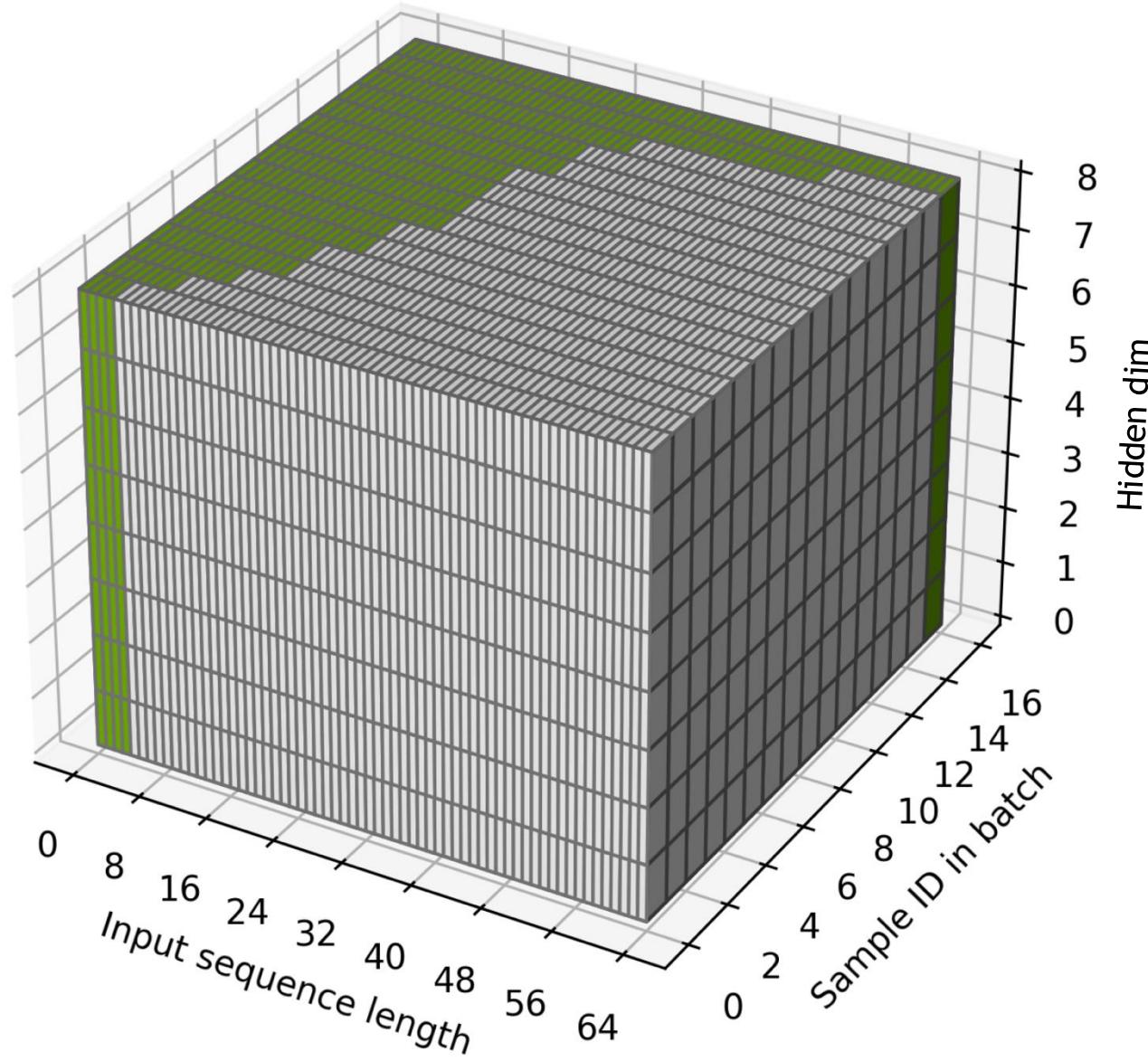
BEFORE AND AFTER (15% SPEEDUP)





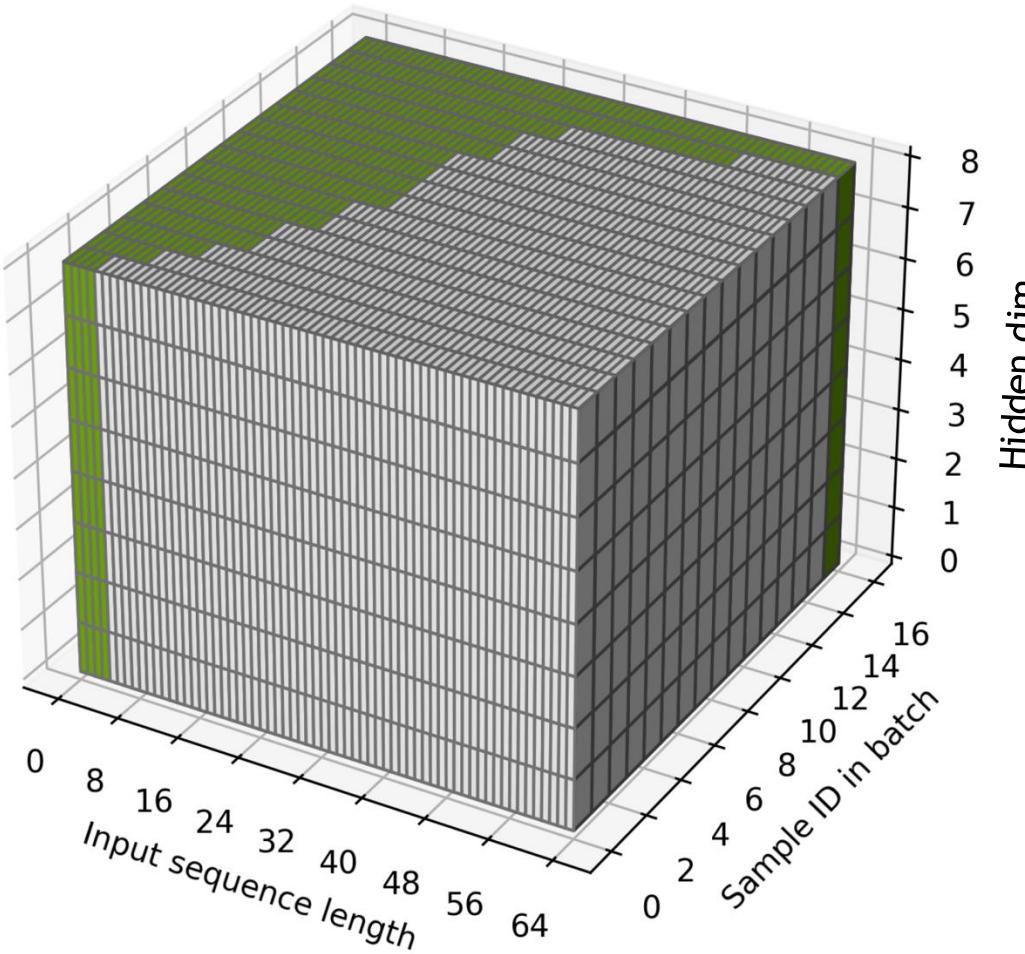
THE FUTURE IS SHAPED DYNAMIC

FIXED BATCH SIZE

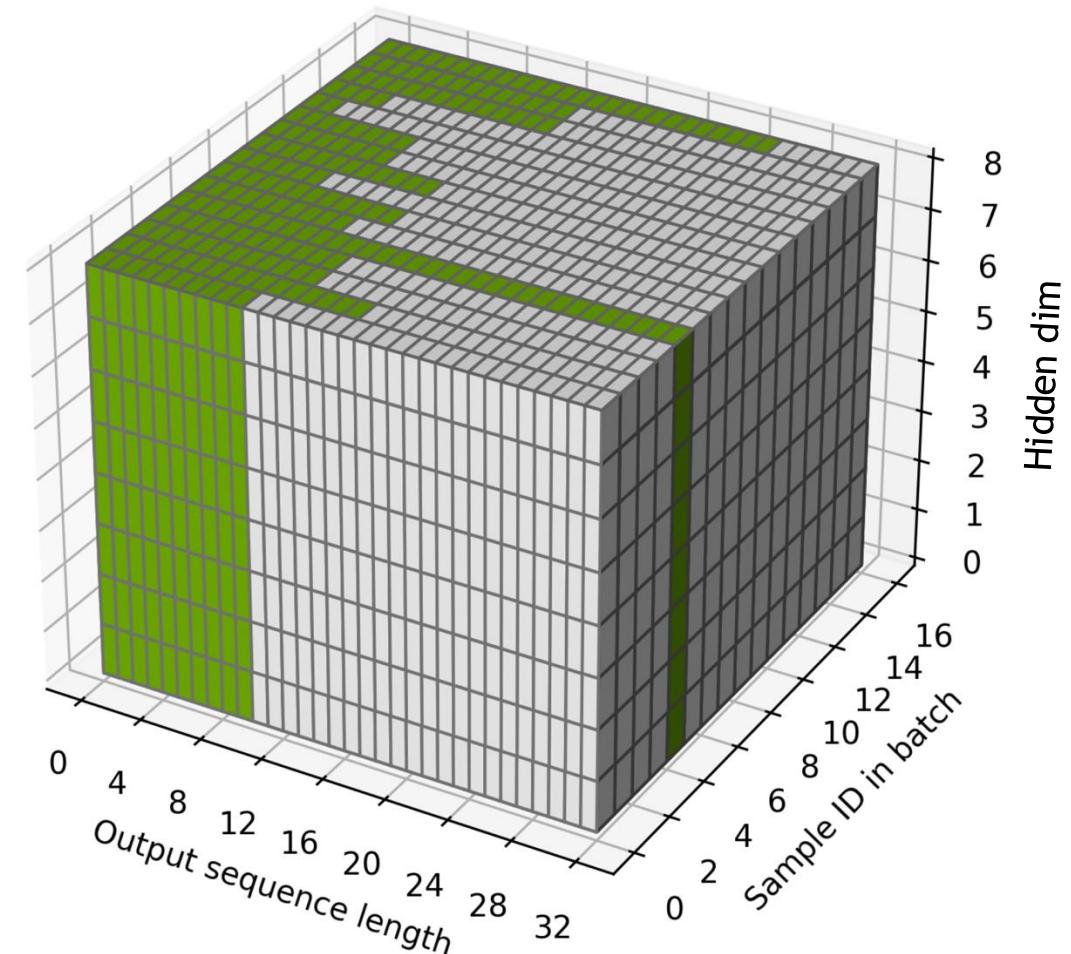


ENCODER-DECODER FIXED BATCH SIZE

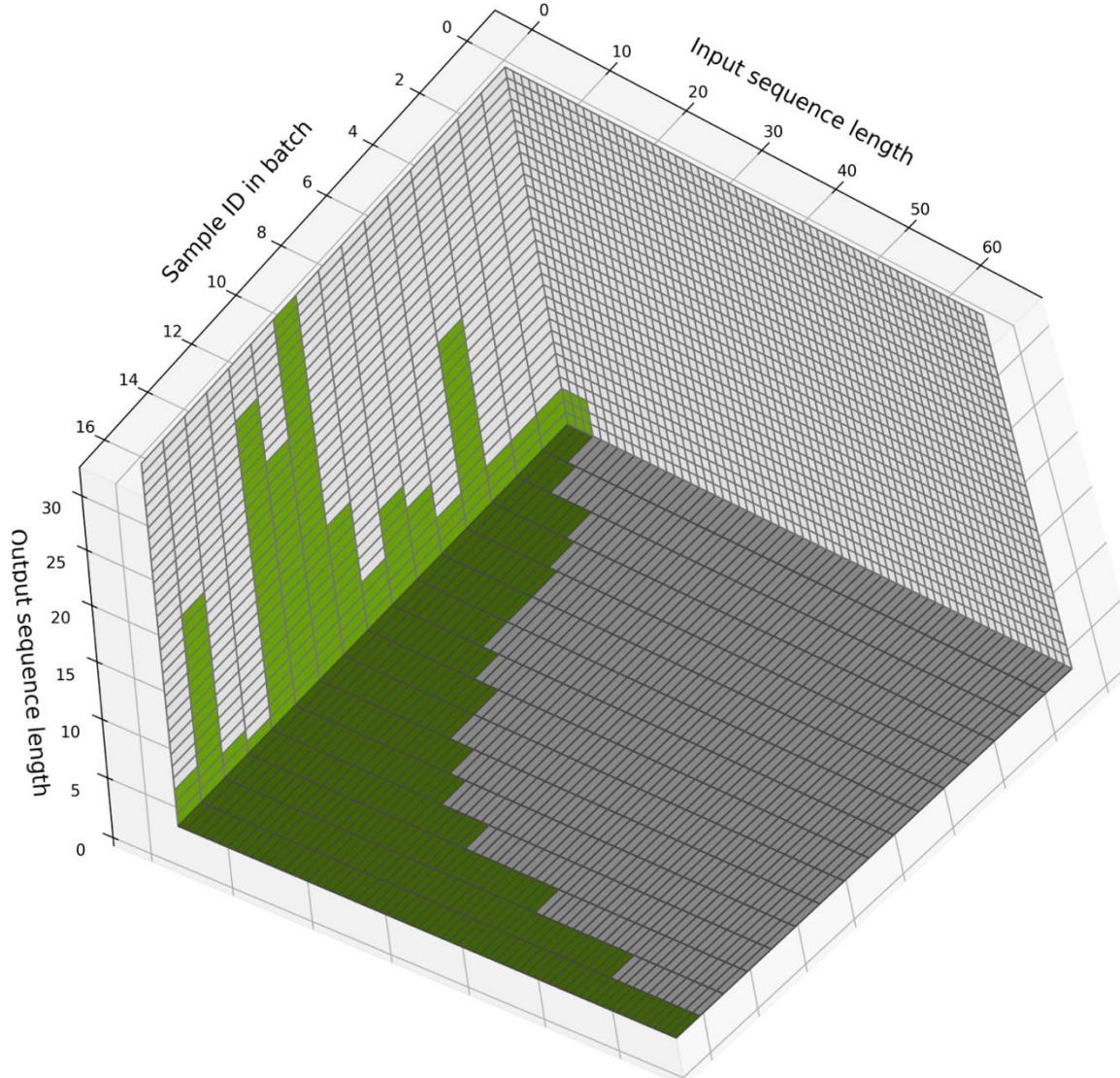
Input length distribution



Output length distribution



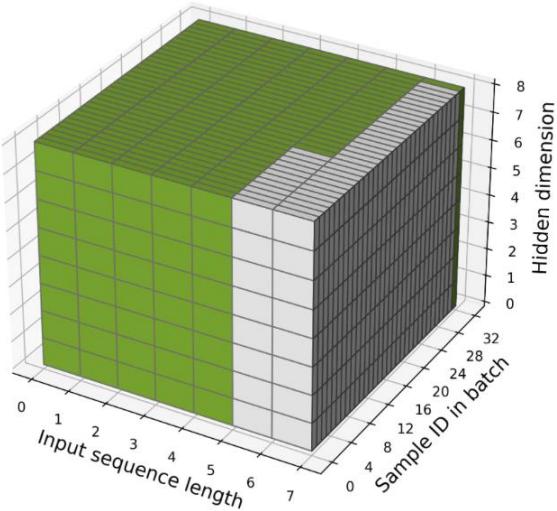
ENCODER-DECODER CROSS-ATTENTION



BUCKETING

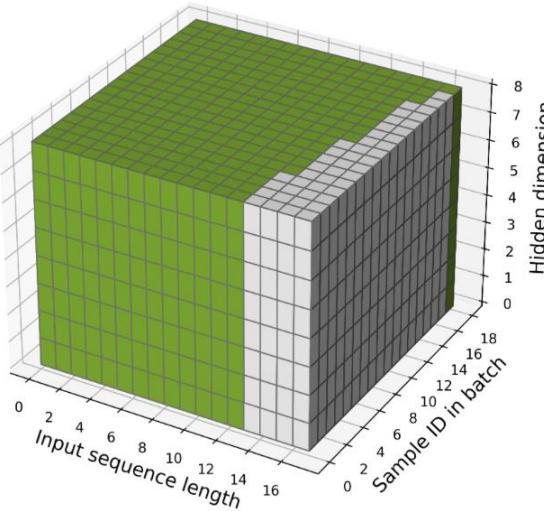
Input len: (0, 7)

Bucket 0



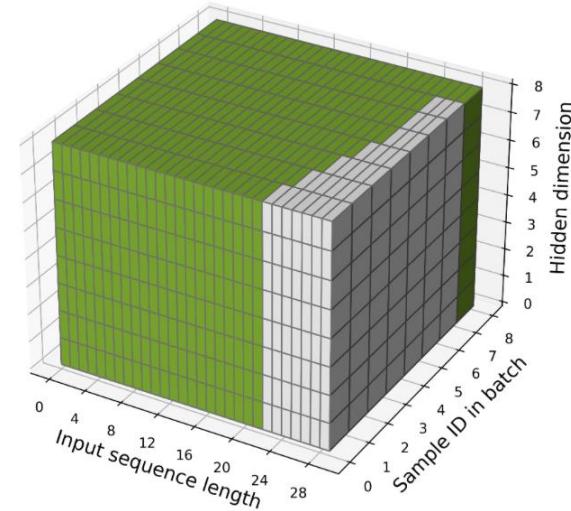
Input len: (8, 17)

Bucket 1



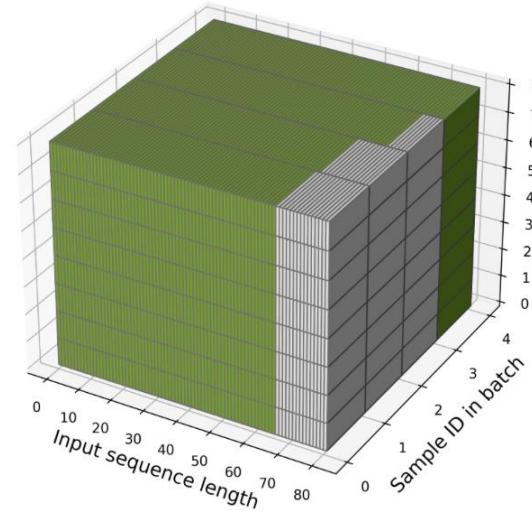
Input len: (17, 25)

Bucket 2



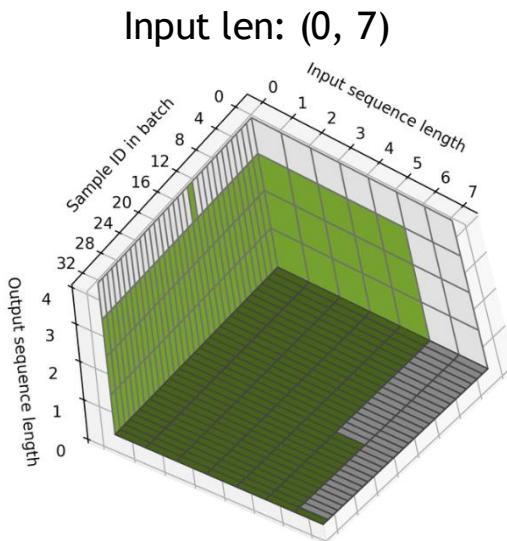
Input len: (25, 37)

Bucket 3

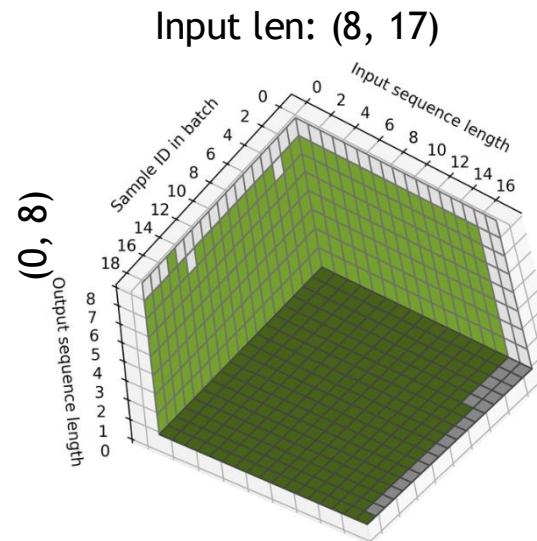


2D BUCKETING

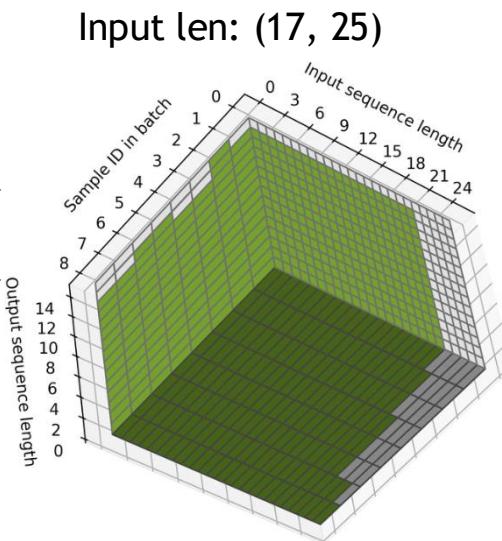
Output len: (0, 4)



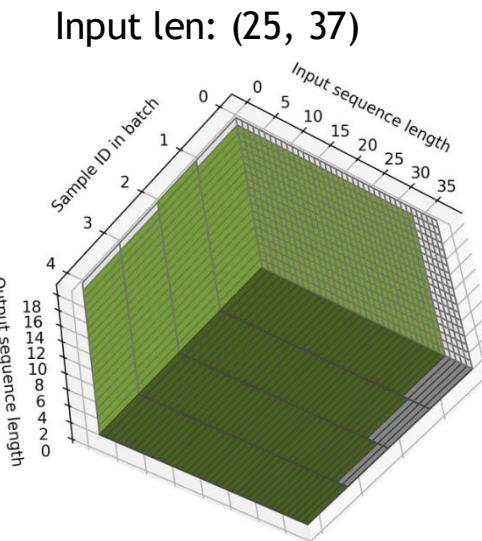
(0, 8)



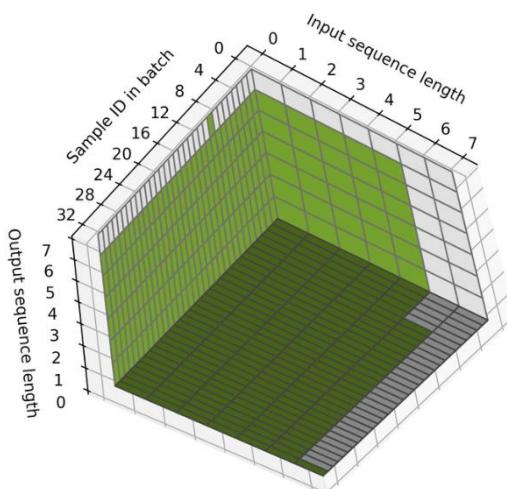
(0, 15)



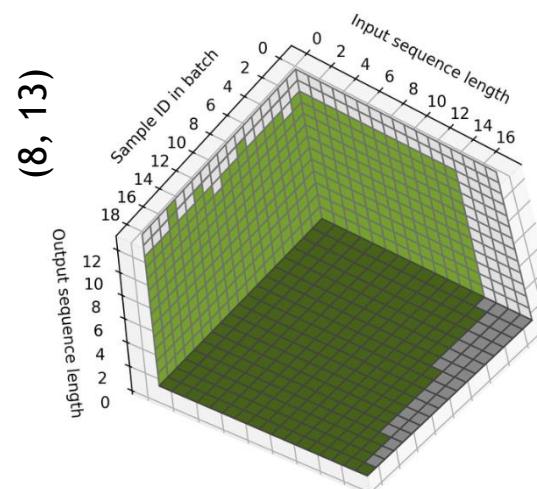
(0, 19)



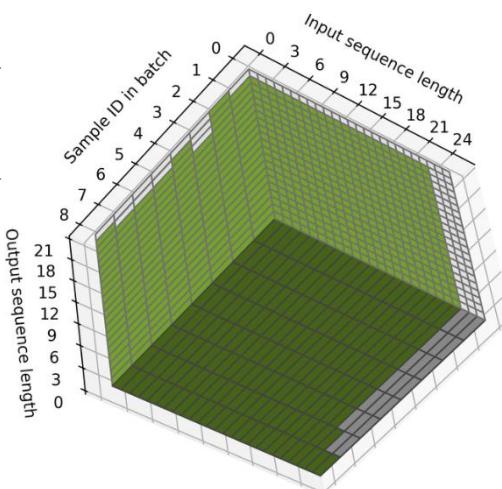
Output len: (4, 7)



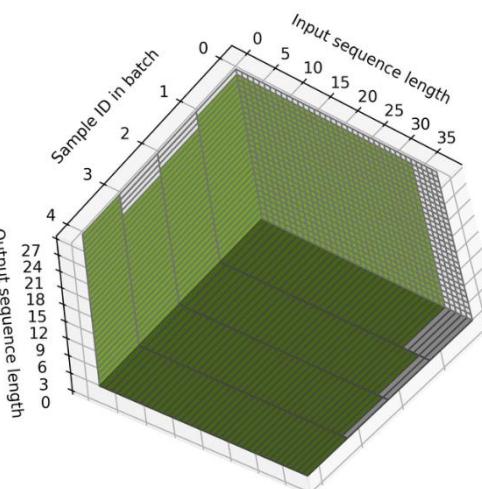
(8, 13)



(15, 21)



(19, 27)



DETERMINE BUCKET BINS

Algorithm:

- Estimate bins based on data distribution
 - measure input sequence lengths
 - measure output sequence lengths
 - measure output-to-input (OTI) ratios
 - filter OTI outliers ($>4\sigma$)
 - each bucket must hold equal cumulative length: $\text{sum}(\text{lens}) / \text{num_buckets}$
- Hyperparameters
 - number of 1st dim buckets
 - number of 2nd dim (sub)buckets

Configuration:
`num_buckets = 3`

Measured data distribution:
`seq_lens = [2, 3, 3, 3, 4, 5, 5, 6, 8, 9]`

Outcome:
`buckets = [4, 6, 9]`

15 16 17

DETERMINE BUCKET BINS

Algorithm:

- Estimate bins based on data distribution
 - measure input sequence lengths
 - measure output sequence lengths
 - measure output-to-input (OTI) ratios
 - filter OTI outliers ($>4\sigma$)
 - each bucket must hold equal cumulative length: $\text{sum}(\text{lens}) / \text{num_buckets}$
- Hyperparameters
 - number of 1st dim buckets
 - number of 2nd dim (sub)buckets

Configuration:
num_buckets = 3

Measured data distribution:
seq_lens = [2, 3, 3, 3, 4, 5, 5, 6, 8, 9]

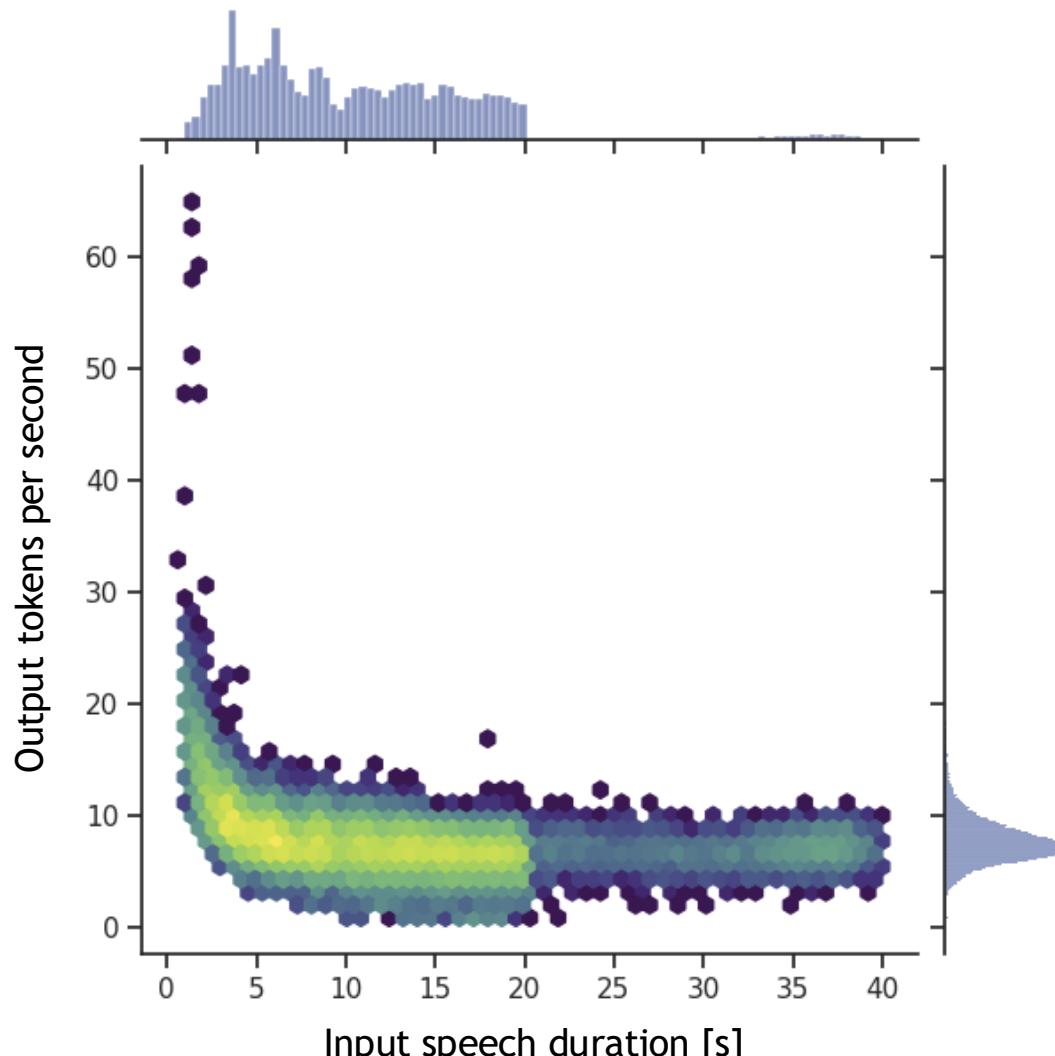
Outcome:
buckets = [4, 6, 9]

Example of actual estimation:

```
[bucket <= 3.79s] [10 - 65] [approx-max-tps: 20.18] Discarded 145 max token outliers min-outlier: 20.31, max-outlier: 110.78).
[bucket <= 4.81s] [10 - 66] [approx-max-tps: 13.95] Discarded 40 max token outliers min-outlier: 14.00, max-outlier: 27.99).
[bucket <= 5.75s] [10 - 73] [approx-max-tps: 13.00] Discarded 37 max token outliers min-outlier: 13.06, max-outlier: 18.16).
[bucket <= 6.43s] [10 - 79] [approx-max-tps: 12.72] Discarded 38 max token outliers min-outlier: 12.75, max-outlier: 15.36).
[bucket <= 7.20s] [10 - 88] [approx-max-tps: 12.36] Discarded 19 max token outliers min-outlier: 12.41, max-outlier: 22.11).
[bucket <= 8.12s] [10 - 100] [approx-max-tps: 12.48] Discarded 17 max token outliers min-outlier: 12.63, max-outlier: 16.70).
[bucket <= 8.79s] [10 - 115] [approx-max-tps: 13.13] Discarded 4 max token outliers min-outlier: 13.19, max-outlier: 15.78).
[bucket <= 9.63s] [10 - 115] [approx-max-tps: 12.73] Discarded 3 max token outliers min-outlier: 13.28, max-outlier: 13.63).
[bucket <= 10.48s] [10 - 127] [approx-max-tps: 12.82] Discarded 0 max token outliers
```

```
bucket_duration_bins=[[3.792,27],[3.792,65],[4.811,34],[4.811,66],[5.750,39],[5.750,73],[6.432,44],[6.432,79],[7.200,47],[7.200,88],
max_tps=[20.18,20.18,13.95,13.95,13.00,13.00,12.72,12.72,12.36,12.36,
```

OUTPUT TO INPUT LENGTH RATIOS



Source: Canary training data

DETERMINE BATCH SIZES: OOMPTIMIZER

```
The current sequence lengths are: input=639792 output=437.
```

```
[BEGIN step] [CUDA RAM CURRENT: 3414.0MB] [CUDA RAM MAX: 3414.0MB]
```

```
Current gap: None... OOM!
```

```
[END step] [CUDA RAM CURRENT: 5022.2MB] [CUDA RAM MAX: 72766.3MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 4944.0MB] [CUDA RAM MAX: 4944.0MB]
```

```
Current gap: None... OK!
```

```
[END step] [CUDA RAM CURRENT: 15265.1MB] [CUDA RAM MAX: 42743.5MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 15226.0MB] [CUDA RAM MAX: 15226.0MB]
```

```
Current gap: 0.5... OK!
```

```
[END step] [CUDA RAM CURRENT: 15284.7MB] [CUDA RAM MAX: 68279.7MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 15226.0MB] [CUDA RAM MAX: 15226.0MB]
```

```
Current gap: 0.25... OOM!
```

```
[END step] [CUDA RAM CURRENT: 11923.9MB] [CUDA RAM MAX: 70644.3MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 11855.5MB] [CUDA RAM MAX: 11855.5MB]
```

```
Current gap: 0.14285714285714285... OOM!
```

```
[END step] [CUDA RAM CURRENT: 11919.0MB] [CUDA RAM MAX: 71567.6MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 11855.5MB] [CUDA RAM MAX: 11855.5MB]
```

```
Current gap: 0.07692307692307693... OK!
```

```
[END step] [CUDA RAM CURRENT: 15287.1MB] [CUDA RAM MAX: 70628.1MB]
```

```
=====
```

```
[BEGIN step] [CUDA RAM CURRENT: 15226.0MB] [CUDA RAM MAX: 15226.0MB]
```

```
Current gap: 0.038461538461538464... OOM!
```

```
[END step] [CUDA RAM CURRENT: 11919.0MB] [CUDA RAM MAX: 71567.6MB]
```

```
=> Optimal setting for bucket=(39.987, 437) (input=639792 output=437) is max_batch_size=25
```

Inputs:

- Estimated bucket bins
- Model configuration

Algorithm:

- For each bucket
 - Run training step with an initial batch size B
 - Double/halve B depending on success/OOM
- Find one setting that works and one that OOMs
- Bisect until convergence

Open issues:

- Tricky to faithfully simulate actual training memory usage
- Model parallelism (TP/PP/CP)

METHOD: RECAP

- I. Estimate bucket bins in a data-driven way.
 - Use 2D bucketing (stratify on input+output sequence lengths).
 - `scripts/speech_recognition/estimate_duration_buckets_2d.py`
- II. How to determine the batch size for each bucket?
 - Solution: given bucket bins, determine actual memory usage and find the maximum batch size succeeding without OOM.
 - `scripts/speech_recognition/oomoptimizer.py`
- III. Output sequence length outliers
 - Output-to-Input (OTI) ratio, aka: tokens/second (tps), token/token (tpt).
 - High OTI ratio outliers often indicate bad data / hallucinations. Filtering those helps both efficiency and convergence!

METHOD: RECAP

- Canary data loading configuration example

```
# Filters
max_tps: 25 # maximum output tokens per input second of speech
max_duration: 40.0
text_field: answer
lang_field: target_lang

# Batch size / bucketing settings
use_bucketing: true
bucket_duration_bins: [[3.971,30],[3.971,48],[4.973,37],[4.973,60],[5.85,42],[5.85,71],[6.56,46],[6.56,79],[7.32,49],[7.32,88],[8.19,54],[8.19,99],[8.88,61],[8.88,107],[9.76,66],[9.76,118],[10.56,72],[10.56,127],[11.214,76],[11.214,135],[11.867,79],[11.867,143],[12.53,82],[12.53,151],[13.08,87],[13.08,157],[13.62,91],[13.62,164],[14.16,93],[14.16,170],[14.7,96],[14.7,177],[15.2,99],[15.2,183],[15.68,101],[15.68,189],[16.135,102],[16.135,194],[16.67,105],[16.67,201],[17.197,108],[17.197,207],[17.73,111],[17.73,213],[18.2,114],[18.2,219],[18.69,117],[18.69,225],[19.153,120],[19.153,230],[19.63,123],[19.63,236],[20.44,122],[20.44,246],[32.567,174],[32.567,391],[36.587,227],[36.587,440],[40.0,253],[40.0,480]]
bucket_batch_size: [273,248,217,190,190,161,165,140,149,126,134,113,120,100,109,93,99,86,94,79,86,72,82,69,79,66,74,63,71,61,68,57,66,55,66,54,63,52,61,50,59,48,57,46,55,45,53,43,51,41,50,39,50,39,28,21,24,18,21,17]
bucket_buffer_size: 40000
shuffle_buffer_size: 10000
```

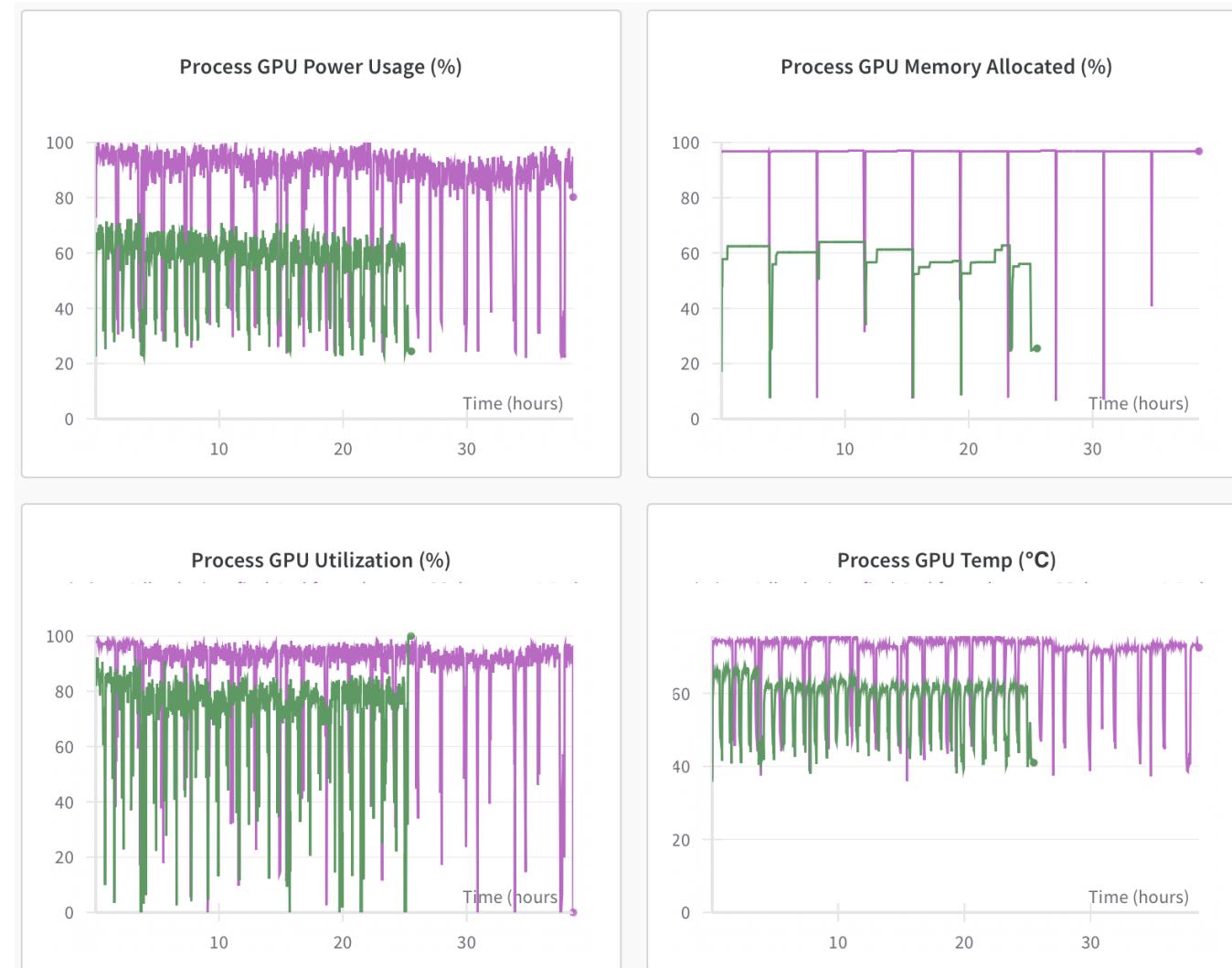
- Applied in a NeMo training config config such as:
https://github.com/NVIDIA/NeMo/blob/main/examples/asr/conf/speech_multitask/fast-conformer_aed.yaml

BUCKETING: FILTERING AND OOMPTIMIZER

Canary-1B training

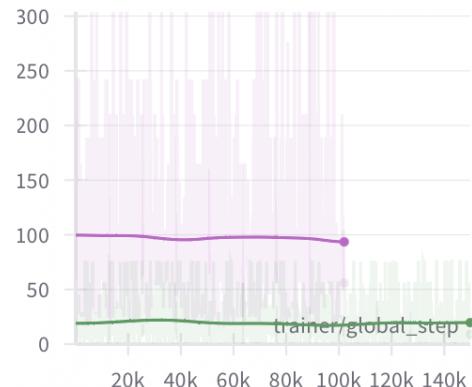
Legend:

- (green) original Canary-1B run; baseline bucketing
- (violet) + 2D bucketing + TPS filtering + OOMoptimizer

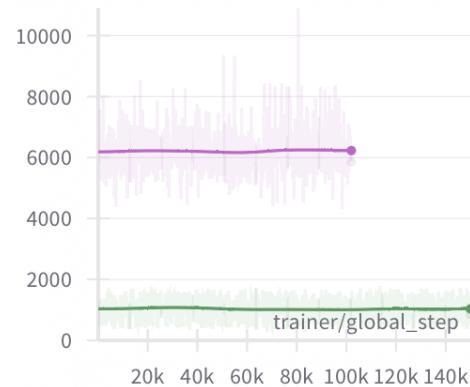


EFFICIENCY: FILTERING AND OOMPTIMIZER

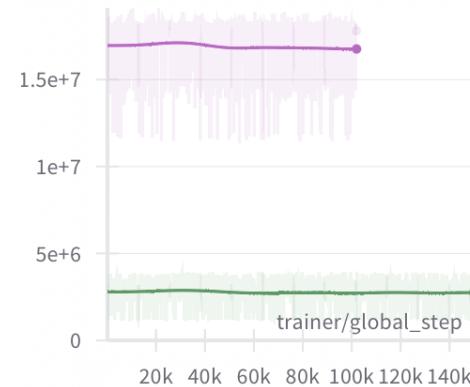
batch_size



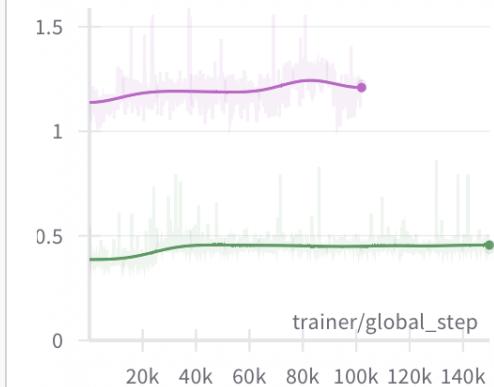
num_tokens



num_frames



train_step_timing in s



Canary-1B training

Legend:

- (green) Baseline 1D bucketing
- (violet) 2D bucketing, TPS filtering & OOMoptimizer

OPTIMIZING CANARY-1B TRAINING

Canary-1B Training Run	Number of GPUs	Training Time	HuggingFace Open ASR Leaderboard WER (mean)
Baseline	128	36h	6.54%
OOMoptimized	32	36h	6.51%
OOMoptimized	128	19h	6.47%

COMPARISON TO WHISPER-STYLE TRAINING



Legend (Canary-1B, 32xA100 GPU, max TPS=25):

- (green): Whisper-style (fixed batch size = 24, global=768)
- (brown): 2D Bucketing (30x2) + OOMoptimizer

CONCLUSIONS

- OOMoptimizer + 2D bucketing + tokens/second filtering enabled 80% training throughput increase (5x batch_size / 2.75x train_step_time)
 - Train Canary-1B with the same resources **2x faster**
 - Train Canary-1B with **4x less resources** at original speed
 - Similar speedups with SALM, BESTOW, GPT, T5 (EMMETT)
- Applicable to multiple modalities (audio, text):
generic efficiency improvement.



MULTIMODAL SPEECHLLM

EMMETT - Motivation

Efficient Multimodal Machine Translation Training

Our first attempt to build a multimodal SpeechLLM.

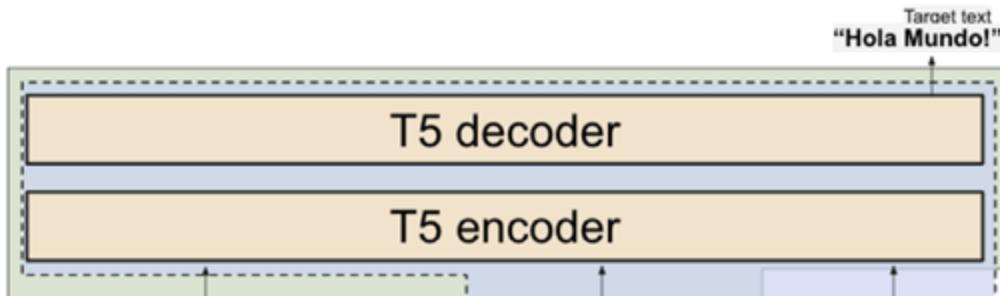
Extend LLMs to speech and demonstrate continual learning.

Understand how to blend multiple data modalities in training.

Extend OOMoptimizer to multimodal training.

EMMETT - Architecture

Text-only T5

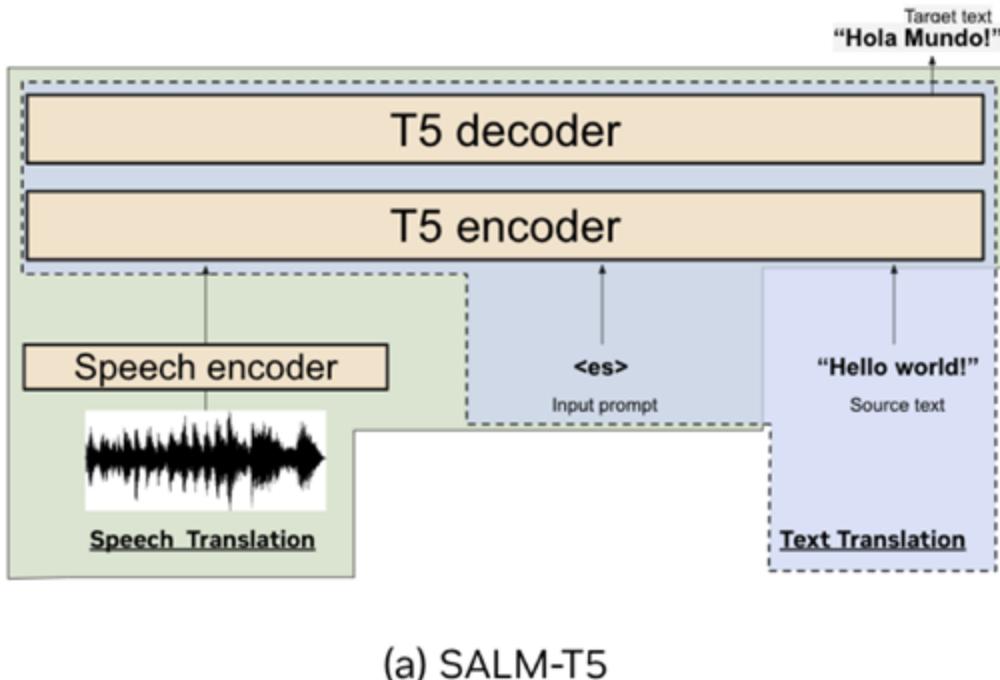


Architecture 1:

- Encoder-decoder Megatron T5
- Trained for NMT
- Text-only

EMMETT - Architecture

Audio-only and multimodal SALM-T5



Architecture 1:

- Encoder-decoder Megatron T5
- Trained for NMT
- **Text-only**

Extension to speech:

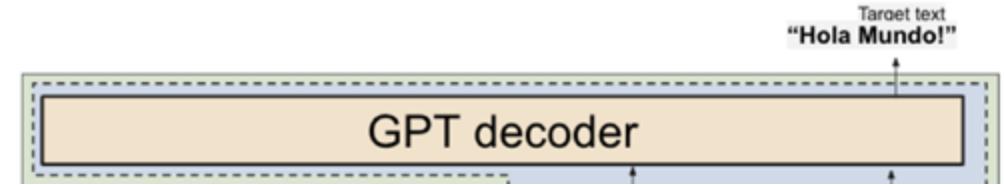
- Canary-1B encoder
- SALM mechanism
- Full SFT (no adapters)
- Training variants:
 - **audio-only**
 - **audio+text (multimodal)**

EMMETT - Architecture

Text-only GPT

Architecture 2:

- Decoder-only GPT (TinyLlama)
- Trained as a general LLM
- **Text-only**



EMMETT - Architecture

Audio-only and multimodal BESTOW-GPT

Architecture 2:

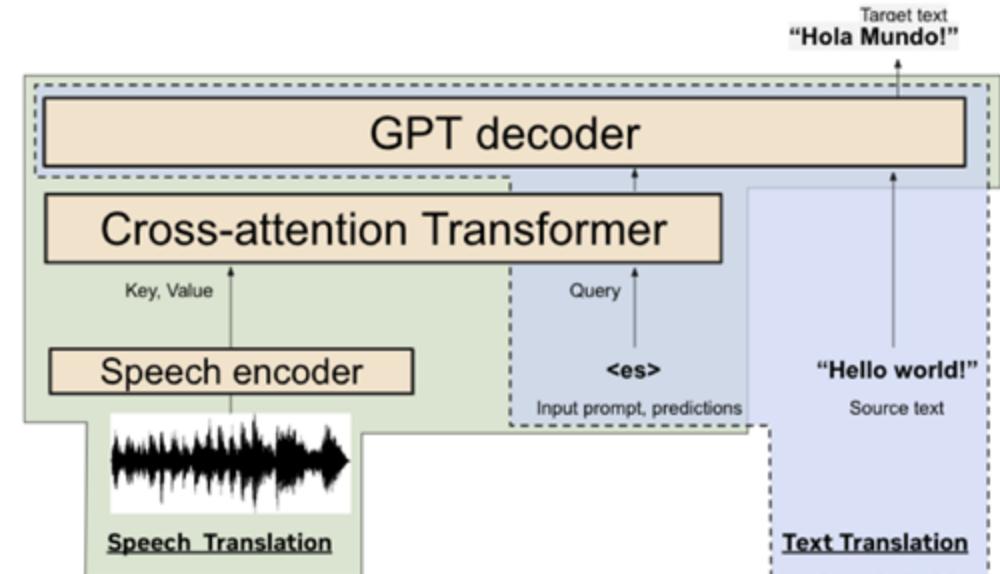
- Decoder-only GPT (TinyLlama)
- Trained as a general LLM
- **Text-only**

Extension to speech:

- Canary-1B encoder
- BESTOW mechanism
- Full SFT (no adapters)

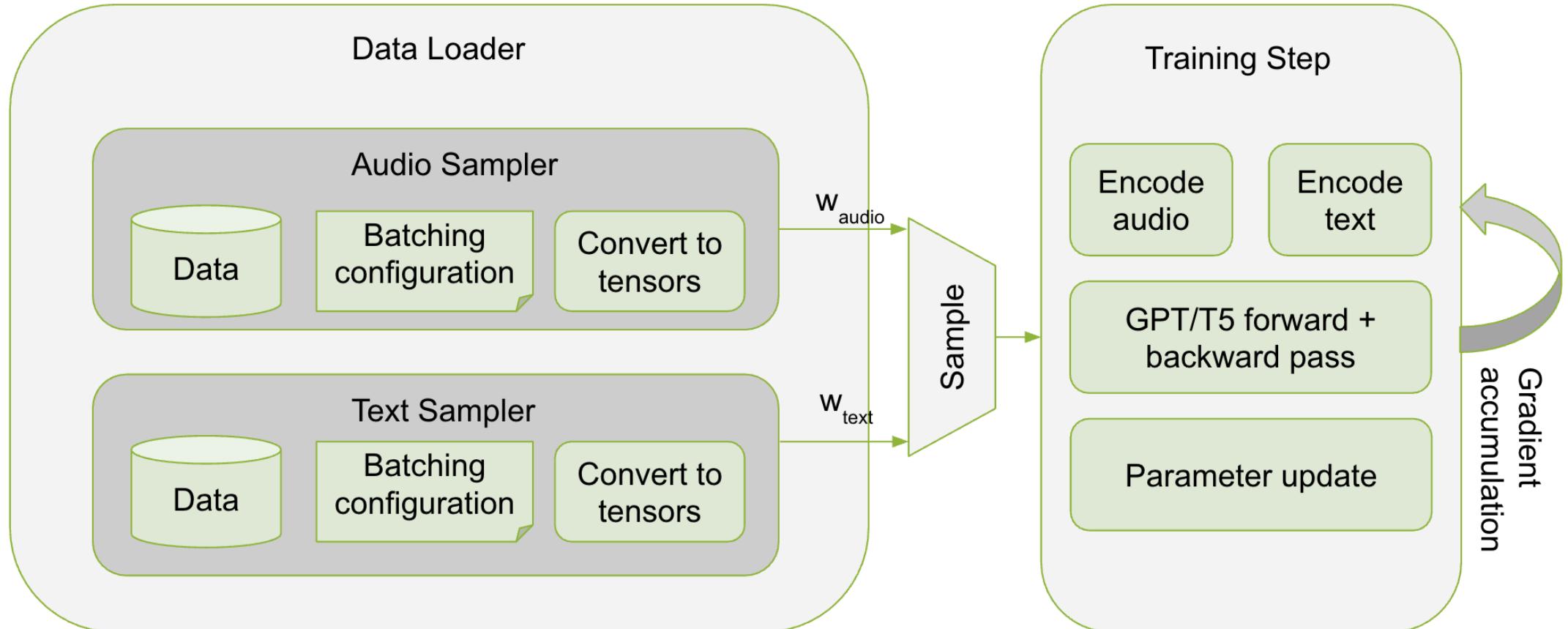
Training variants:

- **audio-only**
- **audio+text (multimodal)**



(b) BESTOW-GPT

EMMETT - Multimodal Training



EMMETT - Experimental Setup

Speech:

- Canary-1B training setup (85k hours)
- ASR + EN->X + X->EN, X={DE, FR, ES}

Text:

- Internal text data with translation pairs for NMT
- 33 language pairs, 2.7TB of text

Optimization:

- Baseline: 1D bucketing, batch size determined heuristically (batch_tokens)
- Optimized: 2D bucketing + OOMoptimizer

EMMETT - AST Results

TABLE I
SPEECH TRANSLATION SACREBLEU SCORES ON A SUBSET OF FLEURS.
THE BASELINES ARE SEAMLESSM4T-v2 [34], CANARY-1B AST [20],
AND CANARY-1B ASR CASCaded WITH PRETRAINED T5 NMT.
BESTOW-GPT AND SALM-T5 ARE TINYLLAMA AND T5 MODELS
FINETUNED EXCLUSIVELY ON SPEECH. MULTIMODAL INDICATES JOINT
FINETUNING ON SPEECH AND TEXT.

Model	En →			Fr			Avg
	De	Es	Fr	De	Es → En	Fr	
<i>Baselines</i>							
SeamlessM4T-v2	33.2	23.7	43.0	37.1	25.4	30.9	32.2
Canary-1B	32.1	22.7	40.8	34.0	21.8	31.0	30.4
+T5 cascade	32.6	23.9	42.6	39.4	27.3	37.9	34.0
BESTOW-GPT	32.0	23.1	41.2	35.8	23.9	35.1	32.0
+multimodal	33.2	24.5	42.7	37.9	24.9	36.8	33.3
SALM-T5	34.2	24.1	43.5	38.8	25.9	37.0	33.9
+multimodal	34.6	24.5	44.6	37.9	27.5	37.4	34.4

EMMETT - NMT Results

TABLE II
TEXT TRANSLATION SACREBLEU SCORES ON A SUBSET OF FLORES.
THE T5 NMT BASELINE IS TRAINED EXCLUSIVELY ON TEXT, AND
EXTENDED TO AUDIO MODALITY AS SALM-T5.

Model	En →			De	Es	Fr	Avg
	De	Es	Fr				
T5 NMT	38.1	27.5	50.2	44.7	30.3	45.7	39.4
SALM-T5				N/A			
+multimodal	38.7	27.5	50.4	44.1	30.9	45.8	39.6

EMMETT - Efficiency Results

TABLE III
MULTIMODAL TRAINING EFFICIENCY GAINS FROM 2D BUCKETING AND
OOMPTIMIZER (OPT).

Model	Steps	Runtime	Mean batch size		Steps/sec
			Audio	Text	
BESTOW-GPT	450k	7 days	13	100	1.15
+ <i>opt</i>	100k	2.5 days	55	269	0.81
SALM-T5+ <i>opt</i>	7.5k	5 hours	30	251	0.7

EMMETT: Lessons Learned

- “Weak” continual learning achieved.
 - Provided the original text data is available, the model retains its text-domain capabilities.
- Multimodal variant is at least as good as the single-modal variant.
- Improved batching strategies (2D bucketing, OOMoptimier)
 - Apply to both audio and text modalities
 - Drastically reduce the training time (2.5x observed for BESTOW-GPT)

VOICE-TEXT (VT) BLENDER

Voice-Text Blender Goal

To build a multi-modal Speech-Text Language Model (STLM) that seamlessly integrates **speech understanding** and **multi-turn mixed-modal chat** capabilities, while preserving the **original text-only performance** and demonstrating **zero-shot capabilities** for speech tasks.

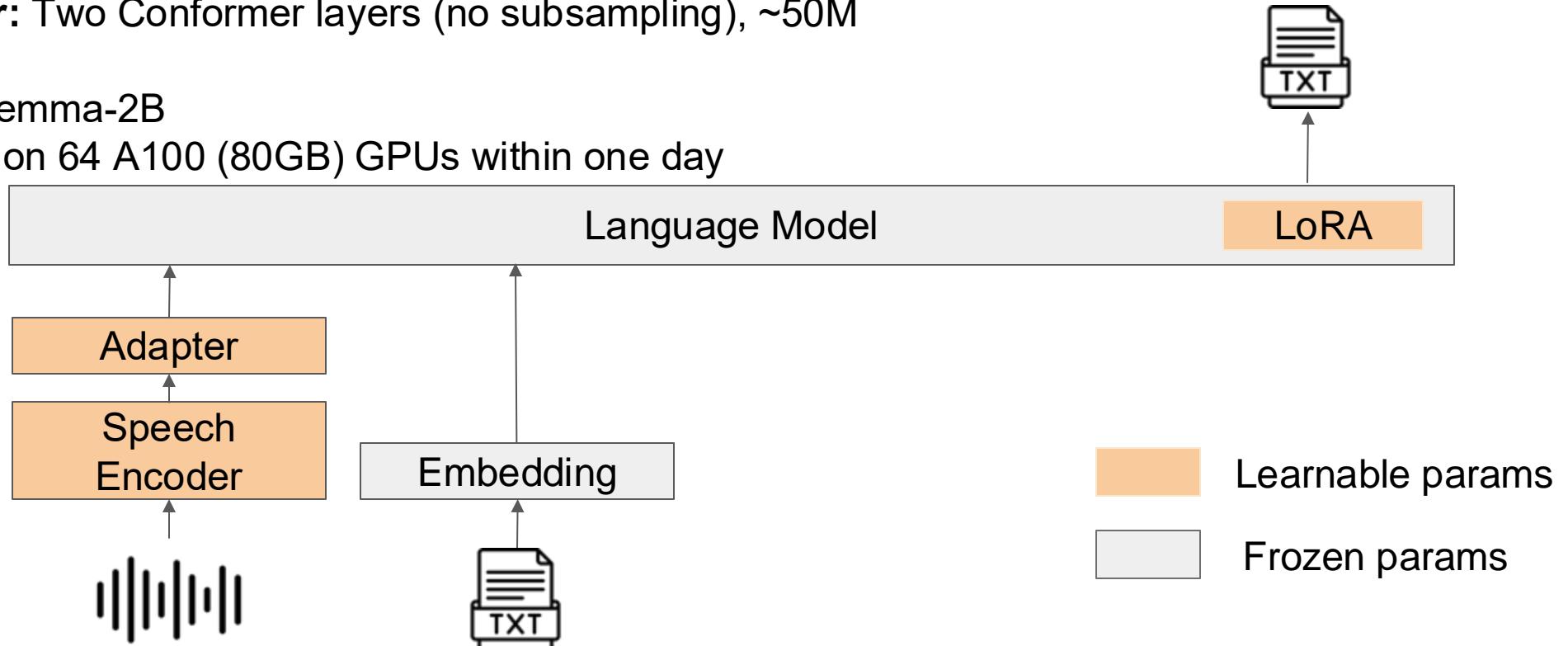
Model Architecture

Speech Encoder: Canary, 80ms

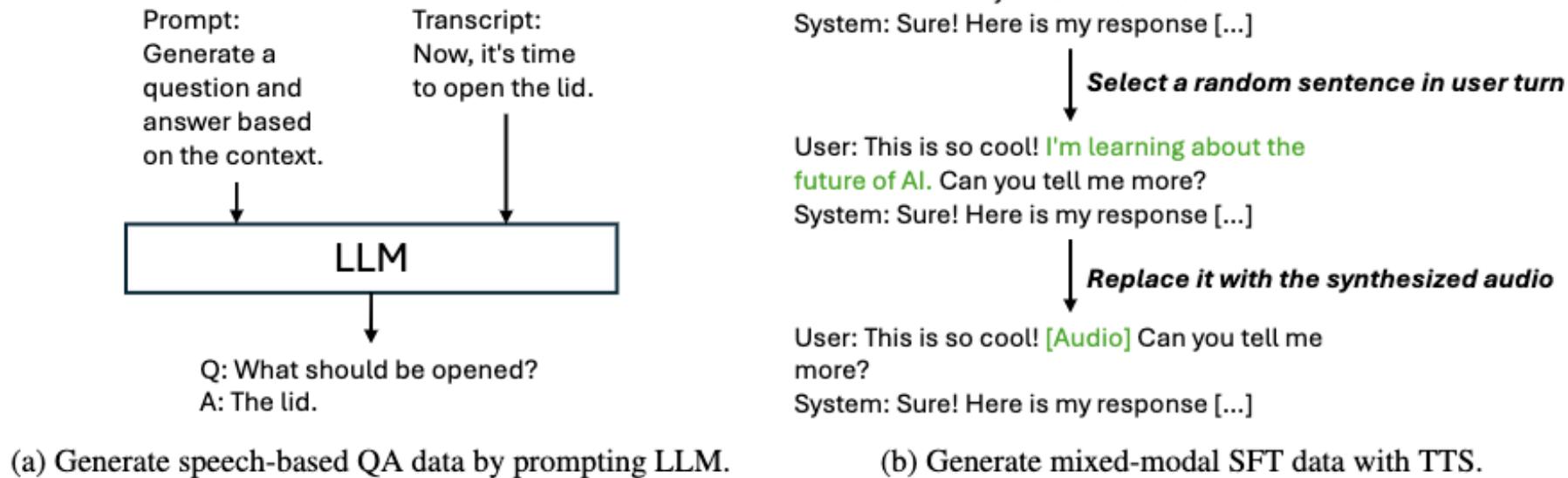
Adapter: Two Conformer layers (no subsampling), ~50M
params

LLM: Gemma-2B

Trained on 64 A100 (80GB) GPUs within one day



Data Generation

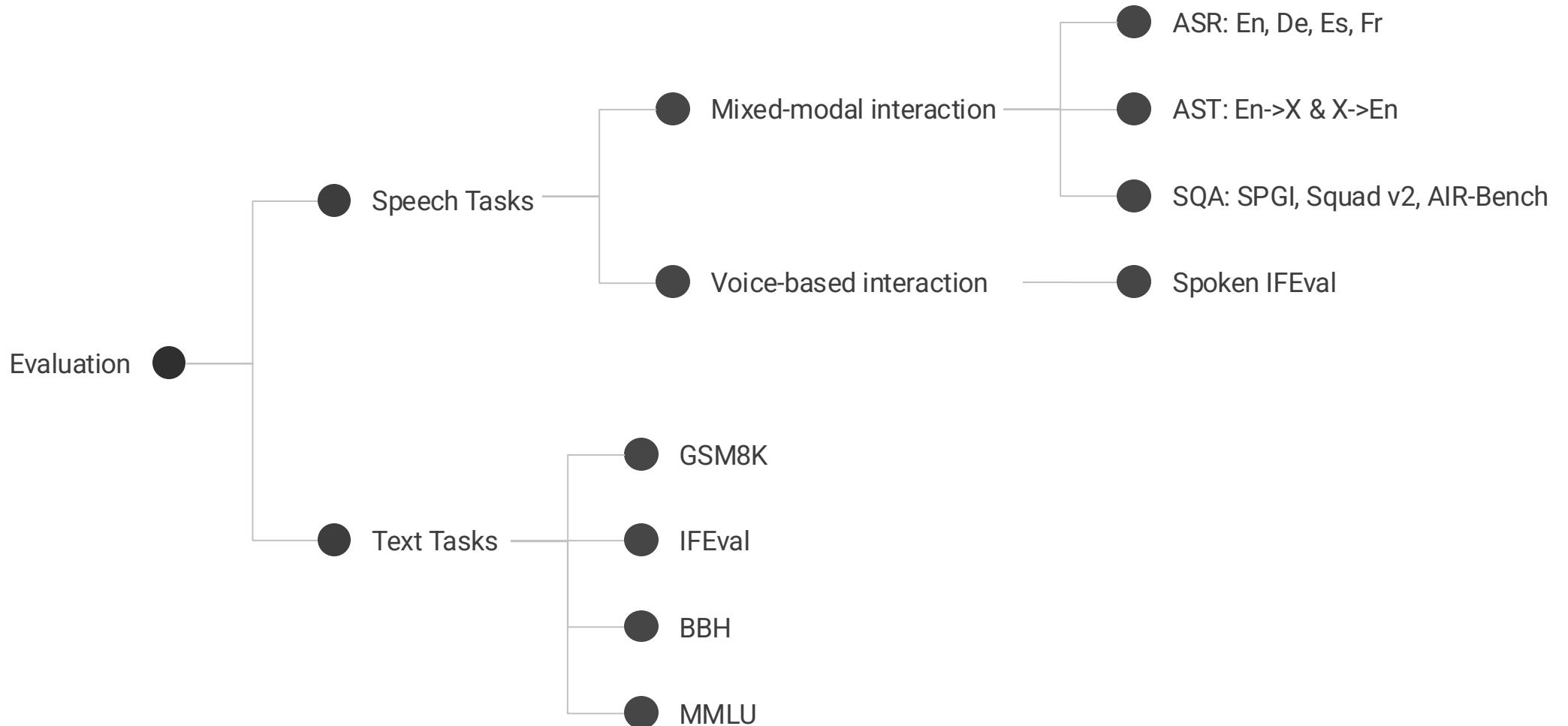


Data Statistics

Task	Dataset	#Samples	#Hours	Sampling Ratio
Text-only SFT	Nemotron	94.0k	N/A	0.1500
ASR, AST	Canary	32.8M	85k	0.7556
Speech-based QA	Canary Subset	4.1M	20k	0.0378
Mixed-modal SFT	Alpaca	55.3k	85	0.0189
	Magpie	254.5k	461	0.0378

Table 1: Statistics of our training data mixture. When creating mini-batches, different types of data are sampled according to the ratio shown in the last column.

Evaluation Design



Evaluation Design

Task	Dataset	Languages	Metric
ASR	CommonVoice	En, De, Es, Fr	WER
AST	FLEURS	En-De, En-Es, En-Fr De-En, Es-En, Fr-En	BLEU
SQA	SPGI SQuAD2 AIR-Bench	En	GPT Score
Speech-only	IFEval	En	Prompt-level Strict Accuracy
Text-only	GSM8K IFEval BBH MMLU	En	5-shot Exact Match (flexible extract) Prompt-level Strict Accuracy 3-shot CoT Accuracy 5-shot Accuracy

Results

Model	ASR WER ↓				En-X BLEU ↑			X-En BLEU ↑			Speech-based QA ↑			Speech ↑		Text ↑			
	En	De	Es	Fr	De	Es	Fr	De	Es	Fr	SPGI	SQuAD2	AIR.	IFEval	GSM8K	IFEval	BBH	MMLU	
<i>Prior studies</i>																			
Whisper-v3 1.5B	9.92	6.17	4.94	11.18		N/A		33.4	22.7	33.7						N/A			
SALMONN 7B	20.84	40.83	37.47	36.78	18.0	17.1	27.8	5.1	7.1	3.3	0.778	0.597	-	0.147	-	-	-	-	
SALMONN 13B	17.07	44.08	28.47	38.52	19.0	18.5	29.1	6.5	3.6	3.8	0.778	0.604	6.16	0.113	-	-	-	-	
Qwen2-Audio 7B [†]	8.78	7.67	5.65	9.49	24.8	18.9	27.7	30.7	22.2	29.6	0.810	0.656	7.24	0.140	-	-	-	-	
<i>Text-only baseline</i>																			
Gemma 2.5B								N/A							0.2479	0.2089	0.3324	0.3554	
<i>Ours</i>																			
VTBlender 3B	7.90	5.53	4.52	7.09	29.6	22.5	38.6	36.3	25.6	33.8	0.828	0.684	6.31	0.191	0.2358	0.2237	0.3003	0.3484	

Results

Model	ASR WER ↓				En-X BLEU ↑			X-En BLEU ↑			Speech-based QA ↑			Speech ↑		Text ↑		
	En	De	Es	Fr	De	Es	Fr	De	Es	Fr	SPGI	SQuAD2	AIR.	IFEval	GSM8K	IFEval	BBH	MMLU
<i>Prior studies</i>																		
Whisper-v3 1.5B	9.92	6.17	4.94	11.18		N/A		33.4	22.7	33.7						N/A		
SALMONN 7B	20.84	40.83	37.47	36.78	18.0	17.1	27.8	5.1	7.1	3.3	0.778	0.597	-	0.147	-	-	-	-
SALMONN 13B	17.07	44.08	28.47	38.52	19.0	18.5	29.1	6.5	3.6	3.8	0.778	0.604	6.16	0.113	-	-	-	-
Qwen2-Audio 7B [†]	8.78	7.67	5.65	9.49	24.8	18.9	27.7	30.7	22.2	29.6	0.810	0.656	7.24	0.140	-	-	-	-
<i>Text-only baseline</i>																		
Gemma 2.5B						N/A									0.2479	0.2089	0.3324	0.3554
<i>Ours</i>																		
VTBlender 3B	7.90	5.53	4.52	7.09	29.6	22.5	38.6	36.3	25.6	33.8	0.828	0.684	6.31	0.191	0.2358	0.2237	0.3003	0.3484

Results

Model	ASR WER ↓				En-X BLEU ↑			X-En BLEU ↑			Speech-based QA ↑			Speech ↑		Text ↑			
	En	De	Es	Fr	De	Es	Fr	De	Es	Fr	SPGI	SQuAD2	AIR.	IFEval	GSM8K	IFEval	BBH	MMLU	
<i>Prior studies</i>																			
Whisper-v3 1.5B	9.92	6.17	4.94	11.18	N/A	33.4	22.7	33.7						N/A					
SALMONN 7B	20.84	40.83	37.47	36.78	18.0	17.1	27.8	5.1	7.1	3.3	0.778	0.597	-	0.147	-	-	-	-	
SALMONN 13B	17.07	44.08	28.47	38.52	19.0	18.5	29.1	6.5	3.6	3.8	0.778	0.604	6.16	0.113	-	-	-	-	
Qwen2-Audio 7B [†]	8.78	7.67	5.65	9.49	24.8	18.9	27.7	30.7	22.2	29.6	0.810	0.656	7.24	0.140	-	-	-	-	
<i>Text-only baseline</i>																			
Gemma 2.5B					N/A										0.2479	0.2089	0.3324	0.3554	
<i>Ours</i>																			
VTBlender 3B	7.90	5.53	4.52	7.09	29.6	22.5	38.6	36.3	25.6	33.8	0.828	0.684	6.31	0.191	0.2358	0.2237	0.3003	0.3484	

Results

Model	ASR WER ↓				En-X BLEU ↑			X-En BLEU ↑			Speech-based QA ↑			Speech ↑		Text ↑			
	En	De	Es	Fr	De	Es	Fr	De	Es	Fr	SPGI	SQuAD2	AIR.	IFEval	GSM8K	IFEval	BBH	MMLU	
<i>Prior studies</i>																			
Whisper-v3 1.5B	9.92	6.17	4.94	11.18	N/A	33.4	22.7	33.7						N/A					
SALMONN 7B	20.84	40.83	37.47	36.78	18.0	17.1	27.8	5.1	7.1	3.3	0.778	0.597	-	0.147	-	-	-	-	
SALMONN 13B	17.07	44.08	28.47	38.52	19.0	18.5	29.1	6.5	3.6	3.8	0.778	0.604	6.16	0.113	-	-	-	-	
Qwen2-Audio 7B [†]	8.78	7.67	5.65	9.49	24.8	18.9	27.7	30.7	22.2	29.6	0.810	0.656	7.24	0.140	-	-	-	-	
<i>Text-only baseline</i>																			
Gemma 2.5B					N/A									0.2479	0.2089	0.3324	0.3554		
<i>Ours</i>																			
VTBlender 3B	7.90	5.53	4.52	7.09	29.6	22.5	38.6	36.3	25.6	33.8	0.828	0.684	6.31	0.191	0.2358	0.2237	0.3003	0.3484	

Results

Model	ASR WER ↓				En-X BLEU ↑			X-En BLEU ↑			Speech-based QA ↑			Speech ↑		Text ↑			
	En	De	Es	Fr	De	Es	Fr	De	Es	Fr	SPGI	SQuAD2	AIR.	IEEval		GSM8K	IEEval	BBH	MMLU
<i>Prior studies</i>																			
Whisper-v3 1.5B	9.92	6.17	4.94	11.18	N/A	33.4	22.7	33.7							N/A				
SALMONN 7B	20.84	40.83	37.47	36.78	18.0	17.1	27.8	5.1	7.1	3.3	0.778	0.597	-	0.147	-	-	-	-	
SALMONN 13B	17.07	44.08	28.47	38.52	19.0	18.5	29.1	6.5	3.6	3.8	0.778	0.604	6.16	0.113	-	-	-	-	
Qwen2-Audio 7B [†]	8.78	7.67	5.65	9.49	24.8	18.9	27.7	30.7	22.2	29.6	0.810	0.656	7.24	0.140	-	-	-	-	
<i>Text-only baseline</i>																			
Gemma 2.5B																0.2479	0.2089	0.3324	0.3554
<i>Ours</i>																			
VTBlender 3B	7.90	5.53	4.52	7.09	29.6	22.5	38.6	36.3	25.6	33.8	0.828	0.684	6.31	0.191	0.2358	0.2237	0.3003	0.3484	

ASR

We can control the output format in a single instruction.



The audio can be placed anywhere in the user turn.

NeMo-STLM

Transcribe the English content. [audio] Use the symbol \$ instead of the word "dollar" if necessary.

▶ 0:00 / 0:02 ━━━━ 🔍 ⏮

The price of the pen is \$10.

The price of the pen is \$10.

NeMo-STLM

Recognize the English content. Write your output in json format with the key "transcript". [audio]

▶ 0:00 / 0:02 ━━━━ 🔍 ⏮

```
{  
  "transcript": "The price of the pen is $10."  
}
```

AST

The model is trained on En->X and X->En, but it can generalize to unseen directions like Es->Fr.

Audio content:

Los modelos de lenguaje grandes son sofisticados sistemas de inteligencia artificial diseñados para comprender y generar texto similar al humano basado en grandes cantidades de datos.

NeMo-STLM

Translate the Spanish content to English. [audio]

▶ 0:00 / 0:10



Large language models are sophisticated artificial intelligence systems designed to understand and generate text similar to human based on large amounts of data.

NeMo-STLM

Translate the Spanish content to French. [audio]

▶ 0:00 / 0:10



Les modèles de langage grand sont des systèmes sophistiqués d'intelligence artificielle conçus pour comprendre et générer du texte similaire à l'humain basé sur de grandes quantités de données.

Speech Understanding

Slot filling



Audio content:

I will travel from San Jose to New York on April 1st. What are the options?

NeMo-STLM

Extract the destination and date mentioned here: [audio] Write your output in json format with two keys "destination" and "date".

▶ 0:06 / 0:06



```
{  
"destination": "New York",  
"date": "2023-04-01"  
}
```

Simple Math based on Mixed-Modal Input

The model can understand the speech content and connect it with additional text information.

NeMo-STLM

Recognize the English content. [audio] There is a 10% discount today. Output the discounted price.

▶ 0:00 / 0:02 ⏸

The discounted price of the pen is \$9.00.

Audio content:

The price of the pen is 10 dollars.

NeMo-STLM

I bought 3 such pens. [audio] I also bought a book with 5 dollars. How much did I spend in total?

▶ 0:00 / 0:02 ⏸ ⏴ ⏵

Here's how to figure that out:

- Total cost of pens: $3 \text{ pens} * 10/\text{pen} = 30$
- Total cost of book: \$5

Total cost: $30(\text{pens}) + 5 (\text{book}) = \35

Let me know if you have any other questions!

Simple Coding with Instruction Following



Audio content:

Write a function that takes a list of integers and returns the sum of all even numbers in the list.

NeMo-STLM

[audio] Just output your code. You don't need to provide any example.

▶ 0:00 / 0:04

```
python
def sum_even_numbers(numbers):
    even_sum = 0
    for num in numbers:
        if num % 2 == 0:
            even_sum += num
    return even_sum
```

Model Capabilities

Voice-Text Blender is capable of:

- Transcribe/Translate audio
- Speech understanding
- Context biasing
- Zero-shot speech capabilities
- Mixed-modal input
- Voice mode interaction and Instruction following
- Mixed-modal Multi-turn chat
- Preserves text-only task performance

Key Findings

- Mixing in pure text SFT data preserves the original text performance without hurting speech performance
- In the current setup, single-stage LoRA training performs better than frozen LM or multi-stage training
 - Note: This might change under other data or training conditions
- Our model enables 0-shot generalization and in-context learning
- Diverse acoustic conditions in SFT data are important for task generalization



THANK YOU FOR YOUR ATTENTION!