LATENT SPEECH-TEXT TRANSFORMER

Anonymous authors
Paper under double-blind review

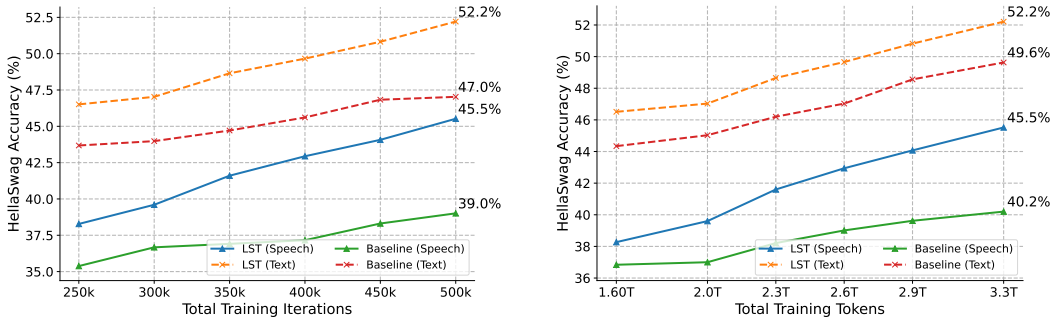
ABSTRACT

Auto-regressive speech-text models are typically pre-trained on a large number of interleaved sequences of text tokens and raw speech encoded as speech tokens using vector quantization. These models have demonstrated state-of-the-art performance in speech-to-speech understanding and generation benchmarks, together with promising scaling laws, primarily enabled by the representational alignment between text and speech. Nevertheless, they suffer from shortcomings, partly owing to the disproportionately longer sequences of speech tokens in contrast to textual tokens. This results in a large compute imbalance between modalities during pre-training as well as during inference, and a potential hindrance to effectively aligning speech and text, ultimately translating to several orders of magnitude slower scaling laws. We introduce the Latent Speech-Text Transformer (LST), which makes pre-training speech-text models more data-efficient by dynamically and inexpensively aggregating speech tokens into latent speech patches. These patches serve as higher-level units that can either align with corresponding textual units to aid capability transfer or even encapsulate common speech sequences like silences to be more compute-efficient. We show that LST outperforms vanilla approaches on speech-to-speech as well as text-to-text benchmarks in both data- and compute-controlled settings, the former indicating more effective representational alignment and the latter indicating steeper scaling laws for speech-text models. On HellaSwag story completion, LST achieves 6.5% absolute gain in speech accuracy under compute-controlled training and 5.3% under data-controlled training, while also improving text performance. We will release our models, code, and the evaluation data to facilitate further research.

1 INTRODUCTION

Inspired by the strong zero- and few-shot understanding and generation capabilities of large auto-regressive textual language models with billions of parameters that are pre-trained on trillions of tokens, Lakhotia et al. (2021) introduce the task of Generative Spoken Language Modeling (GSLM) a.k.a Textless NLP, where raw speech is encoded as a sequence of discrete tokens based on a dictionary of quantized speech features, and an auto-regressive language model (LM) is trained on these tokens with Next Token Prediction (NTP). While initially successful, Cuervo & Marxer (2024) estimate that this approach would require up to three orders of magnitude more data to obtain equivalent capabilities as textual LLMs, largely owing to the same information requiring a significantly larger number of speech tokens to represent compared to text. This increased sequence length also means that these models utilize considerably more compute during inference to process the same amount of semantic content compared to text.

To improve scaling properties of large speech models by taking advantage of the comparatively larger corpus of web text compared to speech, recent efforts have leveraged transfer learning from textual modalities in the form of warm initialization from large pre-trained text models (Hassid et al., 2023), pre-training with interleaved speech-text data (Nguyen et al., 2025), and modeling speech and text in multiple streams to leverage the textual chain of thought or "inner monologue" (Défossez et al., 2024). All these works attempted to some extent to achieve *representational alignment* between text and speech, where a perfect alignment means the model can treat the two modalities interchangeably without any performance difference. Despite this, there remains a large gap between text-to-text and speech-to-speech performance on the same benchmarks, highlighting the incompleteness of the alignment. We hypothesize that the severe mismatch in information densities between the speech and text tokens is one of the primary factors hindering speech-text alignment.



(a) *Compute-controlled*: fixed training iterations

(b) *Data-controlled*: fixed data budget

Figure 1: Comparison of LST and Baseline on HellaSwag story completion under two experimental setups, (a) *compute-controlled*: same number of training iterations and (b) *data-controlled*: same amount of training data.

To overcome the aforementioned challenges, we introduce the Latent Speech-Text Transformer (LST) based on the byte-latent transformer (BLT) architecture (Pagnoni et al., 2024), comprising an encoder that dynamically groups sequences of speech tokens into higher-level speech patches, a global speech-transformer that auto-regressively models interleaved sequences of textual tokens and speech patches, and a light-weight transformer decoder (Vaswani et al., 2017) that maps patches back into speech tokens of dynamic sizes. Working in terms of speech patches allows the model to encode more content given the same training cost, makes inference more efficient. These speech patches can represent higher-level speech concepts or prolonged silences, and serve to level the information density between speech and text, thus making them easier to align (see Figure 1).

We first demonstrate that LST models with fixed-size speech patching schemes similar to what Yu et al. (2023) did with text, are able to significantly outperform their non-patching counterparts. Such models are aware of the internals of patches without expending much compute in the process, in contrast with methods that expand the speech token vocabulary by applying subword tokenization, which yield poor downstream performance (Cuervo & Marxer, 2024). We further improve the performance by introducing speech-patching based on textual alignment at the word/subword levels, which crucially also includes patching large sequences of silences. Since this approach requires text-speech alignment timestamps during training and inference, we also introduce a curriculum-based method to eliminate the need for such alignments during inference.

To summarize, this paper makes the following contributions:

(1) We show improved performance of LST models in both data- and compute-controlled settings compared to vanilla interleaved speech-text models like SpiritLM (Nguyen et al., 2025), as well as models that use subword tokenization, on speech versions of popular text understanding benchmarks such as HellaSwag (Zellers et al., 2019). LST-based models save considerable training and inference compute and improve speech-text representational alignment (see Figure 1).

(2) We introduce different variations of speech patching schemes, including fixed-size static patching and alignment-based patching, and analyze their effectiveness.

(3) We demonstrate that LST continues outperforming the baseline when scaling up the model size from 1B to 7B parameters, which highlights the scalability of our method.

2 Background

Generalized spoken language models (Lakhotia et al., 2021) typically comprise three components: (1) a speech tokenizer model that maps a raw speech waveform s to a sequence of speech tokens $\{s_0, \ldots, s_n\}$, (2) a decoder-only transformer model (Vaswani et al., 2017) with parameters θ that models the distribution of the next speech token given the previous context i.e. $p_\theta(s_i|s_{<i})$, and (3) a vocoder model that maps speech token sequences back to a speech waveform, such as HiFi-GAN (Kong et al., 2020).

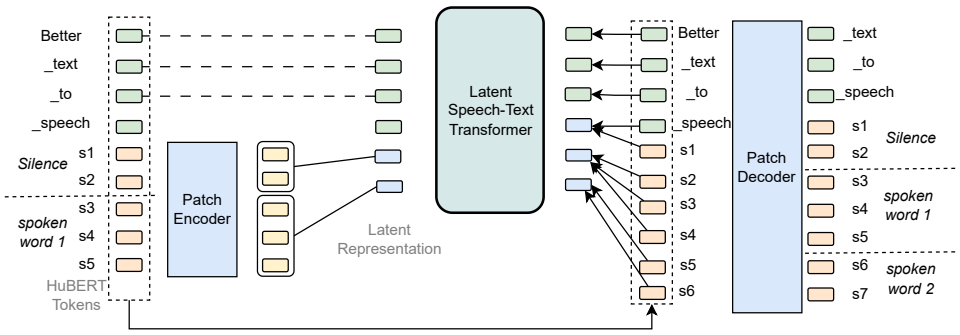


Figure 2: **Latent Speech-Text Transformer (LST).** The model encodes BPE text tokens and HuBERT speech tokens into a shared latent space. A *Patch Encoder* compresses local speech segments into patch representations, which are jointly processed with text tokens. A *Patch Decoder* predicts future speech tokens from latent representations, enabling alignment and transfer across modalities.

Speech tokenization Approaches for speech tokenization include semantic tokens represented by cluster-ids obtained by k-means clustering of frame representations as in Hubert (Hsu et al., 2021), acoustic tokens obtained as discretized embeddings from residual-vector quantization bottlenecks from self-supervised neural codec models (Zeghidour et al., 2021; Défossez et al., 2024), as well as additional tokens for expressivity and also, combinations of different token categories. In this paper, we follow Hassid et al. (2023); Nguyen et al. (2025) and use Hubert tokens using a codebook of 501 speech tokens at 25Hz. Unlike Nguyen et al. (2025) we do not need to deduplicate Hubert tokens as this is organically handled by the LST architecture.

Sequence Modeling Similar to LLMs for text, speech token modeling is typically done using a large transformer decoder model using causal self-attention, to maximize the likelihood of sequences from a large speech pre-training corpus ($\mathcal{D}$) in an auto-regressive fashion:

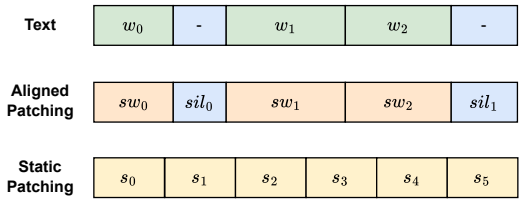
$$\mathcal{L}(\mathcal{D};\theta) = \sum_{s\in\mathcal{D}} \sum_{i} \log p_\theta(s_i|s_{<i}) \tag{1}$$

Interleaved Data Since speech sequences are longer and less compact that their corresponding text sequences, such models can require several orders of magnitude more data in order to achieve performance comparable to text models (Cuervo & Marxer, 2024). In order to bridge the gap, Nguyen et al. (2025) find that training on interleaved sequences of text and speech data directly correlates with improved performance. For a subset of the pre-training dataset that contains the textual sequence $\{t_0, \ldots, t_m\}$, where text tokens are obtained using a tokenizer (we use the Llama 2 tokenizer (Touvron et al., 2023) in this paper) and each text token can correspond to a span of speech tokens, the model is trained on an interleaved sequence obtained by replacing arbitrary spans of speech tokens in the sequence sequence with text tokens separated by special modality tokens. This allows the same model to be used for S→S, S→T, T→S and T→T tasks. We discuss the process of producing interleaved data from parallel text-speech data in Appendix A.1.

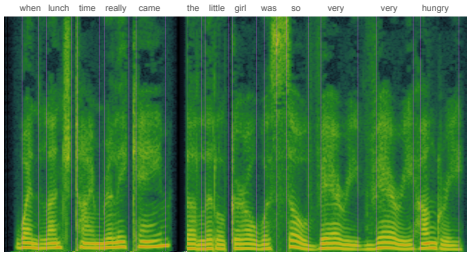
3 Latent Speech-Text Transformers

The core idea of the LST architecture is to auto-regressively model latent patches of tokens (using a global transformer), rather than individual tokens, similar in spirit to BLT (Pagnoni et al., 2024) which models dynamic-sized patches of bytes. The transformation of speech/text spans to patches and vice-versa takes place with the help of a light-weight local encoder and local decoder, and the entire model is trained end-to-end using the same token-level likelihood as before. Figure 2 illustrates this architecture specialized to the task of speech-text modeling. The majority of the compute expended in terms of FLOPs is in the global transformer, which yields savings by operating on information-dense speech patches instead of granular speech tokens.

Local Encoder. Similar to BLT, the local encoder uses a series of sliding window self-attention and cross-attention layers to aggregate token representations into patch representations. In LST,



(a) Static patching segments speech into fixed-size patches, while aligned patching uses Wav2Vec2+CTC boundaries. (`sw` = spoken word and `sil` = silence.)



(b) Example of alignment from Wav2Vec2 + CTC, where purple markers indicate word boundaries, aligning the audio signal with corresponding text.

Figure 3: Illustrations of alignment and patching methods.

we only patch spans of speech tokens using strategies described in Section 3.1. Note that a simple alternative to patching is to use subword tokenization methods like Byte Pair Encoding (BPE) on the speech tokens. This was also explored by Cuervo & Marxer (2024) and similar to them, failed to improve performance in our experiments (ablations in Section 5). Unlike BLT, we do not use hash embeddings, as they did not provide improvements in our experiments.

Local Decoder. A light-weight transformer is used as a decoder and trained with NTP loss, with cross-attention layers inserted between every transformer layer. Each token attends to both the previously generated speech patches and text tokens to incorporate patch-level information (using cross-attention) as well as a sliding window of the past 512 tokens (using self-attention).

3.1 PATCHING

Let $\mathbf{X} = [x_0, \ldots, x_T] \in \mathbb{R}^{T \times d}$ be speech token embeddings obtained using a learned embedding matrix applied to speech tokens $\{s_0, \ldots, s_T\}$. The process of patching maps $\mathbf{X}$ to a shorter sequence of patch embeddings $\mathbf{Z} = [z_0, \ldots, z_{T'}] \in \mathbb{R}^{T' \times d}$ by aggregating local frame segments. For a frame-index set $\mathcal{P}_i \subseteq \{0, \ldots, T\}$, a patch embedding is formed via the local encoder:

$$z_i = \text{LocalEnc}(X_{\mathcal{P}_i}),$$

integrating the frames indexed by $\mathcal{P}_i$ into a single patch embedding. Different patching strategies correspond to different segmentation $\{\mathcal{P}_i\}$.

Static Patching. Speech sequence is split into non-overlapping segments of a fixed length p (patch size). Each patch token is obtained by the local encoder from the embeddings in the patch:

$$\mathcal{P}_i = \{ip, \ldots, \min((i+1)p - 1, T)\}.$$

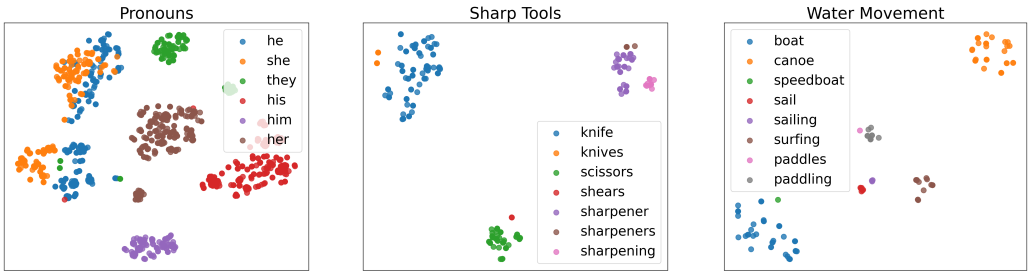
For $p = 3$ and input embeddings $\mathbf{X} = [x_0, x_1, x_2, x_3, x_4, x_5, x_6, \ldots]$, the first patch is $\{x_0, x_1, x_2\}$, the second $\{x_3, x_4, x_5\}$, and so on. Each segment is encoded into a single patch embedding z_i by the local encoder. This provides a uniform compression ratio independent of alignment information.

Alignment Patching. To better synchronize speech and text at the semantic level, alignment patching leverages forced alignment timestamps between speech frames and textual units (e.g. words or BPE tokens). Let $\mathcal{A} = \{(b_k, e_k)\}_{k=1}^{K}$ denotes the aligned frame ranges, where $[b_k, e_k]$ spans the k-th textual unit. The corresponding patch is

$$\mathcal{P}_k = \{b_k, \ldots, e_k\}.$$

Frames outside text spans (e.g., silence) are grouped into separate patches (Fig. 3a). For instance, if one word aligns to $[2, 4]$ and the next to $[6, 7]$, patches are $\{x_2, x_3, x_4\}$ and $\{x_6, x_7\}$, with silence forming $\{x_0, x_1\}$ and $\{x_5\}$.

We obtain alignments with Wav2Vec2+CTC (Baevski et al., 2020), yielding one patch per text unit and silence segment (Fig. 3b). While this enforces cross-modal correspondence, it requires an auxiliary model at inference, introducing possible errors. *Curriculum patching* (Sec. 3.1) mitigates this by gradually shifting from aligned to static patching during training.



(a) Personal Pronouns (b) Sharp Tool Terms (c) Water-related Terms

Figure 4: Visualization of word-level speech patch embeddings from alignment patching models on HellaSwag speech, grouped by different linguistic categories.

Curriculum Patching. Curriculum patching interpolates between alignment-based and static patching during training. Let $P(u) \in [0, 1]$ denote the probability of using alignment at training step u:

$$P(u) = \begin{cases} 1, & u < \tau_1, \\ 1 - \frac{u-\tau_1}{\tau_2-\tau_1}, & \tau_1 \le u < \tau_2, \\ 0, & u \ge \tau_2. \end{cases}$$

At step u, we choose alignment patches with probability $P(u)$ and static patches otherwise. This retains alignment benefits during early training while enabling simple static-only inference.

4 Experimental Setup

We next describe the datasets, models, and evaluation protocols used in our experiments.

4.1 Training and Evaluation Datasets

Our pre-training data comprises a mixture of text and interleaved speech datasets.

Text. Our text training data consists of extensive web and academic corpora, sourced from a selected portion of the Llama 2 pre-training collection (Touvron et al., 2023), totaling 1.8T tokens. We follow the LLaMA 2 setup and apply its SentencePiece (Kudo & Richardson, 2018) BPE tokenizer with a 32K vocabulary.

Speech. Our speech training data includes speech which is discretized into HuBERT tokens (501-entry codebook at 25Hz) together with paired text transcriptions. We use LibriLight (60k hours), People's Speech (30k hours), Multilingual LibriSpeech (50k hours), and Spotify (60k hours), detailed in Table 1. All corpora are aligned using the Wav2Vec2 + CTC framework to provide token-level correspondence between speech and text (Figure 3b).

Table 1: Speech training datasets with total speech hours and the amount of Hubert tokens.

Dataset	Hours	Hubert Tokens (B)
LibriLight (Kahn et al., 2020)	44,174	3.7
People Speech (Galvez et al., 2021)	14,699	1.2
Multilingual LibriSpeech (Pratap et al., 2020)	50,601	4.2
Spotify (Clifton et al., 2020)	55,309	4.6

We evaluate the model on three benchmarks, where each dataset provides a narrative context and candidate endings, and the model selects the most plausible continuation. Together, they test narrative understanding, commonsense reasoning, and topic coherence. We evaluate the model in both speech-to-speech (S→S) and text-to-text (T→T) modes. For the speech mode, we apply Kokoro TTS model (hexgrad, 2025) to generate the speech for evaluation.

sHellaSWAG (HS). We create a speech version of HellaSwag (Zellers et al., 2019) with Kokoro TTS. This benchmark evaluates everyday commonsense reasoning with spoken inputs and outputs. To ensure fairness, we generate the speech for prompts and responses independently and concatenate them afterwards, so that all responses are evaluated against the same speech prompt.

StoryCloze and Topic StoryCloze (SC/TSC). SC (Mostafazadeh et al., 2016) and its topic-based extension TSC (Hassid et al., 2023) are widely used in prior multimodal work (e.g., Nguyen et al. 2025) to test coherence and topic-sensitive reasoning. We resynthesize both datasets with Kokoro TTS for higher-quality speech inputs.

Table 2: Evaluation datasets for story completion (MC = Multiple Choice).

Dataset	Format	Focus
HellaSwag (Zellers et al., 2019)	1-in-4 MC	Commonsense reasoning
StoryCloze (Mostafazadeh et al., 2016)	1-in-2 MC	Narrative coherence
TopicStoryCloze (Hassid et al., 2023)	1-in-2 MC	Topic consistency

We omit sWUGGY and sBLiMP (Nguyen et al., 2020), as they target lexical and syntactic judgments on very short speech segments. Such settings are less aligned with our focus on narrative reasoning, where story-level coherence and commonsense understanding are required.

4.2 LST Models and Baselines

LST Models. We explore four patching strategies for speech tokens:

- **Static Patching.** Fixed-length patches (4 HuBERT tokens) as in Yu et al. (2023), independent of alignment and consistent across training/inference.
- **Aligned Patching.** Uses Wav2Vec2+CTC boundaries (Fig. 3b). For each text span $[b_k, e_k]$, we form patch set $\mathcal{P}_k = \{b_k, \ldots, e_k\}$, synchronizing speech and text tokens (Fig. 3a).
- **Mixed Patching.** Randomly applies static or aligned patching per sequence, combining the robustness of static patching with the fine-grained sync of aligned.
- **Curriculum Patching.** Training shifts from aligned (first third) to mixed (middle) to static (final), leveraging early alignment while ensuring robustness to static-only inference.

Baselines. We include two speechLLM systems as baselines:

- **Base SpeechLLM.** Processes speech tokens directly with text tokens, without patching, similar to SpiritLM (Nguyen et al., 2025).
- **BPE SpeechLLM.** Maps speech tokens into 1k BPE units using a SentencePiece tokenizer (Kudo & Richardson, 2018) trained on 100k random speech sequences, replacing speech tokens with BPE-derived units[1].

4.3 Training Settings

To balance modalities, we set speech tokens to account for about one third (33%) of the total training data, while the rest (67%) is text-only. This ensures that the model benefits from large-scale text pretraining while still maintaining substantial exposure to speech for effective multimodal alignment. For comparison, SpiritLM (Nguyen et al., 2025) adopts a different composition: 33% pure speech, 33% interleaved, and 33% text tokens. Since SpiritLM starts from a text-pretrained model, the relatively smaller text fraction is sufficient. In contrast, when training from scratch, we find that using 33% interleaved and 66% text tokens yields better performance (see Appendix A.3).

5 Results

Compute-controlled. We fix the number of training iterations and per-step sequence budget so that all methods process the same number of units (baseline tokens = LST patches). Table 3 shows three

[1]We use the 1k configuration as our BPE baseline, as larger vocabularies (5k, 10k) showed no benefit.

Table 3: Main comparison of LST models and baselines under the **same computation budget** scheme. Each dataset reports both S→S and T→T.

Model	Tokens (T)		HellaSwag		StoryCloze		TopicStoryCloze	
	Int.	Text	S→S	T→T	S→S	T→T	S→S	T→T
Base SpeechLLM	0.9	1.7	39.0	47.0	59.1	67.8	87.5	95.7
BPE SpeechLLM	1.0	1.9	38.0	47.5	58.0	66.4	87.0	93.5
LST (Static)	1.1	2.2	44.3	51.1	60.5	70.3	87.7	**96.2**
LST (Aligned)	1.1	2.2	42.7	51.7	60.4	70.4	86.6	95.7
LST (Mixed)	1.1	2.2	44.3	51.9	**61.4**	70.8	**88.0**	95.9
LST (Curriculum)	1.1	2.2	**45.5**	**52.2**	61.2	**71.6**	87.9	96.1

Table 4: Main comparison of LST models and baselines under the **same speech/text tokens** scheme. Each dataset reports both S→S and T→T.

Model	Compute Savings	HellaSwag		StoryCloze		TopicStoryCloze	
	(%)	S→S	T→T	S→S	T→T	S→S	T→T
Base SpeechLLM	-	40.2	49.6	60.2	69.1	87.5	95.2
BPE SpeechLLM	8.2%	39.4	48.4	58.3	66.3	86.5	93.9
LST (Static)	19.3%	44.3	51.1	60.5	70.3	87.7	**96.2**
LST (Curriculum)	19.7%	**45.5**	**52.2**	**61.2**	**71.6**	**87.9**	96.1

trends on HellaSwag. First, patching increases the effective token budget, benefiting both modalities: Curriculum Patching improves T→T by +5.2 (47.0→52.2) and S→S by +6.5 (39.0→45.5). Second, Aligned Patching is less effective at evaluation, since variable word spans often yield longer patches, reducing the test-time compute. Finally, Mixed and Curriculum patching combine the advantages of shorter evaluation patches with alignment information, consistently outperforming Static and Aligned across datasets.

Data-controlled. Here we fix the data budget with the same amounts of speech and text tokens. Since LST compresses sequences into patches, it processes fewer patch tokens than the baselines, leading to higher efficiency. Table 4 shows that the BPE baseline fails to surpass vanilla SpeechLLM, whereas LST continues to achieve consistent gains. On HellaSwag, Curriculum Patching improves T→T accuracy from 49.6 to 52.2 despite reduced computation, while boosting S→S from 40.2 to 45.5. Similar improvements are observed on StoryCloze and TopicStoryCloze. Overall, LST with Curriculum Patching reduces the speech–text performance gap from 9.4 to 6.7, demonstrating that alignment through patching benefits both modalities while offering meaningful compute savings.

Visualization of Word-Level Speech Patch Embeddings We use t-SNE (van der Maaten & Hinton, 2008) to project embeddings of representative word groups from the aligned-patching LST model (Fig. 4). Across different categories, embeddings of the same word consistently form tight clusters, while different words remain well separated. Each word forms its own cluster (e.g., he, she, they in pronouns; knife, scissors, sharpener in tools; boat, canoe, surfing in water-related terms). Related variants such as sail–sailing show stability under inflection, while semantically similar pairs like scissors–shears also appear nearby despite being distinct words. These qualitative patterns match quantitative results: within-word similarity is high ($\sim$0.87), between-word similarity is much lower ($\sim$0.43), and silhouette scores (0.65–0.68) (Rousseeuw, 1987) confirm well-separated clusters.

Scaling Trends. Table 5 summarizes results at both 1B and 7B scales, while Figure 5 provides the training curve of the 7B model on HellaSwag. Scaling consistently improves performance across all datasets. At 1B, LST already outperforms the baseline (e.g., 41.3 vs. 36.8 on S→S, and 49.2 vs. 47.1 on T→T). At 7B, the improvements persist: LST reaches 44.2/55.3 compared to the baseline's 42.0/54.8. The figure further shows that LST exhibits a steeper growth curve over iterations, indicating more efficient utilization of larger capacity. Importantly, the 7B model remains far from convergence under the same processed token budget but with much fewer iterations, suggesting that extended training would likely amplify the advantage of LST and further widen the gap.

Table 5: Scaling trends of baseline SpeechLLM and LST models at 1B and 7B parameter scales. Each dataset reports both S→S and T→T.

Model	Batch (M)	Iters (k)	HellaSwag S→S	HellaSwag T→T
Baseline (1B)	0.5	200	36.8	47.1
LST (1B)	0.5	200	**41.3**	**49.2**
Baseline (7B)	4.0	25	42.0	54.8
LST (7B)	4.0	25	**44.2**	**55.3**

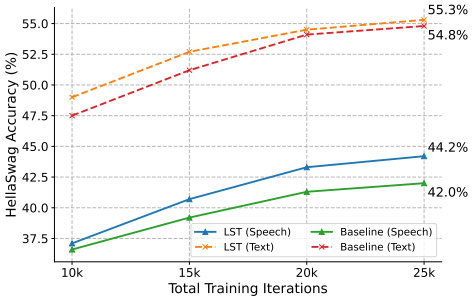


Figure 5: Comparison of LST and baseline on 7B model at 25k iterations. Results are reported on HellaSwag S→S and T→T.

Ablation on Patching Strategies Table 6 compares static and aligned patching. Aligned patching uses word boundaries from alignment, producing semantically coherent patches. We consider two variants: Align (sil sep.), keeping silence spans as separate patches, and Align (sil merged), merging them with adjacent words. Both outperform static patching at similar patch sizes—for instance, Align (sil sep.) reaches 60.3 on StoryCloze S→S vs. 58.7 for static size 6, and Align (sil merged) scores 38.5 on HellaSwag S→S vs. 37.2 for static size 9. *Curriculum* starts with *Align (sil sep.)* and gradually shifts to *Static* during training, retaining alignment benefits while matching the shorter-patch evaluation regime; it yields the strongest and most consistent results (e.g., 41.3 on HellaSwag S→S). Overall, aligned patching better preserves semantics than static, and curriculum combines alignment supervision with static-style evaluation for the best performance. For completeness, we also report BPE-aligned patching experiments in Appendix A.4.

6 Related Work

LLMs using speech tokens. Early neural audio generation methods included direct auto-regressive generation of the speech waveform (van den Oord et al., 2016), or using adversarial approaches (Kong et al., 2020). Following this, *textless NLP* work (Lakhotia et al., 2021) showed that by using discrete speech tokens obtained from self-supervised speech encoders (CPC, wav2vec 2.0, HuBERT) as targets for language modeling, can enable fully spoken LLMs. AudioLM (Borsos et al., 2023) further uses hierarchical generation, first predicts semantic tokens, and subsequent stages predict fine-grained acoustic tokens from SoundStream (Zeghidour et al., 2021), to achieve both high audio quality as well as long-term consistency. In addition to augmenting semantic speech tokens with pitch and style tokens to explicitly model expressivity, SpiritLM (Nguyen et al., 2025) also introduced interleaving speech modeling with text-tokens. More recently, Moshi (Défossez et al., 2024) propose a hierarchical *inner monologue* method, that jointly predicts time-aligned text and acoustic tokens (with distilled semantic information), together with modeling multiple-stream audio for handling full-duplex audio dialogues. Finally, similar to scaling laws for text LLMs (Hoffmann et al., 2022), Cuervo & Marxer (2024) fit scaling law curves to predict the performance of spoken LLMs, and find that they scale upto three order of magnitude more slowly than text LLMs.

Transferring textual knowledge into speech LMs. Slower scaling trends, together with a disproportionately lower amount of data, lead to a knowledge and reasoning gap between speech and text LLMs. To bridge this, AudioPaLM and TWIST (Rubenstein et al., 2023; Hassid et al., 2023) initialize a spoken LLM from a strong text model (PaLM-2, LLaMA), improving both speech understanding/generation and cross-lingual transfer. SpiritLM demonstrates that interleaved speech–text training significantly improves inter-modality knowledge transfer. Spectron (Nachmani et al.) uses a "Chain-of-Modality" pipeline to first produce text and then speech conditioned on the text, trading latency for stronger textual control, while Moshi (Défossez et al., 2024) uses a similar approach but generates interleaved text and speech as an inner monologue. To improve latency, LLaMA-Omni (Fang et al., 2024) style systems decode text and speech simultaneously, by upsampling textual LLM hidden states to decode speech units, before proceeding to decode the next text token.

Speech model efficiency. Compared to text, speech yields much longer token sequences, owing to higher frequency audio codecs, that consume many times additional compute to pre-train and

Table 6: Comparison of patching strategies. Static uses fixed patch lengths. Align (sil sep.) treats silence as separate patches, Align (sil merged) merges silence into words, and Curriculum starts with Align (sil sep.) and gradually shifts to Static during training.

Model	**Ave Patch Size** (tokens)	HellaSwag S→S	HellaSwag T→T	StoryCloze S→S	StoryCloze T→T	TopicStoryCloze S→S	TopicStoryCloze T→T
LST (Static)	4	40.5	48.8	58.2	**69.4**	86.2	95.1
LST (**Curriculum**)	5.8*→ 4	**41.3**	**49.2**	**58.6**	67.8	**86.6**	**95.4**
LST (**Align, sil sep.**)	5.8*	**39.9**	**49.3**	**60.3**	**69.9**	**85.7**	**95.3**
LST (Static)	6	39.4	49.2	58.7	69.6	84.9	94.9
LST (Static)	9	37.2	**49.4**	57.5	**69.7**	84.7	95.9
LST (**Align, sil merged**)	9.4	**38.5**	49.0	**58.8**	**69.7**	**86.9**	**96.0**

* The average patch length is 5.8 for words in Align (sil sep.), while silence has an average of 3.7.

generate. Efforts to mitigate this include methods to produce coarser speech units (Baade et al.; Tseng et al., 2025), hierarchical generation (Borsos et al., 2023), and producing residual tokens using parallel streams (Copet et al., 2023). Attempts at text-inspired approaches to compress token sequences such as BPE (Ren et al., 2022; Li et al., 2024) achieved limited success. In this paper, we take inspiration from recent dynamic patching approaches that have yielded improvements in other modalities such as text (Pagnoni et al., 2024; Yu et al., 2023; Videau et al., 2025) and vision (Pang et al., 2024; Beyer et al., 2023), and extend these methods to speech-text LLMs.

Speech Understanding Benchmarks. Going beyond measuring only acoustic and phonetic capabilities of speech models using scores such as ABX (Kahn et al., 2020), Nguyen et al. (2020) established the Zero Resource Speech Benchmark 2021, comprising datasets/metrics to evaluate lexical (sWUGGY), syntactic (sBLIMP) and lexical-semantic (sSIMI) capabilities of spoken LLMs. Since these benchmarks contrast between very short speech segments, we found that dynamic compute approaches such as ours, do not yield significant improvements. However, subsequently, Hassid et al. (2023) introduced the sStoryCloze and TopicStoryCloze datasets, which are story completion benchmarks in the speech modality measuring commonsense/understanding abilities of Spoken LLMs. We use these benchmarks in this paper, together with a speech version of the popular HellaSWAG textual benchmark, also measuring commonsense reasoning capabilities.

7 Limitations

Our study has several limitations. First, we focus on half-duplex speech–text modeling, where speech and text alternate in turns, and do not yet address full-duplex interaction required for real-time dialogue such as Moshi (Défossez et al., 2024). Second, our analysis is restricted to the pre-training stage, without exploring instruction fine-tuning or downstream adaptation, which we leave for future work. Third, some of our patching strategies, such as alignment and curriculum, rely on forced alignments during pre-training; although curriculum patching reduces this dependency at inference, fully alignment-free approaches remain an open challenge. Finally, our experiments are limited to the speech–text modality, and we have not yet extended LST to additional modalities such as image or video, which represent a promising next direction.

8 Conclusion

We presented the Latent Speech-Text Transformer (LST), a patching-based framework that compresses speech tokens into latent units for efficient and balanced multimodal training. Our experiments demonstrate that LST consistently outperforms baseline SpeechLLMs, with curriculum patching delivering the most robust gains across diverse datasets. By reducing speech–text imbalance, LST improves speech understanding while maintaining strong text performance, and its advantages grow further with model scaling. These findings highlight LST as a practical and scalable approach to bridging speech and text, offering improved efficiency, stronger cross-modal transfer, and greater robustness under varying training conditions.

Ethics Statement

This work adheres to the ICLR Code of Ethics. We use only publicly available datasets (e.g., LibriLight, People's Speech, Multilingual LibriSpeech, Spotify) and did not collect any new human-subject data. No personally identifiable information is included. Our methods aim to improve efficiency and generalization in speech–text modeling, with potential positive impacts on accessibility and multilingual applications. As with all large language models, there remain general risks of misuse, and we encourage responsible use of our work.

References

Alan Baade, Puyuan Peng, and David Harwath. Syllablelm: Learning coarse semantic units for speech language models. In *The Thirteenth International Conference on Learning Representations*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *CVPR*, 2023.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.

Ann Clifton, Aasish Pappu, Sravana Reddy, Yongze Yu, Jussi Karlgren, Ben Carterette, and Rosie Jones. The spotify podcast dataset. *arXiv preprint arXiv:2004.04270*, 2020.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720, 2023.

Santiago Cuervo and Ricard Marxer. Scaling properties of speech language models. *arXiv preprint arXiv:2404.00685*, 2024.

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. *arXiv e-prints*, pp. arXiv–2410, 2024.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*, 2024.

Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. *arXiv preprint arXiv:2111.09344*, 2021.

Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36:63483–63501, 2023.

hexgrad. Kokoro-82m — when smaller means better in text-to-speech. `https://huggingface.co/hexgrad/Kokoro-82M`, 2025. Accessed: 2025-04-22.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7669–7673. IEEE, 2020.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2018.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021.

Bohan Li, Feiyu Shen, Yiwei Guo, Shuai Wang, Xie Chen, and Kai Yu. On the effectiveness of acoustic bpe in decoder-only tts. In *Proc. Interspeech 2024*, pp. 4134–4138, 2024.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*, 2016.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken question answering and speech continuation using spectrogram-powered llm. In *The Twelfth International Conference on Learning Representations*.

Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux. The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. In *NeuRIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, et al. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*, 2024.

Yatian Pang, Peng Jin, Shuo Yang, Bin Lin, Bin Zhu, Zhenyu Tang, Liuhan Chen, Francis EH Tay, Ser-Nam Lim, Harry Yang, et al. Next patch prediction for autoregressive visual generation. *CoRR*, 2024.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

Shuo Ren, Shujie Liu, Yu Wu, Long Zhou, and Furu Wei. Speech pre-training with acoustic piece. In *Proc. Interspeech 2022*, pp. 2648–2652, 2022.

Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Liang-Hsuan Tseng, Yi-Chang Chen, Kuan-Yi Lee, Da-Shan Shiu, and Hung-yi Lee. Taste: Text-aligned speech tokenization and embedding for spoken language modeling. *arXiv preprint arXiv:2504.07053*, 2025.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Proc. SSW 2016*, pp. 125–125, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Mathurin Videau, Badr Youbi Idrissi, Alessandro Leite, Marc Schoenauer, Olivier Teytaud, and David Lopez-Paz. From bytes to ideas: Language modeling with autoregressive u-nets. *arXiv preprint arXiv:2506.14761*, 2025.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36:78808–78823, 2023.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

A Appendix

A.1 Interleaved Data Construction

To generate interleaved sequences from parallel speech–text data, we proceed as follows:

1. **Alignment.** We obtain alignment information by using Wav2Vec2 + CTC to determine the boundaries linking text tokens to their corresponding spans of speech tokens.
2. **Span selection.** For each training example, we randomly select a contiguous span of words. The selected span is replaced by text tokens, while the following span of approximately half that length is kept as speech tokens.
3. **Modality markers.** We insert special tokens `<t>` and `<s>` to indicate the start of text and speech segments, respectively. This ensures the model can disambiguate between modalities.
4. **Dynamic sampling.** Interleaved sequences are generated dynamically at training time, so each epoch exposes the model to different interleaving patterns for better robustness.

This process yields diverse interleaved training examples while preserving alignment between speech and text, allowing the same model to be applied uniformly to S→S, S→T, T→S, and T→T tasks.

A.2 HYPERPARAMETERS

A.2.1 OPTIMIZATION AND TRAINING CONFIGURATION

We trained the model using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay = 0.1). The learning rate was initialized at 4×10^{-4} and scheduled with cosine decay, including a warmup period of 2,000 steps and a minimum ratio of 0.01 at the final step. For the 1B model, training was performed on 32 H100 GPUs with a per-GPU batch size of 4 sequences (sequence length = 4,096 units), leading to a total batch size of 0.5M units. Mixed-precision training with bfloat16 was used for efficiency. Gradient clipping was applied at 1.0, and gradient accumulation was set to 1. Model parallelism used a single partition, and Fully Sharded Data Parallel (FSDP) was enabled for memory efficiency. No dropout was applied. The 1B model was trained for 200k steps, corresponding to approximately 1 trillion units, and required around 17 hours to complete on 32 H100 GPUs.

A.2.2 MODEL ARCHITECTURE

Table 7 summarizes the hierarchical architecture used in our experiments. The model consists of a shallow local encoder, a deep global transformer, and a moderately deep local decoder. The local modules operate with restricted attention windows to capture fine-grained context, while the global transformer uses block-causal attention with rotary position embeddings (RoPE) to handle long-range dependencies efficiently. This design balances local detail preservation with scalable long-context modeling.

Table 7: Model architecture configuration. Each module is shown with its depth, hidden dimension, number of attention heads, and other relevant settings.

Module	Layers	Dim.	Heads	Notes
Local Encoder	1	1024	16	Local window = 512
Global Transformer	25	2048	16	Block-causal; RoPE ($\theta = 5 \times 10^5$)
Local Decoder	9	1024	16	Local window = 512

A.3 EFFECT OF SPEECH PROPORTION

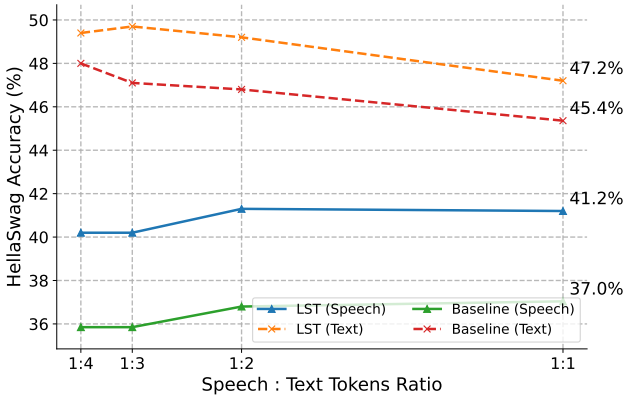


Figure 6: Effect of speech-to-text token ratio at 200k iterations. Results are reported on HellaSwag under both S→S and T→T.

Figure 6 illustrates the effect of varying the training speech–to–text token ratio on HellaSwag. Across all settings, LST consistently outperforms the baseline, and both methods exhibit the best speech–text trade-off at the 1:2 ratio. Moving from 1:3 to 1:2 improves LST (S→S) from 40.2 to 41.3 while keeping LST (T→T) high at 49.7; pushing further to 1:1 does not provide speech gain (41.2) but a large text drop (47.2, −2.5). The baseline shows the same pattern: at 1:2 it reaches 36.8 (S→S) and 47.1 (T→T), whereas 1:1 gives only 37.0 on speech (+0.2) but lowers text to 45.4

(−1.7). Averaging speech and text accuracies, the macro score peaks at 1:2 for both LST and the baseline. These results indicate that allocating about one-third of tokens to speech (1:2) offers a fair and robust operating point for both models to avoid the substantial text-side degradation seen at 1:1 while securing clear gains over lower speech ratios.

A.4 BPE-ALIGNED PATCHING

In addition to word-aligned patching, we also explored BPE-aligned patching, where speech patches are constructed according to BPE segmentation of the text. To ensure comparability, we applied the same forced-alignment procedure at the character level and then mapped aligned spans to their corresponding BPE units. While this provides finer granularity, the resulting boundaries are less precise and the subword pieces do not always correspond to meaningful acoustic events. As shown in Table 8, word alignment generally outperforms BPE alignment in S→S (e.g., 59.4 vs. 55.6 on StoryCloze and 84.8 vs. 79.6 on TopicStoryCloze), reflecting the more reliable word-level boundaries. On the other hand, BPE achieves slightly better T→T results, likely because its patching is directly aligned with the underlying text BPE tokens. Finally, curriculum training further boosts HellaSwag S→S performance, improving from 40.0/39.2 to 41.5/41.3 for Word and BPE, respectively.

Table 8: Comparison of aligned patching strategies under a speech-to-text token ratio of 1:4. *Word Align* uses word-level forced alignment, *BPE Align* uses BPE segmentation, and *Curriculum* gradually shifts from alignment-based to static patching.

Model	**Ave Patch Size** (tokens)	HellaSwag		StoryCloze		TopicStoryCloze	
		S→S	T→T	S→S	T→T	S→S	T→T
LST (Word Align)	5.8*	**40.0**	49.9	**59.4**	68.6	**84.8**	94.6
LST (BPE Align)	5.0*	39.2	**50.1**	55.6	**69.1**	79.6	**95.6**
LST (Word Curr.)	5.8→4	**41.5**	**49.5**	57.9	**68.9**	**86.8**	95.1
LST (BPE Curr.)	5.0→4	41.3	48.6	**59.1**	67.1	86.5	**95.4**

* The average patch length is 5.8 for words, 5.0 for BPEs, and 3.7 for silence spans.

A.5 LLM USAGE.

We used large language models solely for polishing some sentences in this paper.