RETHINKING TEXT-FIRST PRETRAINING FOR AUDIO LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

How should a fixed pretraining budget be allocated when extending a language model from text to audio? Large Audio Language Models (LALMs) are often built by starting from a pretrained text-only language model and then adding audio tokens through continued pretraining. This text-first strategy reuses an already capable linguistic backbone, but it also commits part of the total training budget to a text-only objective before the model begins joint text–audio learning. Determining whether this sequential allocation is more effective than learning both modalities from the beginning requires a controlled comparison. We therefore compare text-first continued pretraining (CPT) with joint text–audio pretraining from scratch (SPT) under matched end-to-end training budgets. We conduct a controlled sweep over training budgets, text–audio mixture ratios, and model scales while holding the decoder-only architecture and data sources fixed across CPT and SPT. Across the evaluated settings, SPT generally achieves a better balance between text and audio performance than CPT. Our results suggest that CPT can be inefficient because part of the budget is spent optimizing a text-only objective before training shifts to the joint text–audio objective, while text replay further reduces the budget available for audio-side training. SPT's advantage, however, does not appear to arise from simple positive cross-modal transfer, as the two objectives can still interfere. Instead, the results suggest that allowing both objectives to shape the model throughout training contributes to a better balance between text and audio capabilities. This trend extends to a Mixture-of-Experts architecture and a speech-generation objective, and persists after supervised fine-tuning on speech question answering.

1 INTRODUCTION

The success of Large Language Models (LLMs) has established autoregressive next-token prediction as a standard training paradigm for generative models (Brown et al., 2020; Touvron et al., 2023; Gemma Team et al., 2025; Yang et al., 2025). Building on this paradigm, Large Audio Language Models (LALMs) jointly model text and audio tokens within a single sequence, enabling a single model to support diverse tasks such as text understanding, text generation, speech understanding, and speech generation. Many training curricula for LALMs follow a *text-first pretraining* strategy (Zhang et al., 2023; Défossez et al., 2024; Zeng et al., 2024; Chen et al., 2025; Nguyen et al., 2025; Huang et al., 2025; Sugiura et al., 2026). In this approach, an LLM is first pretrained on text and then undergoes continued pretraining on a mixture of text and audio tokens. The motivation behind this recipe is that a strong text backbone already encodes linguistic knowledge, sequence modeling ability, and general reasoning capabilities. Reusing such a pretrained LLM is therefore expected to provide an efficient and effective starting point for building LALMs, both in terms of token efficiency and downstream performance.

However, when text-backbone pretraining is included in a fixed end-to-end training budget, it is unclear whether allocating part of that budget to text-only pretraining is more effective than learning text and audio jointly from the beginning. Manakul et al. (2026) show that text-first initialization does not always improve LALM training. Their setup, however, targets an audio-centric model trained on a pretraining mixture of 95% speech data and 5% text-only data. While informative, this setting differs from many practical LALM scenarios where preserving text capability alongside audio adaptation is an important objective. Han et al. (2026) show in vision–language pretraining that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

early joint training can outperform delayed modality integration. It remains unclear whether this training-order effect extends to discrete-token LALMs and persists across training budgets, text–audio allocations, backbone choices, model scales, and model architectures. The role of training order becomes particularly relevant when available pretrained checkpoints do not match the desired architecture, model scale, or tokenizer, requiring the text backbone to be trained as part of the end-to-end training budget.

To address this gap, we systematically compare text-first continued pretraining (CPT) with joint text–audio pretraining from scratch (SPT) under matched end-to-end training budgets including text-backbone pretraining. For each CPT configuration, we construct an SPT counterpart with the same cumulative text and audio token allocation so that the comparison differs in training order. We vary the end-to-end training budget, cumulative text–audio allocation, text backbone, and model scale, and further examine sensitivity to the optimization strategy at the CPT transition. We additionally test whether the training-order effect extends to a Mixture-of-Experts architecture, a speech-generation objective, and downstream supervised fine-tuning.

Across the budget and scale sweeps, SPT retains its advantage over CPT, while transition-specific learning-rate tuning and the MoE architecture narrow the gap. CPT improves substantially with transition-specific learning-rate tuning, and the gap is smaller under a Mixture-of-Experts architecture, but the overall trend remains. The same trend extends to a speech-generation objective and persists after identical supervised fine-tuning on speech question answering.

Our main contributions are as follows.

- We systematically compare text-first CPT and SPT under matched end-to-end training budgets. Across different training budgets, model scales, and cumulative text–audio allocations, SPT generally achieves a better balance between text and audio performance than CPT. We separately analyze how CPT depends on the choice of text backbone.
- We examine when CPT can narrow the gap with SPT. CPT improves substantially with transition-specific learning-rate tuning, while the gap between CPT and SPT is also reduced under a Mixture-of-Experts architecture.
- We show that the training-order effect extends beyond speech understanding. The same overall trend is observed for speech generation, and SPT retains an advantage after identical supervised fine-tuning on speech question answering.
- We provide practical guidance on when each strategy is most appropriate. When text-backbone pretraining is part of a fixed end-to-end training budget, SPT is generally more token-efficient. When a strong pretrained text checkpoint already matches the desired architecture, model scale, and tokenizer, CPT remains a practical alternative because the text backbone does not need to be trained again.

2 MOTIVATION

Prior work has shown that a pretrained text model can provide an effective initialization for speech language modeling. TWIST (Hassid et al., 2023) shows that text-pretrained initialization can outperform random initialization when both models receive the same speech-token adaptation budget. However, TWIST replaces the text vocabulary with speech tokens and trains only on speech-token sequences, so it does not address unified text–audio modeling. VoxtLM (Maiti et al., 2024) further shows that text-pretrained initialization can benefit unified speech–text modeling, including text LM, speech LM, and ASR. However, it compares initialization strategies within a fixed multimodal training stage and does not account for the cost of pretraining the text backbone. It therefore does not answer whether text-first CPT is more token-efficient when text-backbone pretraining is included in the end-to-end training budget.

This distinction becomes important when the text backbone must be trained from scratch. In this case, text-only pretraining is not a free initialization but part of the end-to-end training budget. We therefore compare two training orders under the same cumulative text–audio token allocation: text-only pretraining followed by mixed CPT, and joint text–audio training from the beginning.

Although matched in cumulative text–audio token allocation, the two strategies differ in when the audio-side objective is introduced, which can lead to different training dynamics. In text-first CPT,

Table 1: Token counts of the text and audio training data.

Dataset	Audio	Text
FineWeb-Edu	–	99.660B
LibriSpeech	0.087B	0.010B
Common Voice	0.152B	0.014B
MLS English	4.020B	0.478B
GigaSpeech	0.900B	0.123B
VoxPopuli	0.047B	0.005B
People's Speech	0.531B	0.062B
Total	5.737B	100.352B

Table 2: Text evaluation benchmarks.

Benchmark	Shot	Metric
ARC-C	25	`acc`
HellaSwag	10	`acc`
WinoGrande	5	`acc`
PIQA	0	`acc`
OpenBookQA	0	`acc`

the model first forms representations under the text-only objective and must later integrate the audio-side objective while preserving text capability. Text replay used for this preservation also consumes part of the CPT-stage budget. In SPT, by contrast, the text-only and audio-side objectives influence representation formation from the beginning. We therefore compare text-first CPT and SPT under a matched end-to-end training budget that includes text-backbone pretraining, and examine how training order affects the final text–audio trade-off.

3 Experimental Setup

3.1 Task Formulation

We model text-only sequences and ASR sequences, consisting of audio tokens followed by transcript tokens, with a single decoder-only Transformer. Let L_t and L_a denote their autoregressive next-token losses, respectively; training minimizes $L_\lambda = \lambda L_t + (1-\lambda)L_a$, where λ controls the data mixture.

3.2 Model Architecture and Tokenizer

We use the publicly available Qwen2.5-1.5B architecture for reproducible experiments. For SPT, we randomly initialize the model and train it on text and audio from the beginning. For text-first CPT, we first train a text-only backbone and then continue training it on a text–audio mixture.

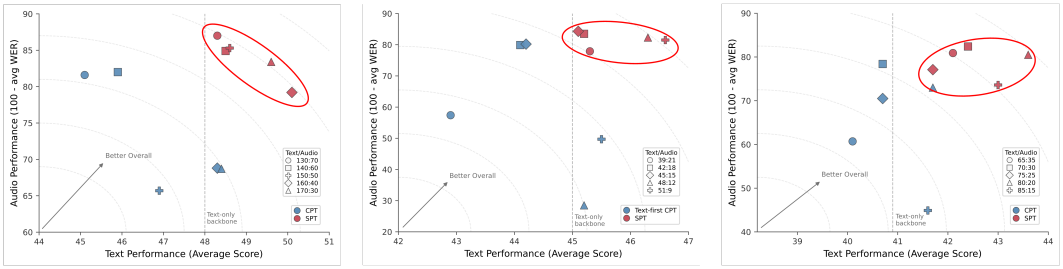
We use the Qwen2.5 tokenizer for text and CosyVoice 3 (Du et al., 2025) for audio. This provides a practical setup for building a discrete-token LALM by combining a text tokenizer with a strong discrete speech tokenizer. CosyVoice converts 16 kHz speech into discrete semantic audio tokens at 25 tokens per second. We build a unified Qwen–CosyVoice (QC) tokenizer by extending the Qwen2.5 vocabulary with audio-related tokens: two boundary tokens, `<boa>` and `<eoa>`, and 6,561 audio tokens from `<aud_0>` to `<aud_6560>`. The newly added audio-related token embeddings are randomly initialized from $\mathcal{N}(0, 0.02^2)$, and the input embedding matrix is tied with the LM head.

3.3 Training Dataset

We construct the training data from public text and audio corpora. For text, we use FineWeb-Edu 100B (Penedo et al., 2024)[1]. We construct the audio corpus by combining a broad set of large-scale publicly available English speech corpora: LibriSpeech (Panayotov et al., 2015), Common Voice (Ardila et al., 2020), MLS English (Pratap et al., 2020), GigaSpeech (Chen et al., 2021), VoxPopuli (Wang et al., 2021), and People's Speech (Galvez et al., 2021). All audio waveforms are tokenized with CosyVoice and mapped to the audio-token IDs of the QC tokenizer. We use only tokenized train splits; test and dev splits are excluded.

Table 1 reports the token counts. Under the QC tokenizer, FineWeb-Edu contains about 99.66B text tokens, while the audio corpus contains about 5.737B audio tokens and 0.692B transcript text tokens. The transcript tokens are used as part of the paired audio–text sequences, rather than as a separate text-only corpus.

[1] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu/tree/main/sample/100BT



(a) 67× text tokens per parameter

(b) 20× text tokens per parameter (Chinchilla)

(c) 0.5B model

Figure 1: Text–audio trade-offs between text-first CPT and SPT under different training budgets and model scales. Each point denotes a different cumulative text–audio token allocation. Blue curves denote text-first CPT, red curves denote SPT, dashed vertical lines indicate text-only baselines, and red ellipses highlight the SPT region for visual clarity.

During pretraining, we sample the text corpus and the audio corpus with a fixed mixture ratio. SPT samples from both corpora from the beginning, whereas text-first CPT first trains a text-only backbone and then continues training on the same text–audio mixture.

3.4 Evaluation

We evaluate the model by separately measuring text reasoning ability and speech understanding ability.

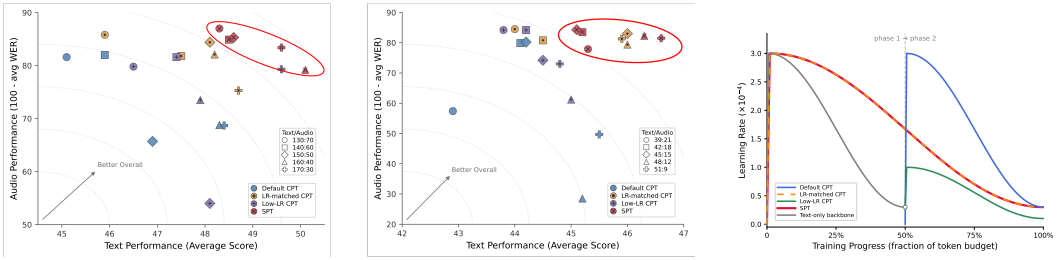
Text benchmarks. Text performance is evaluated with lm-evaluation-harness (Gao et al., 2024). We choose the benchmarks by referring to the evaluation protocols of public pretrained LLMs from the Qwen (Qwen et al., 2025), Gemma (Gemma Team et al., 2025), and Llama (Touvron et al., 2023) families. The benchmark suite focuses on commonsense and reasoning tasks. Table 2 summarizes the benchmarks, few-shot settings, and metrics. We report the final text performance as the average score over the five benchmarks.

Speech benchmarks. Audio performance is evaluated with AudioBench ASR tasks (Wang et al., 2025). Given a discrete audio token sequence, the model generates the corresponding transcript, and performance is measured by word error rate (WER) against the reference transcript. We use three English ASR test sets: LibriSpeech test-clean, Common Voice 15 English test, and GigaSpeech test. All speech benchmarks are evaluated in the zero-shot setting. We report the final audio performance as the average WER over the three ASR benchmarks. In the experimental results, we use $100-\mathrm{WER}$ as the audio performance score so that higher values indicate better performance.

4 Results

4.1 Comparing Pretraining Orders for Text–Audio Modeling

Figure 1a compares how CPT and SPT affect the final text–audio trade-off. In this experiment, we limit the end-to-end training budget to 200B tokens. Text-first CPT first trains a text-only backbone for 100B tokens and then uses the remaining 100B tokens for CPT on a text–audio mixture. For our 1.5B-parameter model, this backbone pretraining corresponds to approximately 67× text tokens per parameter. In contrast, SPT trains on text and audio together from the beginning for the full 200B-token budget. Here, T_s and A_s denote the numbers of text-only and paired ASR-sequence tokens used during from-scratch pretraining. In text-first CPT, only T_s is used to train the text-only backbone, whereas SPT uses both T_s and A_s. T_c and A_c denote the corresponding token counts used during the subsequent CPT stage. Text-first CPT starts from a $\mathrm{T}_s(100\mathrm{B})$ backbone, and we vary the T_c:A_c ratio within the 100B-token CPT stage from 3:7 to 7:3. For each CPT configuration, we construct an SPT counterpart with the same cumulative text and audio token counts. For example,



(a) 67× tokens per parameter. (b) 20× tokens per parameter (Chinchilla). (c) Learning-rate schedules.

Figure 2: Sensitivity to the learning-rate schedule at the CPT transition. (a–b) Transition tuning generally improves the CPT trade-off, although the effect varies across text–audio allocations and does not consistently make CPT superior to SPT. SPT achieves a better text–audio balance without stage-specific transition tuning. (c) Learning-rate schedules for Default CPT, LR-matched CPT, Low-LR CPT, and SPT.

$\mathrm{T}_s(100\mathrm{B}) + \mathrm{T}_c(30\mathrm{B})\mathrm{A}_c(70\mathrm{B})$ is compared with $\mathrm{T}_s(130\mathrm{B})\mathrm{A}_s(70\mathrm{B})$. Consequently, the cumulative text–audio allocations range from $13:7$ to $17:3$ for both training strategies.

The results show that starting from a strong text model does not guarantee a strong final text–audio trade-off. Text-first CPT does not form a stable frontier after audio is introduced. Although it can preserve some of the initial text capability, it has difficulty integrating audio while maintaining balanced performance across the two modalities.

By contrast, SPT achieves a better balance between text and audio performance, consistent with the hypothesis that exposing the model to both modalities from the beginning facilitates joint adaptation. Under the same 200B-token end-to-end training budget, SPT models lie on a better text–audio frontier than text-first CPT models. The results indicate that the final text–audio trade-off depends on training order even when CPT starts from a strong text backbone.

4.2 Robustness Across Training Budgets and Model Scales

Different training budgets. Figure 1b shows that the advantage of SPT persists with a less heavily pretrained text backbone. For our 1.5B model, 30B tokens of text-only pretraining correspond to 20× tokens per parameter, following the Chinchilla-style compute-optimal ratio (Hoffmann et al., 2022). We therefore consider a 60B-token end-to-end training budget: text-first CPT uses 30B tokens for text-only backbone pretraining and performs text–audio CPT for the remaining 30B tokens, whereas SPT trains jointly for the full 60B tokens. We vary the text–audio mixture ratio and compare the resulting trade-offs.

Smaller model scale. Figure 1c shows that the same trend persists at a smaller model scale. We repeat the comparison with a smaller model that follows the Qwen2.5-0.5B architecture[2]. Text-first CPT starts from a 50B-token text backbone and is followed by a 50B-token CPT stage, while SPT trains jointly for the same 100B-token end-to-end training budget. We again vary the text–audio mixture ratio and compare the final text–audio trade-off.

Together, these results suggest that the CPT–SPT difference is neither specific to the heavily pretrained 67× text backbone nor to the 1.5B model scale.

4.3 Sensitivity to the Optimization Strategy at the CPT Transition

In our default training setup (Appendix B), CPT treats text–audio adaptation as a new training stage following text-only pretraining. Because this stage introduces the audio-side objective together with newly initialized audio-token embeddings, we reinitialize the optimizer and restart the learning-rate schedule used for backbone pretraining. In contrast, SPT is trained continuously with a single

[2]https://huggingface.co/Qwen/Qwen2.5-0.5B

Table 3: Robustness across random seeds in the Chinchilla-style 60B-token setting. We report the mean and standard deviation across three random seeds for text performance and audio performance.

	CPT		SPT	
Text:Audio	Text	Audio	Text	Audio
39:21	43.07 ± 0.38	48.60 ± 10.83	$\mathbf{45.80 \pm 0.56}$	$\mathbf{82.97 \pm 4.40}$
42:18	44.00 ± 0.17	76.77 ± 8.07	$\mathbf{45.23 \pm 0.25}$	$\mathbf{84.63 \pm 1.00}$
45:15	44.70 ± 0.50	77.83 ± 6.48	$\mathbf{45.77 \pm 0.61}$	$\mathbf{84.43 \pm 0.32}$
48:12	45.13 ± 0.21	62.97 ± 29.92	$\mathbf{46.10 \pm 0.20}$	$\mathbf{81.97 \pm 0.42}$
51:9	45.57 ± 0.06	43.53 ± 19.40	$\mathbf{46.57 \pm 0.35}$	$\mathbf{80.80 \pm 0.61}$

optimizer state and learning-rate schedule throughout training. We therefore examine how these transition-specific optimization settings affect the CPT–SPT comparison.

We vary the CPT transition strategy across both backbone settings and compare Default CPT with two variants. As illustrated in Figure 2c, LR-matched CPT preserves the optimizer state and training iteration from text-only pretraining and continues along the corresponding SPT learning-rate trajectory. Low-LR CPT instead reinitializes the optimizer and restarts the schedule with a peak learning rate reduced to one-third of the default value, from 3×10^{-4} to 1×10^{-4}. Apart from these transition-specific choices, all other settings are unchanged.

As shown in Figures 2a and 2b, transition tuning generally improves the CPT trade-off, although the effect varies across text–audio allocations. CPT outperforms SPT at some text–audio allocations, but SPT achieves a better overall balance between text and audio performance. Thus, transition tuning can reduce the CPT–SPT gap, but does not remove the overall advantage of SPT. Importantly, an effective transition strategy is not known a priori and may require stage-specific hyperparameter search, introducing additional tuning and computational cost. In contrast, SPT uses a single optimizer state and learning-rate schedule throughout training without such transition-specific search. We also do not separately optimize the SPT training schedule, leaving open the possibility of further gains through additional optimization tuning.

4.4 Robustness Across Random Seeds

To examine whether the difference between CPT and SPT depends on a specific random initialization, we run additional experiments with different random seeds in the Chinchilla-style 60B-token setting. Given the computational cost of large-scale pretraining, we restrict the multi-seed analysis to this setting rather than repeating all experiments across multiple seeds. For each text–audio allocation, Table 3 reports the mean and standard deviation of text performance and audio performance across seeds. As in the main trade-off analysis, audio performance is computed as $100 - \mathrm{WER}$.

Overall, the main trend of the CPT–SPT comparison remains unchanged across random seeds. SPT achieves higher average text and audio performance than CPT for all text–audio allocations. In addition, SPT shows relatively small variation in audio performance across seeds for most allocations, while CPT shows larger variation when the audio ratio is low. These results indicate that the better balance between text and audio performance observed for SPT in the main experiments does not depend on a particular random seed.

4.5 Text Preservation Under Fixed-Ratio CPT Scaling

Prior work (Hsiao et al., 2025) shows that text replay mitigates catastrophic forgetting during sequential speech adaptation. We examine how text performance changes as the CPT-stage budget scales under a fixed T_c:A_c ratio. Motivated by Figure 1a, where T_c:$\mathrm{A}_c = 6$:4 provides a representative text-preserving configuration, we use this ratio as a controlled setting. As shown in Figure 3, text performance remains close to that of the text-only backbone across the evaluated CPT-stage scales, although the trajectory is not strictly monotonic. We do not interpret 6:4 as an optimal or minimum replay ratio, since the required balance may depend on the model, data distribution, tokenizer, and training recipe. Rather, the result shows that allocating 60% of the CPT-stage budget to text replay can preserve text performance, but leaves only 40% for audio-side training. This suggests that preserving text capability during CPT can itself introduce inefficiency, as part of the fixed training budget must be spent replaying text data rather than adapting to audio.

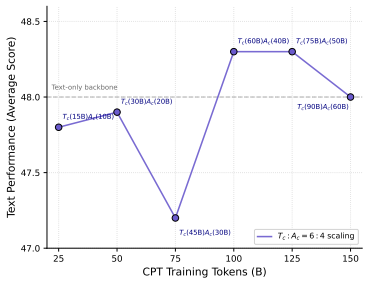


Figure 3: Text preservation as the CPT-stage budget scales at a fixed T_c:A_c = 6:4 ratio.

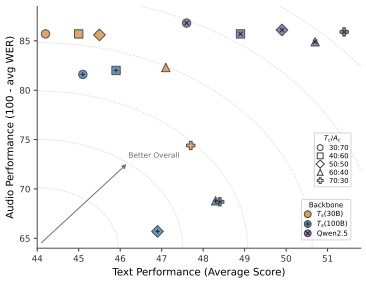
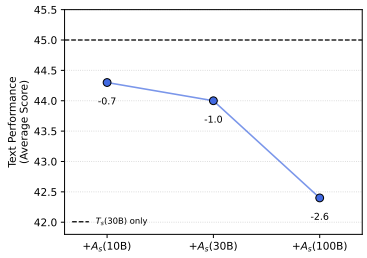
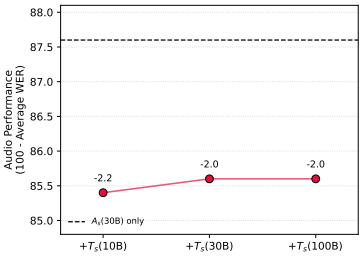


Figure 4: CPT text–audio trade-offs across $T_s(30B)$, $T_s(100B)$, and Qwen2.5 backbones.



(a) Adding A_s to T_s



(b) Adding T_s to A_s

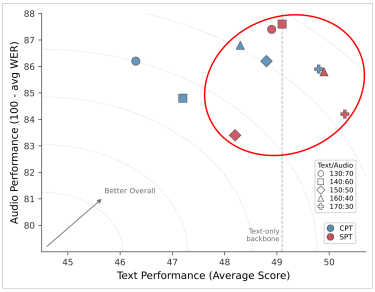
Figure 5: Cross-modal interference under SPT. We fix one modality at 30B tokens and add the other at different scales. Dashed lines denote the corresponding single-modality baselines, and the numbers indicate performance changes relative to them. Adding audio reduces text performance, while adding text reduces audio performance.

4.6 Effect of Text Pretraining Scale on CPT

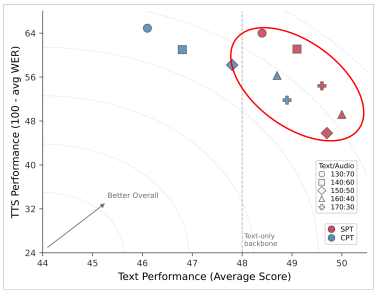
Text-first CPT assumes that additional text-only pretraining provides a better initialization for LALM training. To test this assumption, we compare our $\mathrm{T}_s(30\mathrm{B})$ and $\mathrm{T}_s(100\mathrm{B})$ backbones, which share the same architecture, text distribution, and training recipe, together with the released Qwen2.5 checkpoint as a practical initialization. For each backbone, we fix the CPT-stage budget to 100B tokens and vary T_c:A_c from 3:7 to 7:3. The in-domain backbones use the same text distribution as CPT replay, whereas Qwen2.5 uses a different pretraining distribution and recipe.

Figure 4 shows that the $\mathrm{T}_s(100\mathrm{B})$ backbone retains stronger text performance than $\mathrm{T}_s(30\mathrm{B})$, but does not consistently achieve better audio performance or a stronger text–audio frontier. Local measurements provide a possible explanation. The replay-text gradients of both backbones are below measurement resolution, so residual text gradients do not explain the CPT difference. However, $\mathrm{T}_s(100\mathrm{B})$ has higher text-side curvature along the audio-gradient direction, indicating a larger local text-loss penalty under the quadratic model in Appendix A.2. We treat this as a local diagnostic rather than a complete explanation; detailed measurements are provided in Appendix D.

Qwen2.5 forms a smoother and stronger frontier while exhibiting a non-negligible replay-text gradient and lower text-side curvature along the audio-gradient direction. These results suggest that CPT effectiveness depends on the relationship between the initialization and the CPT objectives, rather than on text pretraining quantity or initial text performance alone. Because Qwen2.5 also differs in pretraining scale, data distribution, and optimization recipe, however, the comparison does not isolate the cause of its stronger frontier. Our claims thus concern cases where the backbone must be trained, while CPT remains a practical option when a strong released checkpoint matches the desired architecture, scale, and tokenizer, although its effectiveness is difficult to predict from the checkpoint alone.



(a) Text–audio trade-off with the MoE architecture.



(b) Text–audio trade-off with the speech-generation objective.

Figure 6: Extensions of the training-order comparison across architectures and tasks.

4.7 Cross-Modal Interference Under SPT

SPT uses both modalities from the beginning of training. Therefore, audio data may help text modeling, or text data may help audio modeling. To test this possibility, we fix one modality at 30B tokens and measure how the performance of that modality changes when the other modality is added.

Figure 5 shows that positive transfer is not clearly observed in either direction. In the text-side experiment, we keep text training fixed at 30B tokens and add audio training at different scales. This decreases text performance. In the audio-side experiment, we keep audio training fixed at 30B tokens and add text training at different scales. This also decreases audio performance.

These results suggest that SPT's advantage does not primarily arise from positive cross-modal transfer, since the two modalities can still interfere. Instead, SPT exposes the model to both objectives throughout representation formation, whereas CPT introduces the audio-side objective only after text-specialized representations have formed.

5 Extensions Across Architectures and Tasks

We further examine whether the training-order effect generalizes across model architectures, speech-generation objectives, and downstream supervised adaptation.

5.1 Extension to a Mixture-of-Experts Architecture

To examine whether the training-order effect is specific to dense architectures, we repeat the comparison using a Mixture-of-Experts (MoE) model (Shazeer et al., 2017; Fedus et al., 2022) under the same tokenizer, training data, objective, token budget, and evaluation setting. The model uses 16 Transformer layers with 64 experts and top-4 routing, with approximately 5.62B total and 0.66B active parameters per token. The full configuration is provided in Table 8.

Figure 6a shows that SPT still achieves a better balance between text and audio performance than CPT, although the gap is smaller than in the dense model. One possible explanation is that sparse expert routing allows different subsets of parameters to specialize, thereby reducing interference between modalities. Prior work has shown that MoE sparsification can mitigate interference between modalities in multimodal models (Akbari et al., 2023), while modality-specific experts can emerge even without explicitly enforcing modality-dependent routing (Mustafa et al., 2022). Although we do not directly analyze expert specialization in our model, the reduced CPT–SPT gap is consistent with the possibility that sparse routing partially isolates text and audio updates. SPT nevertheless retains an overall advantage, suggesting that training order still matters under the MoE architecture.

5.2 Extension to Speech Generation

We further examine whether the training-order trend observed in speech understanding extends to speech generation. We convert the same paired training data used in the main experiments into a

Table 4: Speech question answering performance before and after LibriSQA fine-tuning (P1: ROUGE-L; P2: EM). Both models use the same 200B-token end-to-end training budget and final cumulative 17:3 text–audio allocation.

Task	Dataset	CPT	SPT	CPT-FT	SPT-FT
P1	LibriSQA P1	29.3	29.0	56.6	**57.2**
	PublicSG (Wang et al., 2025)	18.9	20.5	36.2	**37.3**
	SLUE-SQA5 (Shon et al., 2023)	2.9	9.8	14.7	**16.1**
	SpokenSQuAD (Lee et al., 2018)	3.3	5.4	11.2	**11.4**
P2	LibriSQA P2	0.1	0.1	57.6	**67.5**
	CN College (Hu et al., 2024)	0.0	0.1	35.5	**35.7**
	DREAM-TTS (Sun et al., 2019)	0.0	0.2	36.1	**40.3**

TTS format by reversing the original ASR sequence, such that the transcript text is followed by the corresponding audio token sequence.

For evaluation, the generated audio tokens are converted back into waveforms and transcribed using whisper-large-v3 (Radford et al., 2022). We compute WER between the resulting transcription and the original reference text. As in the ASR experiments, we report $100 - \mathrm{WER}$ as the TTS performance score.

Figure 6b shows the text–TTS trade-off between CPT and SPT. Text-first CPT achieves competitive TTS performance in the more audio-heavy configurations, but its TTS performance decreases as more training budget is allocated to text. In contrast, SPT maintains more stable TTS performance across the evaluated configurations and provides a better balance between text and TTS performance in the medium- and high-text-performance regions.

5.3 Extension to Speech Question Answering

Finally, we test whether the difference between SPT and CPT remains after downstream supervised fine-tuning. We compare $\mathrm{T}_s(100\mathrm{B}) + \mathrm{T}_c(70\mathrm{B})\mathrm{A}_c(30\mathrm{B})$ and $\mathrm{T}_s(170\mathrm{B})\mathrm{A}_s(30\mathrm{B})$, which have the same 200B-token end-to-end training budget and final 17:3 text–audio allocation. We fine-tune all parameters of both models on LibriSQA (Zhao et al., 2025) using the same training settings. For each backbone, we fine-tune separate models for Part I (P1), an open-ended generation task, and Part II (P2), a multiple-choice task. Further details are provided in Appendix F.

As shown in Table 4, fine-tuning substantially improves both models. SPT performs better on most evaluation sets before fine-tuning and on all seven sets afterward, with generally small differences on P1 but clearer differences on P2. Because both models use the same downstream data and training settings, these results suggest that the effect of pretraining order is not fully removed by supervised fine-tuning and can carry over to speech question answering.

6 Conclusion

In this paper, we showed that text-first CPT is not always the most token-efficient recipe for discrete-token LALMs when backbone training is included in the end-to-end training budget. Across the budget, scale, and text–audio allocation sweeps, SPT generally achieves a better text–audio balance than CPT. In CPT, preserving text capability can require substantial text replay, reducing the budget available for audio-side training. CPT can narrow the gap through transition-specific optimization tuning, and the gap becomes smaller with a Mixture-of-Experts architecture, but SPT achieves a favorable text–audio balance without stage-specific tuning. The same overall trend extends to speech generation and remains after identical supervised fine-tuning on speech question answering. When the text backbone must be trained as part of the end-to-end training budget, our results favor SPT. When a suitable pretrained text checkpoint is already available, however, CPT remains a practical alternative.

AI USE STATEMENT

Generative AI tools were used solely to provide feedback on the mathematical reasoning in Appendix A and to improve the clarity and readability of the manuscript. All AI-assisted feedback and edits were reviewed by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We use publicly available model architectures, tokenizers, datasets, and evaluation benchmarks. The model architecture, data construction, training objectives, and evaluation protocols are described in Section 3, with detailed training hyperparameters provided in Appendix B and downstream finetuning details in Appendix F.

REFERENCES

Hassan Akbari, Dan Kondratyuk, Yin Cui, Rachel Hornung, Huisheng Wang, and Hartwig Adam. Alternating gradient descent and mixture-of-experts for integrated multimodal perception. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=uTlKUAm68H.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4218–4222, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.520/.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech 2021*, 2021.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025. doi: 10.1109/TASLPRO.2025.3530270.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, Guanrou Yang, Yabin Li, Yanni Chen, Zhifu Gao, Qian Chen, Yue Gu, Mengzhe Chen, Yafeng Chen, Shiliang Zhang, Wen Wang, and Jieping Ye. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training, 2025. URL https://arxiv.org/abs/2505.17589.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue, 2024. URL https://arxiv.org/abs/2410.00037.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Daniel Galvez, Greg Diamos, Juan Torres, Keith Achorn, Juan Felipe Cerón, Anjali Gopi, David Kanter, Max Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *NeurIPS Datasets and Benchmarks*, 2021. URL http://dblp.uni-trier.de/db/conf/nips/neurips2021db.html#GalvezDTACGKLMR21.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena

Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Junlin Han, Shengbang Tong, David Fan, Minghao Chen, Philip Torr, Filippos Kokkinos, and Mike Lewis. Towards physics of multimodal pretraining: Knowledge flow, modality synergy, early unification, and recipes, 2026. URL https://arxiv.org/abs/2608.05000.

Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, Roy Schwartz, and Yossi Adi. Textually pretrained speech language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Chi-Yuan Hsiao, Ke-Han Lu, Kai-Wei Chang, Chih-Kai Yang, Wei-Chih Chen, and Hung yi Lee. Analyzing mitigation strategies for catastrophic forgetting in end-to-end training of spoken language models, 2025. URL https://arxiv.org/abs/2505.17496.

Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, Linquan Liu, and Furu Wei. WavLLM: Towards robust and adaptive speech large language model. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4552–4572, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.263. URL https://aclanthology.org/2024.findings-emnlp.263/.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, Peng Liu, Ruihang Miao, Wang You, Xi Chen, Xuerui Yang, Yechang Huang, Yuxiang Zhang, Zheng Gong, Zixin Zhang, Hongyu Zhou, Jianjian Sun, Brian Li, Chengting Feng, Changyi Wan, Hanpeng Hu, Jianchang Wu, Jiangjie Zhen, Ranchen Ming, Song Yuan, Xuelin Zhang, Yu Zhou, Bingxin Li, Buyun Ma, Hongyuan Wang, Kang An, Wei Ji, Wen Li, Xuan Wen, Xiangwen Kong, Yuankai Ma, Yuanwei Liang, Yun Mou, Bahtiyar Ahmidi, Bin Wang, Bo Li, Changxin Miao, Chen Xu, Chenrun Wang, Dapeng Shi, Deshan Sun, Dingyuan Hu, Dula Sai, Enle Liu, Guanzhe Huang, Gulin Yan, Heng Wang, Haonan Jia, Haoyang Zhang, Jiahao Gong, Junjing Guo, Jiashuai Liu, Jiahong Liu, Jie Feng, Jie Wu, Jiaoren Wu, Jie Yang, Jinguo Wang, Jingyang Zhang, Junzhe Lin, Kaixiang Li, Lei Xia, Li Zhou, Liang Zhao, Longlong Gu, Mei Chen, Menglin Wu, Ming Li, Mingxiao Li, Mingliang Li, Mingyao Liang, Na Wang, Nie Hao, Qiling Wu, Qinyuan Tan, Ran Sun, Shuai Shuai, Shaoliang Pang, Shiliang Yang, Shuli Gao, Shanshan Yuan, Siqi Liu, Shihong Deng, Shilei Jiang, Sitong Liu, Tiancheng Cao, Tianyu Wang, Wenjin Deng, Wuxun Xie, Weipeng Ming, Wenqing He, Wen Sun, Xin Han, Xin Huang, Xiaomin Deng, Xiaojia Liu, Xin Wu, Xu Zhao, Yanan Wei, Yanbo Yu, Yang Cao, Yangguang Li, Yangzhen Ma, Yanming Xu, Yaoyu Wang, Yaqiang Shi, Yilei Wang, Yizhuang Zhou, Yinmin Zhong, Yang Zhang, Yaoben Wei, Yu Luo, Yuanwei Lu, Yuhe Yin, Yuchu Luo, Yuanhao Ding, Yuting Yan, Yaqi Dai, Yuxiang Yang, Zhe Xie, Zheng Ge, Zheng Sun, Zhewei Huang, Zhichao Chang, Zhisheng Guan, Zidong Yang, Zili Zhang, Binxing Jiao, Daxin Jiang, Heung-Yeung Shum, Jiansheng Chen, Jing Li, Shuchang Zhou, Xiangyu Zhang, Xinhao Zhang, and Yibo Zhu. Step-audio: Unified understanding and generation in intelligent speech interaction, 2025. URL https://arxiv.org/abs/2502.11946.

Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. *Proc. Interspeech 2018*, pp. 3459–3463, 2018.

Soumi Maiti, Yifan Peng, Shukjae Choi, Jee-Weon Jung, Xuankai Chang, and Shinji Watanabe. Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13326–13330, 2024. doi: 10.1109/ICASSP48485.2024.10447112.

Potsawee Manakul, Woody Haosheng Gan, Martijn Bartelds, Guangzhi Sun, William Held, and Diyi Yang. Scaling open discrete audio foundation models with interleaved semantic, acoustic, and text tokens, 2026. URL https://arxiv.org/abs/2602.16687.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025. doi: 10.1162/tacl_a_00728. URL https://aclanthology.org/2025.tacl-1.2/.

nostalgebraist. Interpreting gpt: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. MLS: A Large-Scale Multilingual Dataset for Speech Research. In *Interspeech 2020*, pp. 2757–2761, 2020. doi: 10.21437/Interspeech.2020-2826.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Suwon Shon, Siddhant Arora, Chyi-Jiunn Lin, Ankita Pasad, Felix Wu, Roshan Sharma, Wei-Lun Wu, Hung-yi Lee, Karen Livescu, and Shinji Watanabe. SLUE phase-2: A benchmark suite of diverse spoken language understanding tasks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8906–8937, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.496. URL https://aclanthology.org/2023.acl-long.496/.

Issa Sugiura, Shuhei Kurita, Yusuke Oda, and Ryuichiro Higashinaka. Llama-mimi: Exploring the limits of flattened speech language modeling, 2026. URL https://arxiv.org/abs/2509.14882.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019. doi: 10.1162/tacl_a_00264. URL https://aclanthology.org/Q19-1014/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F Chen. Audiobench: A universal benchmark for audio large language models. *NAACL*, 2025.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 993–1003, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.80. URL https://aclanthology.org/2021.acl-long.80/.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot, 2024. URL https://arxiv.org/abs/2412.02612.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15757–15773, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1055. URL https://aclanthology.org/2023.findings-emnlp.1055/.

Zihan Zhao, Yiyang Jiang, Heyang Liu, Yu Wang, and Yanfeng Wang. Librisqa: A novel dataset and framework for spoken question answering with large language models. *IEEE Transactions on Artificial Intelligence*, 6(11):2884–2895, 2025. doi: 10.1109/TAI.2024.3427069.

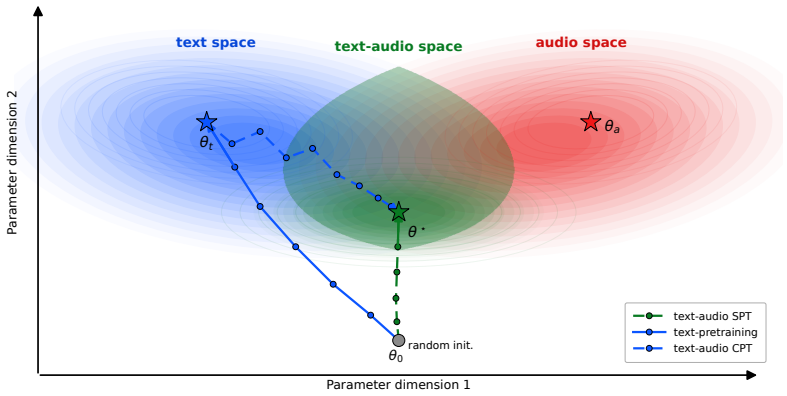


Figure 7: Conceptual illustration of how pretraining order may affect parameter movement under SPT and text-first CPT. SPT optimizes the mixed objective throughout training, whereas CPT first optimizes a text-only objective and introduces the audio-side objective later. The figure illustrates the hypothesis that this sequential objective change may require additional adaptation under a fixed end-to-end training budget; it does not depict measured optimization trajectories.

A Local Quadratic View of Pretraining Order

To isolate the effect of the expansion point, we consider an idealized local comparison in which SPT and the CPT stage are evaluated under the same mixed objective L_λ. This analysis abstracts away the different temporal mixture schedules used in our experiments, where the two strategies are matched in cumulative token allocation rather than in stage-wise mixture ratio.

A.1 SPT: Joint Adaptation During Representation Formation

SPT optimizes

$$L_\lambda^{\mathrm{SPT}}(\theta) = \lambda L_t(\theta) + (1-\lambda) L_a(\theta).$$

Let θ_s be a parameter point reached during joint text–audio training. Using the local coordinate $\theta = \theta_s + \delta$, both losses generally have first-order terms:

$$\begin{aligned} L_t(\theta_s+\delta) &\approx L_t(\theta_s) + g_t^{s\top}\delta + \frac{1}{2}\delta^\top H_t^s \delta, \qquad g_t^s = \nabla L_t(\theta_s), \quad H_t^s = \nabla^2 L_t(\theta_s), \\ L_a(\theta_s+\delta) &\approx L_a(\theta_s) + g_a^{s\top}\delta + \frac{1}{2}\delta^\top H_a^s \delta, \qquad g_a^s = \nabla L_a(\theta_s), \quad H_a^s = \nabla^2 L_a(\theta_s). \end{aligned} \tag{1}$$

Dropping constants independent of δ, the local quadratic surrogate is

$$Q_\lambda^{\mathrm{SPT}}(\delta) = \left[\lambda g_t^s + (1-\lambda) g_a^s\right]^\top \delta + \frac{1}{2}\delta^\top \left[\lambda H_t^s + (1-\lambda) H_a^s\right] \delta. \tag{2}$$

Assuming that the mixed Hessian is locally invertible, its stationary point is

$$\delta_{\mathrm{SPT}}^\star = -\left[\lambda H_t^s + (1-\lambda) H_a^s\right]^{-1} \left[\lambda g_t^s + (1-\lambda) g_a^s\right]. \tag{3}$$

For the direction-wise interpretation, we use a local shared-basis approximation in which the Hessians of the two objectives are approximately diagonal in the same basis. Under the shared-basis approximation, the i-th component becomes

$$\delta_{\mathrm{SPT},i}^\star = -\frac{\lambda g_{t,i}^s + (1-\lambda) g_{a,i}^s}{\lambda h_{t,i}^s + (1-\lambda) h_{a,i}^s}. \tag{4}$$

This direction-wise form shows that the local stationary displacement for SPT is jointly determined by the text-only and audio-side gradients, while its magnitude is scaled by the corresponding mixed curvature. Thus, both objectives influence which directions are favored during joint representation formation.

A.2 TEXT-FIRST CPT: AUDIO ADAPTATION AROUND A TEXT-ONLY SOLUTION

Text-first CPT starts from a text-only pretrained backbone, denoted by θ_t. Since this backbone has already been optimized for the text-only objective, we locally treat it as approximately stationary:

$$\nabla L_t(\theta_t) \approx 0. \tag{5}$$

Using the local coordinate $\theta = \theta_t + \delta$, Eq. 5 removes the text first-order term, while the first-order term of the audio-side objective remains because this objective has not been optimized at θ_t. Applying the same quadratic approximation as in Section A.1 to the mixed objective gives the direction-wise stationary point

$$\delta^\star_{\mathrm{CPT},i} = -\frac{(1-\lambda)g_{a,i}}{\lambda h_{t,i} + (1-\lambda)h_{a,i}}. \tag{6}$$

Here, $g_{a,i}$ is the audio-side gradient component, while $h_{t,i}$ and $h_{a,i}$ are the text-side and audio-side curvature components at θ_t, respectively. Thus, CPT movement is driven by $g_{a,i}$, but filtered by the mixed curvature $\lambda h_{t,i} + (1-\lambda)h_{a,i}$. In directions with large positive text curvature $h_{t,i}$, the local stationary displacement induced by the audio-side objective can be smaller even when $g_{a,i}$ is large.

This local expression describes the early stage of CPT around the text-only solution θ_t. If the audio fraction is large, CPT may move beyond this local neighborhood, making the text gradient non-negligible again. In that regime, text preservation and audio adaptation can directly compete, which may contribute to less stable optimization under a fixed end-to-end training budget.

A.3 IMPLICATIONS

To isolate the effect of the expansion point, Eq. 4 and Eq. 6 compare SPT and CPT under the same local mixed objective L_λ for a fixed λ. Under this controlled approximation, the SPT stationary displacement is evaluated around a point reached during joint text–audio training, where both the text-only and audio-side gradients contribute. In contrast, the CPT stationary displacement is evaluated around an approximately text-stationary pretrained point, where first-order adaptation is driven by the audio-side gradient and filtered by the mixed curvature. This comparison isolates how delaying the introduction of the audio-side objective changes the local geometry from which adaptation begins. The full training runs also differ in their temporal mixture schedules: SPT uses a fixed mixture corresponding to the target cumulative allocation throughout training, whereas CPT first performs text-only pretraining and then uses a stage-specific mixture during the CPT stage. We therefore interpret this analysis as an idealized local view of the expansion-point effect rather than a complete model of the empirical training dynamics.

Despite this idealization, the local view suggests two potential disadvantages of CPT. First, as conceptually illustrated in Figure 7, CPT may require redirecting parameters from a text-only solution toward a region that better supports the mixed objective. If the audio-side fraction is large, the optimization trajectory may leave the local neighborhood of the text-only solution, causing the text-only and audio-side gradients to compete more directly. Second, part of the end-to-end training budget is spent before the audio-side objective is introduced, leaving fewer updates under the mixed objective than in SPT, where both objectives are present throughout training. We examine whether the empirical results are consistent with these hypotheses in Section 4.

B TRAINING DETAILS

Unless otherwise noted, all experiments use the same optimization and batching settings. We vary only the model size, data mixture, and end-to-end training budget, as described in the main text. In our experiments, the end-to-end training budget refers to the total number of training tokens processed by the model. When the target training budget exceeds the size of a corpus, we continue sampling from the corpus until the target budget is reached. Table 5 summarizes the shared hyperparameters. The number of warmup iterations is approximately 1% of the iterations in each training stage.

Table 5: Common training hyperparameters used across experiments.

Optimization hyperparameters

Hyperparameter	Value
Optimizer	AdamW
β_1, β_2	$0.9, 0.95$
Weight decay	0.1
Gradient clipping	1.0
Peak learning rate	3×10^{-4}
Minimum learning rate	3×10^{-5}
LR schedule	Cosine decay
Warmup	1% of iterations
Precision	BF16
Dropout	0.0

Batching and infrastructure

Hyperparameter	Value
Sequence length	2,048
Global batch size	256
Tokens per step	524,288
Micro-batch size	8
Tensor parallelism	1
Pipeline parallelism	1

For CPT, we initialize the second stage from the text-only backbone but reinitialize the optimizer and restart the learning-rate schedule. Since text–audio adaptation introduces the audio-side objective together with newly initialized audio-token embeddings, we treat it as a new training stage and apply the same learning-rate schedule used for backbone pretraining. In contrast, SPT is trained as a single continuous run with one optimizer state and learning-rate schedule. Sensitivity to this stage-transition schedule is examined in Section 4.3.

C Evaluation Benchmarks

C.1 Text Benchmarks

ARC-C (Clark et al., 2018) is the challenge split of the AI2 Reasoning Challenge, a multiple-choice question-answering benchmark built from grade-school science exam questions. It contains questions that are not correctly answered by simple retrieval-based or word co-occurrence methods, making it a more difficult benchmark for evaluating scientific knowledge and reasoning.

HellaSwag (Zellers et al., 2019) is a commonsense inference benchmark where the model must choose the most natural continuation of a given situation. It is constructed with adversarial filtering, which selects machine-generated but plausible wrong endings. As a result, the task is easy for humans but difficult for models that rely only on shallow surface patterns.

WinoGrande (Sakaguchi et al., 2019) is a commonsense reasoning benchmark based on the Winograd Schema Challenge. Each example gives a sentence with a blank and two candidate answers, and the model must choose the option that best fits the sentence. The dataset is built at a large scale and uses adversarial filtering to reduce dataset biases, making it harder for models to solve by using simple word associations.

PIQA (Bisk et al., 2020) is a physical commonsense reasoning benchmark that tests whether a model understands everyday interactions with the physical world. Each example presents a goal or situation and two possible solutions, and the model must choose the more physically plausible option. The task is designed to evaluate commonsense knowledge that is often not explicitly written in text, such as how objects are used and how physical actions work.

OpenBookQA (Mihaylov et al., 2018) is a multiple-choice science question-answering benchmark modeled after open-book exams. The model is given a question and answer choices, rather than a long supporting passage. Thus, the task evaluates whether the model can apply elementary science knowledge and commonsense reasoning beyond simple word matching.

C.2 Speech Benchmarks

LibriSpeech (Panayotov et al., 2015) is an English ASR benchmark built from public-domain audiobooks from the LibriVox project. It contains about 1,000 hours of 16 kHz read speech, with carefully segmented and aligned transcripts. Because it provides clean read speech and standard train/dev/test splits, it is widely used for evaluating speech recognition systems.

Common Voice (Ardila et al., 2020) is a multilingual speech corpus designed for speech technology research, especially automatic speech recognition. In this work, we use only the English subset of Common Voice. The corpus is built through crowdsourcing, where volunteers record sentences and other users validate the transcriptions, resulting in diverse speakers, accents, and recording conditions.

MLS English (Pratap et al., 2020) is the English subset of the Multilingual LibriSpeech corpus, a large-scale speech corpus built from public-domain audiobooks. It provides read speech with aligned transcripts and is designed for training and evaluating ASR systems at larger data scales. Because it contains much more audiobook speech than the original LibriSpeech corpus, it is useful for measuring ASR performance in a large read-speech setting.

GigaSpeech (Chen et al., 2021) is a large-scale English ASR corpus designed for speech recognition training and evaluation. It contains 10,000 hours of high-quality transcribed speech collected from diverse sources such as audiobooks, podcasts, and YouTube. Because it covers both read and spontaneous speech across multiple domains, it is useful for evaluating ASR robustness beyond clean read-speech settings.

VoxPopuli (Wang et al., 2021) is a large-scale multilingual speech corpus built from European Parliament event recordings. It provides transcribed speech across many European languages, and in this work we use only the English subset. Because it contains real parliamentary speech, it is useful for evaluating ASR performance on formal but naturally recorded speech.

People's Speech (Galvez et al., 2021) is a large-scale English speech corpus designed for ASR training and evaluation. It is collected from diverse public-domain and permissively licensed speech sources, with transcripts aligned to the audio. Because it covers a broad range of speakers, topics, and recording conditions, it is useful for evaluating ASR robustness in diverse real-world speech settings.

D Local Gradient and Curvature Analysis

We further analyze the backbone checkpoints used in Section 4.6 by measuring the replay-text gradient, the audio-side gradient, and the text-side curvature along the audio-gradient direction. To estimate the replay-text gradient norm, we separate the mean-gradient signal from minibatch sampling noise using 256 independently sampled minibatches. The resolution floor denotes the smallest gradient norm that can be reliably distinguished from minibatch sampling noise under this estimator. All checkpoints are evaluated on identical token batches with the replay-text loss restricted to the shared text vocabulary. For the curvature measurement, we estimate the mean audio gradient over 64 minibatches and use a symmetric finite difference in FP32.

We define the text-side curvature along the audio-gradient direction as

$$\kappa_{t \leftarrow a} = \frac{g_a^\top H_t g_a}{\|g_a\|^2},$$

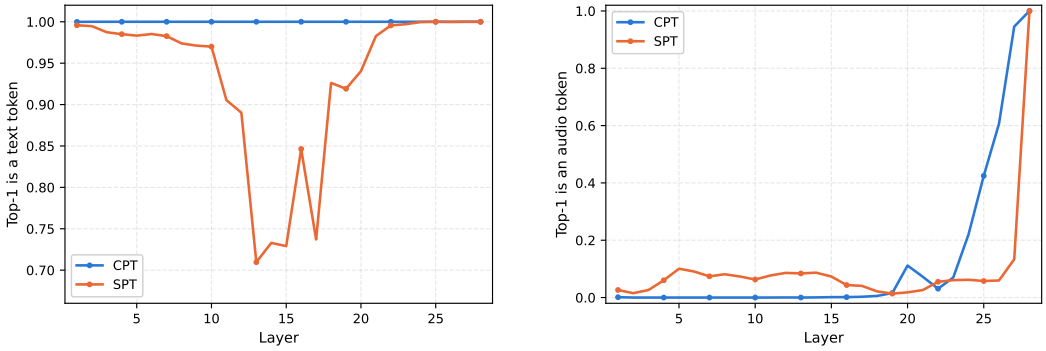
where $g_a = \nabla L_a$ and $H_t = \nabla^2 L_t$. A larger value indicates greater replay-text curvature along the direction preferred by the audio-side objective and, under the local quadratic model, may reduce the corresponding local stationary displacement.

As shown in Table 6, the replay-text gradients of both $\mathrm{T}_s(30\mathrm{B})$ and $\mathrm{T}_s(100\mathrm{B})$ lie below the corresponding resolution floor, while the Qwen2.5 gradient is clearly resolved. Thus, residual replay-text gradients do not reliably distinguish the two in-domain backbones. By contrast, $\mathrm{T}_s(100\mathrm{B})$ exhibits higher text-side curvature along the audio-gradient direction than $\mathrm{T}_s(30\mathrm{B})$. Under the local model, this difference is qualitatively consistent with a smaller local audio-directed displacement.

Qwen2.5 has a non-negligible replay-text gradient and the lowest text-side curvature along the audio-gradient direction. These measurements are qualitatively consistent with its smoother and stronger CPT frontier, although they remain local diagnostics and do not identify which aspect of its pretraining scale, data distribution, or optimization recipe is causally responsible.

Table 6: Local gradient and curvature measurements at the backbone checkpoints under matched audio-vocabulary expansion. The replay-text gradients of the two in-domain backbones are below the measurement resolution, whereas the Qwen2.5 gradient is resolved.

Backbone	$\|\nabla L_t\|$	Resolution floor	$\|\nabla L_a\|$	$\kappa_{t\leftarrow a}$
$\mathrm{T}_s(30\mathrm{B})$	0.107	0.199	90.5	1.26
$\mathrm{T}_s(100\mathrm{B})$	0.184	0.285	80.3	2.10
Qwen2.5	1.279	0.395	47.8	0.50



(a) Transcript-text region

(b) Audio-token region

Figure 8: Logit-lens analysis of layer-wise modality alignment in CPT and SPT. For each intermediate layer, we project the hidden states through the LM head and measure the fraction of positions whose top-1 prediction belongs to the target-modality vocabulary. Both CPT and SPT are trained with the same final cumulative allocation of 170B text tokens and 30B audio tokens.

E Layer-Wise Modality Alignment Through the Logit Lens

We perform a logit-lens analysis to examine how the layer-wise representations differ between SPT and CPT. The logit lens applies the LM head to the hidden state of each intermediate layer and interprets the resulting logits as a vocabulary distribution (nostalgebraist, 2020). Following Wendler et al. (2024), who analyze layer-wise language behavior in multilingual LLMs, we extend this analysis to the text–audio setting. We compare a CPT model, $\mathrm{T}_s(100\mathrm{B}) + \mathrm{T}_c(70\mathrm{B})\mathrm{A}_c(30\mathrm{B})$, and an SPT model, $\mathrm{T}_s(170\mathrm{B})\mathrm{A}_s(30\mathrm{B})$, which have the same cumulative text–audio allocation. We perform the analysis on the ASR test set, and Figure 8 reports the average results over the full test set.

An ASR sequence consists of an audio-token region followed by a transcript-text region. For each layer, we project the hidden states through the LM head. In the audio-token region, we measure the fraction of positions whose top-1 prediction belongs to the audio vocabulary. In the transcript-text region, we measure the fraction whose top-1 prediction belongs to the text vocabulary. This allows us to compare how each region becomes aligned with its target modality across layers.

As shown in Figure 8, the main difference between CPT and SPT appears in the transcript-text region. CPT keeps its top-1 predictions in the text vocabulary across almost all intermediate layers. In contrast, SPT shows high text alignment in the early layers, lower text alignment in the middle layers, and then returns to the text vocabulary in the later layers. This suggests that the intermediate-layer states of CPT remain more strongly aligned with the text vocabulary, while this alignment becomes weaker in the middle layers of SPT.

In the audio-token region, CPT and SPT show a similar overall pattern. Both models remain mostly text-oriented in the early and middle layers, and then rapidly shift toward the audio vocabulary in the later layers. SPT shows a slightly higher audio top-1 fraction than CPT in some middle layers, but the overall layer-wise transition is similar for the two models.

These results are qualitatively consistent with the view that representations formed during text-only pretraining can remain strongly reflected in the intermediate layers after CPT. In contrast, SPT

Table 7: Examples of LibriSQA Part I and Part II used for supervised fine-tuning. For readability, we show the transcript of the speech input; the model receives only the corresponding discrete audio tokens, not the transcript. Both examples are constructed from the same LibriSpeech utterance.

Task	Example
Part I	**Audio content:** Giving a great slap of one hand on her back, the room was light and bright and roused Mary. Perhaps the slap on her back might help a little too, and she felt the awkwardness of accounting for her presence to a little bustling old woman. **Question:** What action startled Mary and made the room feel lively? **Answer:** Giving a great slap of one hand on her back.
Part II	**Audio content:** Giving a great slap of one hand on her back, the room was light and bright and roused Mary. Perhaps the slap on her back might help a little too, and she felt the awkwardness of accounting for her presence to a little bustling old woman. **Question:** What effect did the slap on Mary's back have on her? **Choices:** (A) It made her feel light and bright. (B) It helped her become more awake and alert. (C) It made her feel awkward. (D) It made her feel a little embarrassed. **Answer:** (B) It helped her become more awake and alert.

shows a different layer-wise alignment pattern in the middle layers. We do not claim that these representation differences directly cause the stronger text–audio trade-off of SPT. Rather, the logit-lens analysis provides complementary evidence that the two training recipes differ not only in final text–audio performance, but also in their internal layer-wise modality organization, complementing the trade-off results in Section 4.1.

F Speech Question Answering Fine-Tuning Details

We fine-tune both backbones on LibriSQA (Zhao et al., 2025). Each example consists of a speech passage represented as discrete audio tokens, followed by a text question and a text answer. LibriSQA contains two formats: Part I (P1), an open-ended generation task, and Part II (P2), a multiple-choice task. We train separate models for the two formats. Table 7 shows representative examples constructed from the same LibriSpeech utterance.

For both backbones, we fine-tune all model parameters using AdamW with a peak learning rate of 2×10^{-5}, 3% warmup followed by cosine decay, a global batch size of 128, and a maximum sequence length of 1,536. The audio and question tokens are masked from the loss, and the causal language modeling loss is applied only to the answer and end-of-sequence tokens. All models are trained for two epochs.

For evaluation, we follow the SpeechQA protocol of AudioBench (Wang et al., 2025). We evaluate the P1 models on the LibriSQA P1 test set, PublicSG (Wang et al., 2025), SLUE-SQA5 (Shon et al., 2023), and SpokenSQuAD (Lee et al., 2018). We evaluate the P2 models on the LibriSQA P2 test set, CN College (Hu et al., 2024; Wang et al., 2025), and DREAM-TTS (Sun et al., 2019). P1 is evaluated with ROUGE-L, whereas P2 is evaluated with exact match. Across all evaluation sets, the speech input provides the passage or dialogue and the question is given as text.

Table 8: Configuration of the MoE model with 64 experts and top-4 routing. The model contains approximately 5.62B total parameters, of which 0.66B are active per token.

Component	**Value**
Backbone	
Number of layers	16
Hidden size	1,536
Number of attention heads	12
Number of KV heads (GQA)	2
Sequence length	2,048
Vocabulary size	158,272
MoE	
Number of experts	64
MoE top-k	4
MoE FFN hidden size	1,120
MoE auxiliary loss coefficient	1×10^{-2}
Parallelization and Scale	
Parallelization	TP=1 / PP=1 / EP=8
Scale	active $\sim$0.66B / total $\sim$5.62B