Foundation Models for Spoken Language Understanding : Capability, Efficiency, and Application

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the registrar's office.

Thèse n. xxxx 2026
présentée le 15 juin 2026
Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique
École polytechnique fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Mutian HE

acceptée sur proposition du jury :

Prof Andrei Popescu-Belis, président du jury
Prof Jean-Philippe Thiran, Dr Philip N. Garner, directeur de thèse
Dr Imanol Schlág, rapporteur
Prof Elisabeth André, rapporteur
Prof Rico Sennrich, rapporteur

Lausanne, EPFL, 2026

Acknowledgements

I would never have completed this thesis without the help, support, and encouragement of many people. First and foremost, I would like to thank my advisor, Dr. Philip Garner, for his guidance, support, and trust throughout my doctoral studies. I am especially grateful for the freedom he gave me to explore uncharted research topics, and for his encouragement as I pursued directions that were new to our group.

I also thank my thesis director, Prof. Jean-Philippe Thiran, for his support and for helping to smooth the academic and administrative processes within EPFL. I am grateful to the members of my thesis jury for taking the time to review this thesis, participate in the defense, and provide valuable feedback.

My sincere thanks go to my colleagues in the broader research group of Idiap for their support, discussions, and exchange of ideas. I am also grateful to the linear attention and FLA community, whose open exchange of ideas and infrastructure greatly influenced the later stages of my PhD and helped shape my research.

I gratefully acknowledge the SNSF STEADI project for funding my doctoral research. I thank all the collaborators from UNIL and UniNE within the project. I also thank all the staff at Idiap for their kindness and essential help, which made it much easier for me to navigate and adapt to life here.

Above all, I am profoundly thankful to my family, my partner, and my close friends especially all the lovely Shikas for their unwavering support. Their presence carried me through the intellectual challenges, emotional burdens, and sleepless nights of this journey, which I could never have survived without them. This thesis is as much a testament to their love, patience, and faith in me as it is to my own work. I owe them more than words can express, and I dedicate my deepest gratitude to them.

Martigny, June, 2026

Mutian He

Abstract

At a rather high level, this thesis is motivated by the goal of inference of job suitability from audio recordings of interviews and work samples. This in turn defines the technical goal of analysis and understanding of long audio recordings in French with a limited amount of data. Of particular relevance, the emergence of large-scale pretrained generalist models, including large language models (LLMs), is reshaping the landscape of speech processing. Such models offer an effective solution to our goal, whilst several issues remain: (i) how capable such models are for speech, (ii) how to best apply or adapt them to the target tasks in the speech domain, and (iii) how to efficiently process long speech sequences that often exceed text by an order of magnitude. We explore them through a series of evaluations, model architecture design, and training techniques, as presented in this thesis.

We begin by studying an auxiliary supervision stage for pretrained speech models prior to task-specific fine-tuning. We show that speech translation mid-training substantially improves semantic understanding and few-shot cross-lingual transfer for models such as wav2vec2. We further demonstrate that preserving pretraining knowledge via Bayesian regularizers can improve downstream performance.

With the emergence of LLMs starting from ChatGPT, the next part of the thesis evaluates the role of ChatGPT and other text-only LMs on spoken language understanding. We demonstrate their strong zero- and few-shot performance, approaching supervised baselines, and confirm the emergent in-context learning behavior unique in LLMs. Meanwhile we also identify limitations of such models including sensitivity to ASR errors and degradation on under-specified and vague tasks.

Third, we address the computational bottleneck of long-form speech modeling arising from high sampling rates and the information-sparse nature of speech, compounded by the $O(N^2)$ time and $O(N)$ space demands of transformer-based models. Although efficient sequence models such as linear attention are advancing rapidly, few pretrained speech models exist in these architectures. To bridge this gap without re-pretraining, we propose a systematic layer-wise distillation framework that converts pretrained speech and language Transformers into target efficient architectures (e.g., Linformer, Mamba) during fine-tuning on downstream speech or NLP data. We also develop distillation strategies that improve downstream performance by better preserving pretraining knowledge.

Fourth, we analyze "forgetfulness" in linear attention that current linear attention models gradually lose information of distant past, due to limited recurrent state or memory capacity. We propose hybrid linear–sparse mechanisms that grant direct access to distant tokens without

sacrificing efficiency. Particularly, we investigated query-aware sparse attention mechanisms, as well as a novel contextualized learnable token eviction approach that maintains linear attention's $O(1)$ space appeal. Empirical studies confirmed the advantage of our methods over existing hybrid baselines (e.g. with sliding-window attention) in long-context scenarios, without compromising efficiency thanks to our hardware-efficient implementation.
In the end, we apply these methods to long French interviews recording. We show that pretrained speech models can be adapted to identify storytelling, an important aspect of interview evaluation. We also show the critical role of contextual information in this task.
In summary, the thesis contributes to the advancement of speech and language understanding through improved model architecture, pretraining, and adaptation techniques. In particular, we investigate efficient attention approaches including linear attention, sparse attention, and their hybrids, which can be essential for long-context scenarios.
Keywords: spoken language understanding, long-form modeling, large language models, linear attention, sparse attention, hybrid attention

Résumé

À un niveau relativement général, cette thèse est motivée par l'objectif d'inférer l'adéquation à un emploi à partir d'enregistrements audio d'entretiens et d'échantillons de travail. Cela définit à son tour l'objectif technique d'analyse et de compréhension de longs enregistrements audio en français, avec une quantité limitée de données.
L'émergence de grands modèles généralistes préentraînés, notamment les grands modèles de langage (LLM), est particulièrement pertinente, car elle transforme le paysage du traitement de la parole. De tels modèles offrent une solution efficace à notre objectif, même si plusieurs questions demeurent : (i) dans quelle mesure ces modèles sont capables de traiter la parole, (ii) comment les appliquer ou les adapter au mieux aux tâches cibles dans le domaine de la parole, et (iii) comment traiter efficacement de longues séquences de parole, qui dépassent souvent le texte d'un ordre de grandeur. Nous explorons ces questions au moyen d'une série d'évaluations, de conceptions d'architectures de modèles et de techniques d'entraînement, présentées dans cette thèse.
Nous commençons par étudier une étape de supervision auxiliaire pour des modèles de parole préentraînés avant l'affinage spécifique à la tâche. Nous montrons que l'entraînement intermédiaire par traduction de la parole améliore substantiellement la compréhension sémantique et le transfert interlingue à faible nombre d'exemples pour des modèles tels que wav2vec2. Nous démontrons en outre que la préservation des connaissances acquises lors du préentraînement, au moyen de régularisateurs bayésiens, peut améliorer les performances sur les tâches en aval.
Avec l'émergence des LLM à partir de ChatGPT, la partie suivante de la thèse évalue le rôle de ChatGPT et d'autres modèles de langage uniquement textuels dans la compréhension du langage parlé. Nous démontrons leurs fortes performances en régime zéro exemple et à faible nombre d'exemples, proches des références supervisées, et confirmons le comportement émergent d'apprentissage en contexte propre aux LLM. Parallèlement, nous identifions également les limites de ces modèles, notamment leur sensibilité aux erreurs de reconnaissance automatique de la parole (ASR) et la dégradation de leurs performances sur des tâches insuffisamment spécifiées et vagues.
Troisièmement, nous abordons le goulot d'étranglement computationnel de la modélisation des longs enregistrements audio, qui résulte des fréquences d'échantillonnage élevées et de la faible densité informationnelle de la parole, aggravées par les exigences en temps $O(N^2)$ et en espace $O(N)$ des modèles fondés sur les Transformers. Bien que les modèles de séquences efficaces, tels que l'attention linéaire, progressent rapidement, peu de modèles

de parole préentraînés existent dans ces architectures. Afin de combler cette lacune sans recourir à un nouveau préentraînement, nous proposons un cadre systématique de distillation couche par couche qui convertit des Transformers préentraînés pour la parole et le langage en architectures cibles efficaces, telles que Linformer ou Mamba, lors de l'affinage sur des données de parole ou de TAL en aval. Nous développons également des stratégies de distillation qui améliorent les performances en aval en préservant mieux les connaissances issues du préentraînement.
Quatrièmement, nous analysons « l'oubli » dans l'attention linéaire, c'est-à-dire le fait que les modèles actuels d'attention linéaire perdent progressivement l'information du passé distant en raison d'un état récurrent ou d'une capacité mémoire limités. Nous proposons des mécanismes hybrides linéaires–creux qui permettent un accès direct aux tokens distants sans sacrifier l'efficacité. En particulier, nous avons étudié des mécanismes d'attention parcimonieuse sensibles aux requêtes, ainsi qu'une nouvelle approche contextualisée d'éviction apprenable de tokens, qui conserve l'attrait de l'attention linéaire en espace $O(1)$. Des études empiriques confirment l'avantage de nos méthodes par rapport aux références hybrides existantes, par exemple celles fondées sur l'attention à fenêtre glissante, dans des scénarios à long contexte, sans compromettre l'efficacité grâce à notre implémentation adaptée au matériel.
Enfin, nous appliquons ces méthodes à de longs enregistrements d'entretiens en français. Nous montrons que des modèles de parole préentraînés peuvent être adaptés afin d'identifier le storytelling, un aspect important de l'évaluation des entretiens. Nous montrons également le rôle crucial de l'information contextuelle dans cette tâche.
En résumé, cette thèse contribue aux progrès de la compréhension de la parole et du langage grâce à des améliorations portant sur l'architecture des modèles, le préentraînement et les techniques d'adaptation. En particulier, nous étudions des approches d'attention efficaces, notamment l'attention linéaire, l'attention parcimonieuse et leurs hybrides, qui peuvent être essentielles dans les scénarios à long contexte.
Mots-clés : compréhension du langage parlé, parole longue, grands modèles de langue, attention linéaire, attention parcimonieuse, attention hybride

Contents

1 Introduction

1.1 Motivation

The understanding and analysis of spoken language has been always a core task for artificial intelligence (AI) from the early efforts on voice assistants to the current era of multimodal large language models (LLMs). In many real-world settings, speech is not merely a vehicle for lexical content, but a rich and temporally extended signal through which speakers convey intentions, experiences, attitudes, and social meaning. This is particularly evident in complex and lengthy interactional settings such as meetings, lectures, and interviews, where the participants will not simply recognize words, but infer higher-level semantic and pragmatic properties; and we expect an AI system to do the same.

More specifically, this thesis is situated in the context of the project *Storytelling and First Impressions in Face-to-Face and Algorithm-Powered Digital Interviews* (SteADI), an interdisciplinary collaboration between psychologists and computer scientists funded by the Swiss National Science Foundation. Within this broader effort, a long-term objective is to develop methods for the automatic analysis of audio recordings of interview responses; particularly, we are interested in the assessment of favorable constructs, e.g., storytelling quality, which can ultimately support interviewees in improving their self-presentation while also assisting interviewers in their evaluations. This application introduces a number of technical difficulties beyond semantic understanding itself: The material to be analyzed is speech rather than text, interview recordings are often long (e.g. up to 30 minutes), the project targets the French language, and the annotated dataset in this domain is scarce. These characteristics place the problem squarely within the domain of spoken language understanding (SLU), and, at the same time, reveal the limitations of existing methods.

Meanwhile, the landscape of SLU research is under a significant shift. Large pretrained models, also known as foundation models, have first transformed natural language processing (NLP) and are now increasingly reshaping speech processing. Such models trained on massive datasets can be adapted to effectively handle downstream tasks with limited supervision. The subsequent emergence of large language models (LLMs) has demonstrated highly powerful

capabilities in complicated language and reasoning tasks without task-specific supervision, and more recent speech, multimodal, or *omni* LLMs have extended these capabilities directly to the modality of speech. Together, these developments mark a paradigm shift from task-specific modeling toward the use of *generalist* models with sufficient intelligence to handle a broad range of complex tasks. Such flexibility is appealing in light of the practical constraints that commonly arise in realistic scenarios and are especially pronounced in the project setting: The task demands sophisticated understanding, which requires sufficient data to train a machine learning (ML) model following the classical paradigm, yet available data is limited; the task lies in a narrow domain (interview analysis) using a non-English language (French), hence no existing off-the-shelf system is likely to match the task closely. Under these conditions, a strong generalist model is the only feasible solution.

Even with the promise of generalist models, substantial challenges remain before they can be applied effectively to our problem, and this thesis investigates a series of questions organized around the following overarching research question:

> How can pretrained and generalist models be effectively applied to spoken language understanding, especially in long-form and low-resource settings, while preserving both semantic performance and computational efficiency?

This question can be further decomposed into three dimensions of more specific challenges below:

1. **Model Capability**: Current speech foundation models are trained in a self-supervised manner to capture local and low-level acoustic pattern. How can we further enhance them to better acquire and preserve high-level and multilingual semantic understanding capabilities? With the emergence of LLMs, to what extent can they contribute to SLU under limited supervision, and what are their strengths and limitations in this role?
2. **Model Efficiency**: The core of modern large-scale models, namely transformer, requires computation that scales quadratically and memory that scales linearly with context length. This becomes a major limitation when directly processing extended speech recordings, whose information density is much lower than that of text. How can we build more efficient yet still effective alternatives? Particularly, most strong pretrained models are still Transformers, so replacing them with efficient architectures usually requires costly re-pretraining. How can we train efficient models at minimal cost without sacrificing performance? The major type of efficient attention, namely linear attention, compresses past information into limited memory, which can lead to the progressive loss of distant but relevant context. How can this well-known weaknesses in long-range retrieval and memorization be addressed?
3. **Application**: How can these methodological advances be translated back to the motivating application of interview analysis based on extended French speech recordings?

In light of these challenges, this thesis focuses on spoken language understanding from speech recordings in realistic settings characterized by extended context and limited labeled data. Within this scope, it investigates how modern pretrained and generalist models can be adapted and deployed effectively, while also advancing scientific understanding of their capabilities and limitations in this domain.

1.2 Main Contributions

The thesis addresses the aforementioned questions by exploring model adaptation, systematic evaluation, architectural design, distillation, and application. The thesis is organized into three phases, each corresponding to one dimension of the challenge and contributing to both the scientific understanding and the practical development of artificial intelligence and spoken language understanding.

The first phase addresses the semantic capabilities of foundation models through two complementary lines of work. First, we introduce speech translation as an intermediate training stage before downstream fine-tuning to compensate for the limited high-level semantic guidance provided by acoustic self-supervision. We show that speech translation induces the model to map speech to intermediate semantic representations, thereby fostering higher-level understanding and stronger few-shot cross-lingual transfer on SLU tasks, including intent detection, spoken question answering, and speech summarization. We further show that preserving pretrained knowledge during adaptation, notably through multi-task training or Bayesian regularization, improves downstream performance, especially in low-resource settings.

The thesis then turns to large language models. The rise of ChatGPT has made it natural to ask whether powerful text-only generalist models can already address a substantial part of SLU. We evaluate LLMs on multilingual intent detection under zero-shot and few-shot in-context learning settings, and find that, given oracle transcripts, they often match the performance of specialized supervised models. In particular, we confirm that such in-context capabilities are unique to large-scale models like GPT-3.5 and ChatGPT, while absent in small-scale models. While these LLMs also exhibit clear limitations: Their performance degrades with transcription errors, and GPT-3.5 cannot reliably correct such errors. Several probing tests suggest that this weakness may partly stem from a lack of pronunciation knowledge. In addition, the task must be specified clearly in the prompt, and LLMs are weak on tasks like slot filling, where the labeling criteria are complex, potentially ambiguous, and difficult to infer from only a few examples. These encouraging findings motivate the second phase of the thesis, which further investigates LLMs, while also motivating the direct use of speech in end-to-end SLU systems.

The second phase of the thesis addresses a critical challenge in modern end-to-end foundation models: efficiency. Efficient alternatives such as linear attention offer a promising direction, yet two major obstacles remain, i.e. the cost to pretrain efficient models, and the weakness of linear attention models in memorization. To tackle the first obstacle, this thesis proposes a layer-wise distillation framework that converts pretrained Transformers into efficient target

architectures such as Linformer and Mamba during downstream fine-tuning, using only the target dataset and requiring no additional pretraining. Across tasks in natural language processing, language modeling, and speech processing, we recover the performance of the original pretrained transformer. We further show that intermediate models obtained during Transformer fine-tuning may provide effective guidance for distillation. To tackle the second, the thesis investigates hybrid linear-sparse mechanisms that restore direct access to distant information while maintaining favorable efficiency properties. We study hybrids based on both query-aware sparse attention (e.g. DeepSeek's Native Sparse Attention) and query-agnostic sparse attention, i.e. token eviction. Particularly, we propose a novel contextualized lightweight learnable token eviction strategy for query-agnostic sparse attention. Based on this technique, the resulting hybrid attention language model achieves substantially better retrieval performance than both pure linear attention and hybrids based on sliding-window attention, while preserving favorable time and memory costs through a hardware-efficient implementation.

The final phase returns to the original motivation of interview analysis. We examine lengthy French interview recordings and show that pretrained speech models can be adapted to assess storytelling, an important aspect of interview evaluation. We further show that incorporating richer contextual information improves performance, reinforcing one of the central messages of the thesis: effective spoken language understanding in such complex scenarios requires both strong semantic modeling and effective use of longer-range context.

1.3 Thesis Outline

This thesis is organized into eight chapters. The current chapter introduces the general motivation, contributions, and overall structure. The remainder of the thesis is organized as follows.

Chapter 2 reviews the background required for the rest of the thesis. It introduces pretrained language and speech models, large language models, efficient attention mechanisms for long sequences, and the interview-analysis setting that motivates the application studied later.

Chapter 3 studies the enhancement of spoken language understanding via speech translation. It examines speech translation as an intermediate training signal for pretrained speech models and analyzes the benefits of preserving pretrained knowledge during downstream adaptation.

Chapter 4 evaluates the spoken language understanding capabilities of large language models. It investigates zero-shot and few-shot in-context learning performance, compares them with supervised baselines, and analyzes both the promise and the limitations of LLMs for SLU.

Chapter 5 turns to efficiency and introduces a cross-architecture layerwise distillation framework for converting existing pretrained models into efficient linear-complexity architectures. It demonstrates the effectiveness of this approach across multiple tasks and models, and

further shows the value of using intermediate models during distillation.

Chapter 6 continues the study of long-context modeling by investigating hybrid linear-sparse attention. It examines mechanisms that mitigate the forgetting behavior of linear attention and improve retrieval of distant context without sacrificing time and space efficiency.

Chapter 7 returns to the motivating application of interview analysis. It applies pretrained models to French interview recordings in order to identify storytelling and related response categories, and further explores the benefits of incorporating more contextual information.

Finally, Chapter 8 concludes the thesis by synthesizing the main findings, discussing their implications and limitations, and outlining directions for future work.

2 Background

2.1 Pretrained Language Models and Large Language Models

Early AI, including SLU systems, were typically based on small-scale models dedicated to individual tasks such as automatic speech recognition (ASR), trained by an adequate amount of per-task dataset. The introduction of BERT (Devlin et al., 2019) marked a major shift, after which **pretrained language models** (PTLMs), also known as foundation models, have since become a dominant paradigm in natural language processing. PTLMs including BERT, GPT-2 (Radford, Wu, et al., 2019), and T5 (Raffel et al., 2019) are large-scale neural networks (ranging from hundreds of millions to billions of parameters) trained on vast text data through self-supervised objectives, such as autoregressive or masked language modeling. These tasks, which do not require human annotated labels, enable models to learn from an enormous amount of comprehensive text data from books and the Internet at minimal cost. Success in these tasks requires a model to develop general knowledge, understand human language, and make contextualized inferences. Backed by such knowledge and capabilities, these PTLMs enjoy the versatility to be later fine-tuned or adapted to downstream tasks using limited task-specific data.

Following the success of PTLMs, it is demonstrated that by progressively scaling up the size of the model and the training dataset, the model can achieve ever-increasing performance (Kaplan et al., 2020; Hoffmann et al., 2022). These scaling laws pave the way for the emergence of **large language models** (LLMs). Together with rapid advances in computing infrastructure, including GPU capacity and distributed training systems, as well as progress in model architecture, optimization, and reinforcement learning, this development established LLMs as a central paradigm in contemporary AI research and applications. In addition to broader knowledge coverage, LLMs exhibit *emergent abilities* that become pronounced only at sufficient large scales, including instruction following, in-context learning, and chain-of-thought reasoning (Brown, Mann, et al., 2020; Wei, Tay, et al., 2022). Consequently, frontier systems such as ChatGPT, Gemini, Claude, Qwen, DeepSeek, and Kimi have reached up to trillions of parameters, demonstrated versatile and close-to-human performance across language under-

standing, reasoning, coding, and agentic tasks, and now driving an ongoing transformation with profound impact on the whole human society.

2.2 Spoken Language Understanding

Spoken language understanding (SLU) refers to the analysis of speech for the purpose of extracting semantic meaning beyond transcription alone. To understand spoken language, a straightforward approach is to first transcribe speech into text using ASR, and then process the transcript using various **natural language understanding** (NLU) techniques. This cascaded approach has been widely adopted in many SLU systems (Qin, Xie, et al., 2021). However, cascaded SLU systems are vulnerable to ASR errors, which can propagate and amplify through the pipeline (Chang and Chen, 2022; Cheng, Cao, et al., 2023). In addition, information carried by prosody, timing, and other non-lexical aspects of the signal is discarded in transcription, which can be relevant. For these reasons, many recent approaches perform SLU end to end, directly ingesting speech as model inputs rather than relying on an intermediate transcription bottleneck (Serdyuk et al., 2018; Haghani et al., 2018).

Following the success of PTLMs in text processing, the paradigm has been extended to speech and made end-to-end SLU substantially more feasible. Speech foundation models (SFMs) like Wav2Vec2 (Baevski et al., 2020), HuBERT (Hsu, Bolte, et al., 2021), and WavLM (Chen, Wang, Chen, et al., 2022) are trained on large amounts of unlabeled audio through self-supervised objectives, e.g. predicting masked audio frames, whereas Whisper (Radford, Kim, et al., 2023) is trained on large-scale weakly annotated ASR data. Similar to PTLMs in texts, such SFMs are capable of performing various downstream speech tasks effectively with limited supervision (Yang, Chi, et al., 2021; Wang, Boumadane, and Heba, 2022).

More recently, models like SALMONN (Tang, Yu, et al., 2024), Moshi (Défossez et al., 2024), LlaMA-Omni (Fang, Guo, et al., 2025), and Qwen3.5-Omni (Qwen Team, 2026), have endowed LLMs with native multimodal capabilities. In such systems, speech is typically processed by a dedicated **audio encoder**, often derived from a speech foundation model or a neural speech tokenizer, and the resulting encoded audio are aligned with the token-based interface of the LLM through training on a cohort of speech-related tasks including ASR, audio question-answering (AQA), speech translation (ST), etc. Leveraging the semantic capabilities of the underlying LLM, recent speech LLMs have demonstrated strong performance on a broad range of SLU tasks, and have become an important direction in current speech processing research. These developments provide the broader modeling context for the present thesis on how pretrained and generalist models can be adapted and applied to SLU with limited supervision.

2.3 Efficient Attention

2.3.1 Transformer Architecture and its Efficiency Issue

The dominant architecture underlying the modern foundation models discussed above is Transformer (Vaswani et al., 2017). The architecture roughly consists of two types of network blocks: **token mixer** and **channel mixer**. Channel mixers, such as multilayer perceptron (MLP) blocks, process each token individually. Token mixers, by contrast, operate on the sequence as a whole, capture dependencies between tokens, and enable the model to incorporate contextual information. The central strength of the Transformer lies in its token mixer, namely the attention mechanism, which allows each token to interact directly with (or "attend to") all other tokens in the sequence. However, this pairwise all-to-all interaction also makes the architecture computationally expensive, especially for long sequences of the kind considered in this thesis. More specifically, given an input sequence of N encoded tokens (or audio frames, etc.), each *head* of the mechanism first projects it into query, key, and value vectors $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in R^{N\times d}$, and then computes

$$A(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{P}\boldsymbol{V} = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{V}, \tag{2.1}$$

or, equivalently, $A_i = \sum_{j=1}^{N} \text{softmax}(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d}})\boldsymbol{v}_j$ for each query $\boldsymbol{q}_i$. Current LLMs use causal attention for autoregressive generation, i.e., each query token can only attend to previous tokens. This is equivalent to adding a mask $\boldsymbol{M}$ to the pre-softmax attention scores $\boldsymbol{Q}\boldsymbol{K}^T$, where

$$\boldsymbol{M}_{i,j} = \begin{cases} 0, & j \le i \\ -\infty, & j > i \end{cases} \tag{2.2}$$

Outputs of different heads are then concatenated and projected to produce the final output.

Given the full input sequence, attention is a compute-intensive process in that it processes all the pairwise interactions in parallel, resulting in $O(N^2)$ time. This quadratic cost arises both during training and during the initial inference stage, known as **prefilling**, when the model processes the entire input prompt. In current autoregressive LLMs, computed $\boldsymbol{K}$ and $\boldsymbol{V}$ are also stored into an $O(N)$-space key-value (KV) cache. Then during the **decoding** phase, the LLM generates tokens one by one, and each token attends to all previous tokens in the KV cache. Each decoding step takes $O(N)$ time, bounded by memory bandwidth. As an autoregressive process, the decoding stage dominates the overall inference time.

These properties create several major efficiency bottlenecks. The $O(N^2)$ computation time is prohibitive for long sequences. Also, the $O(N)$ KV cache limits the maximum context length and inference batch size due to memory constraints, and it also contributes to inference latency because decoding is bounded by memory bandwidth. Several practical remedies for these issues have become standard in modern LLMs. Flash Attention (Dao, Fu, et al., 2022) accelerates attention computation through a hardware-aware implementation of tiled online

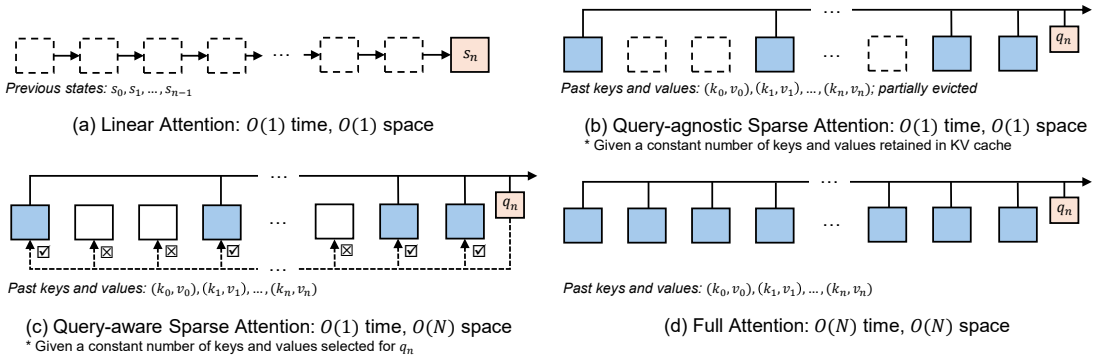


Figure 2.1: A hierarchy of different token mixers with different time and space complexity, from the cheapest linear attention and query-agnostic sparse attention to the more expensive query-aware sparse attention and full attention.

softmax and avoids the materialization of the $O(N^2)$ attention weight matrix $\boldsymbol{P}$. Techniques such as grouped-query attention, multi-query attention (Ainslie, Lee-Thorp, et al., 2023), and multi-latent attention (DeepSeek-AI et al., 2024) directly reduce the KV-cache size per token. However, these methods do not alter the essence of pairwise attention computation. These limitations motivate a broad family of alternative token mixers known as efficient attention mechanisms, including sparse attention and linear attention, as illustrated in Figure 2.1. These mechanisms form an important part of the methodological background for this thesis.

2.3.2 Sparse Attention

The efficiency problem arises from pairwise interactions, so a straightforward strategy is to restrict attention to a small subset of token pairs. More specifically, sparse attention attends only to KV pairs indexed by a subset $\mathcal{I} \subseteq \{1, 2, ..., n\}$, or more generally by a query-dependent subset $\mathcal{I}_i$ for each $\boldsymbol{q}_i$, believed to be important or relevant according to some criterion. This is equivalent to assigning a value of $-\infty$ to unselected positions in $\boldsymbol{M}$. In this way, the computation time of the attention mechanism is reduced to $O(N|\mathcal{I}|)$, but each token can only directly interact with a limited set of tokens. This can be particularly risky for memorization or retrieval-intensive tasks such as question answering, which require the model to recall relevant information from a long context; for example, reading a long biography and answering about details of a mentioned event. Performance can be substantially compromised if $\mathcal{I}$ fails to include the tokens containing the information needed to answer the query. Therefore, the effectiveness of sparse attention depends on selecting the most relevant KV pairs for each query while maintaining a favorable balance between retrieval quality and computational cost.

Such methods can be broadly categorized into two groups: **query-agnostic** and **query-aware** sparse attention. Selection can be performed at the token level or, more fine-grainedly, at the head level, and the corresponding strategies may be either learning-based or training-free and applied only at inference time.

Query-agnostic Sparse Attention

Query-agnostic sparse attention methods, also referred to here as **token eviction** methods, determine whether a KV pair is retained based solely on information associated with that KV pair itself, without considering how other queries may attend to it. In causal LLMs, this makes it possible to remove KV pairs not covered by $\mathscr{I}$ from the KV cache during decoding, thereby maintaining a lower, possibly $O(1)$ memory footprint. The simplest choice to define $\mathscr{I}$ is through a predetermined and fixed sparse mask. A prominent example is **sliding-window attention** (SWA) in which each query attends only to the closest w tokens, and older KVs can be discarded during decoding. SWA is often combined with *global tokens*, e.g., the first s tokens of the input (Beltagy, Peters, and Cohan, 2020; Ainslie, Ontanon, et al., 2020; Zaheer et al., 2020), also known as the *attention sink* (Xiao, Tian, et al., 2023). As a result, a λ-shaped attention mask or sparsity pattern is formed, with $\mathscr{I}_i = \{j : j \leq s \vee i - w < j \leq i\}$. While simple and cheap, fixed patterns are input-agnostic and hence often suboptimal.

A more flexible approach is to dynamically predict whether the current KV pair $(\boldsymbol{k}_j, \boldsymbol{v}_j)$ will remain important for future queries. A common training-free strategy is to retain tokens that were important in the past, such as "heavy hitters" with high cumulative attention scores (Zhang, Sheng, et al., 2023). Alternatively, one may train a separate router to predict the future importance of each token. The main appeal of query-agnostic sparse attention is that it can control memory growth directly by enforcing a strict KV cache size budget and often admits simple implementations. However, without knowing how future queries would leverage the past context, useful information may be discarded prematurely.

Query-aware Sparse Attention

If we relax memory constraints and retain all past KVs, we can select $\mathscr{I}_i = \text{select}(\boldsymbol{q}_i, \boldsymbol{K}, \boldsymbol{V})$ based on each query, using a lightweight probing step before actual attention. For example, both the training-based Mixture of Block Attention (MoBA) (Lu, Jiang, et al., 2025) and the training-free Quest (Tang, Zhao, et al., 2024) partition the KV cache into blocks of size M. The keys within each block B_m are compressed into a block-level key $\boldsymbol{k}^B_m = \phi(\boldsymbol{k}_j), j \in B_m$ where ϕ is mean pooling. A block-level score $\text{score}(\boldsymbol{q}, B_m) = \langle \boldsymbol{q}, \boldsymbol{k}^B_m \rangle$ is then used as an approximation of the attention scores of keys within the block, used to select the top-K KV blocks to perform the actual attention. In this way, only MK of tokens are attended per query, which keeps the attention cost per step constant, while the blockwise computation is also favorable for efficient hardware implementation.

Query-aware mechanisms retrieve past tokens according to the current query, thereby avoiding the risk of permanently discarding useful information and improving the accuracy of long-range retrieval. However, these advantages come at the cost of higher memory usage, since the full KV cache must be retained, and additional computation for the probing step, which still requires scanning the cache. Even so, the approximate attention cannot always guarantee exact selection, which can in turn degrade performance.

2.3.3 Linear Attention

Instead of relying on direct pairwise attention over the sequence, another line of work revisits recurrent architectures that process context sequentially and compress it into fixed-size memory, known as linear attention [1]. A canonical starting point is vanilla linear attention (Katharopoulos et al., 2020), which can be simplified as a causal attention without softmax:

$$\boldsymbol{o}_i = \sum_{j=1}^{i} (\boldsymbol{q}_i^T \boldsymbol{k}_j) \boldsymbol{v}_j \tag{2.3}$$

$$= (\sum_{j=1}^{i} \boldsymbol{v}_j \boldsymbol{k}_j^T) \boldsymbol{q}_i = \boldsymbol{S}_i \boldsymbol{q}_i, \tag{2.4}$$

where the second form follows from associativity. Here, $\boldsymbol{S}_i = \sum_{j=1}^{i} \boldsymbol{v}_j \boldsymbol{k}_j^T = \boldsymbol{S}_{i-1} + \boldsymbol{v}_i \boldsymbol{k}_i^T$ is a matrix-valued hidden state that accumulates past information recurrently through outer products of keys and values, and therefore functions as an associative memory. The readout $\boldsymbol{o}_i$ is then obtained from $\boldsymbol{S}_i$ using the current query $\boldsymbol{q}_i$. Despite its expanded d^2 capacity, this recurrent process remains conceptually similar to a classical RNN. A key difference, however, is that classical RNNs use nonlinear state transitions, whereas linear transitions such as accumulation admit parallel computation across the sequence, much like standard attention, and therefore enable scalable training.

Later approaches introduce more sophisticated but still linear transition of $\boldsymbol{S}_i$, such as memory decay or forget gates, in order to reduce interference between stored KV pairs. These range from fixed decay rates, as in RWKV and RetNet (Peng, Alcaide, et al., 2023; Sun, Dong, et al., 2023), to input-dependent decay mechanisms, as in GLA (Yang, Wang, Shen, et al., 2024) and Mamba (Gu and Dao, 2024). A widely used recent variant is Mamba2 (Dao and Gu, 2024), which uses $\boldsymbol{S}_i = \gamma_i \boldsymbol{S}_{i-1} + \boldsymbol{v}_i \boldsymbol{k}_i^T$ with the decay rate γ_i is predicted from the current token.

These methods can be also interpreted as a form of online learning, in which $\boldsymbol{S}_i$ is a "fast weight" that is updated at each step for a linear projection applied to $\boldsymbol{q}_i$. For example, in Mamba2, the update to $\boldsymbol{S}_i$ can be interpreted as one gradient step on the loss $\mathscr{L}(\boldsymbol{S}) = -\langle \boldsymbol{S}\boldsymbol{k}_i, \boldsymbol{v}_i \rangle$ with learning rate $\beta = 1$, together with a weight decay term controlled by γ_i. The objective can be interpreted as promoting dot-product similarity between the label $\boldsymbol{v}_i$ and the prediction $\boldsymbol{S}\boldsymbol{k}_i$. A more straightforward approach is to use L2 similarity: by picking $\mathscr{L}(\boldsymbol{S}) = \frac{1}{2} \| \boldsymbol{S}\boldsymbol{k}_i - \boldsymbol{v}_i \|^2$ with learning rate β_i, DeltaNet employs the update

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} + \beta_i (\boldsymbol{v}_i - \boldsymbol{S}_{i-1} \boldsymbol{k}_i) \boldsymbol{k}_i^T \tag{2.5}$$

$$= (\boldsymbol{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^T) \boldsymbol{S}_{i-1} + \beta_i \boldsymbol{v}_i \boldsymbol{k}_i^T \tag{2.6}$$

which yields a diagonal-plus-low-rank transition matrix (Schlag, Irie, and Schmidhuber, 2021). Yang, Wang, Zhang, et al. (2024) further propose a chunkwise parallelizable implementation

[1] We use it as an umbrella term broadly covering various modern recurrent models, including state-space models.

of the delta update rule, and demonstrate its scalability. Building on DeltaNet, Gated DeltaNet (Yang, Kautz, and Hatamizadeh, 2024) incorporates Mamba-style forget gates, while the latest Kimi Delta Attention introduces per-channel gating (Kimi Team et al., 2025a). All of these methods enable more selective and compressive memory updates, and they have demonstrated performance comparable to, or in some cases stronger than, full attention on tasks such as language modeling and state tracking (Siems et al., 2025; Grazzi et al., 2025), while preserving the recurrent formulation. These models admit $O(1)$ time and space complexity and, in principle, an unbounded context window, as past context is compressed into fixed-size memory. This property is particularly attractive for the SLU settings studied in this thesis, where speech signals are often highly compressible. However, the lack of direct access to past tokens leads to deteriorated performance on tasks which demand retrieval quality, even at modest context lengths (Jelassi et al., 2024; Arora, Eyuboglu, Timalsina, et al., 2024; Arora, Eyuboglu, Zhang, et al., 2024; Wen, Dang, and Lyu, 2025), which highlights a challenge to resolve.

2.3.4 Hybrid Attention

No single token mixer offers the best trade-off between retrieval quality and computational efficiency. Full attention provides precise access to relevant context, but its computational cost is high. Linear attention is much more efficient and can compress long histories into fixed-size memory, but often performs poorly on tasks that require accurate retrieval of distant information. Sparse attention lies between these two extremes by allowing direct access to only a selected subset of past tokens, thereby improving retrieval accuracy while avoiding the full cost of dense attention. These complementary strengths have motivated a broad family of **hybrid attention** mechanisms that combine different token mixers within the same model. A common design principle is to delegate different functional roles to different parts of the model, as also observed to emerge in standard Transformers (Wu, Wang, Xiao, et al., 2024). One component may provide efficient but fuzzy long-range memory and strong short-range modeling through linear recurrence, while another preserves exact and direct retrieval through full or sparse attention. Such hybridization can be implemented by interleaving different layer types or, at a finer granularity, within a layer, for example across different heads (Irie, Yau, and Gershman, 2025; Dong, Fu, et al., 2025). Specifically, interleaving a few full attention layers within a linear attention backbone has emerged as an effective architectural design in recent LLMs like Qwen3.5 (Qwen Team, 2025). From this perspective, hybrid attention is not a single mechanism, but a general architectural principle for balancing compression, retrieval, and efficiency.

Hybrid attention is especially relevant in the present thesis because long-context spoken language understanding requires both efficient processing of extended sequences and reliable access to distant but important information. Designing mechanisms that achieve this balance more effectively remains an active research problem.

2.3.5 Efficient Attention for Speech

Speech differs from text in that it is represented as a high-resolution temporal signal, for example at 16 kHz in waveform samples or around 50 Hz in acoustic frames. As a result, speech sequences are typically much longer than text sequences representing the same content. Historically, tasks such as ASR were often assumed to require only limited long-range context. As a result, many speech models were designed with relatively restricted context windows, for example 30 seconds or 1500 frames in Whisper (Radford, Kim, et al., 2023). Longer inputs then had to be handled through chunking or sliding-window methods, which restrict direct access to earlier context. More recently, increasing evidence suggests that longer context can be beneficial (Hori et al., 2021; Flynn and Ragni, 2024; Chen, Kano, et al., 2024; Fox et al., 2024). This motivates mechanisms capable of handling extended speech recordings spanning minutes or even hours of audio, corresponding to tens of thousands of frames or more.

As a result, efficient attention mechanisms have become an active research direction in speech processing. Early work mainly explored static sparse attention and sliding-window attention for ASR (Wang, Sun, et al., 2021; Rekesh et al., 2023; Koluguri et al., 2024; Parcollet et al., 2024; Flynn and Ragni, 2024). With the rise of linear-attention architectures, later studies began to examine recurrent or state-space models on a broader range of speech tasks, including ASR, speech generation, and speech enhancement, although most evaluations remained focused on relatively short inputs (Goel, Gu, et al., 2022; Keyu An, 2023; Miyazaki, Murata, and Koriyama, 2023; Li, Chen, et al., 2024; Jiang, Li, Florea, et al., 2024; Miyazaki, Masuyama, and Murata, 2024; Gao and Chen, 2024; Park et al., 2024; Ali, Falavigna, and Brutti, 2025; Shakhadri, KR, and Angadi, 2025). More recent work has extended this line toward pretraining linear-attention-based speech foundation models (Yadav and Tan, 2024; Bhati, Gong, et al., 2024a; Lin, Kuo, et al., 2025; Zhang, Ma, et al., 2025) and toward speech LLMs built on efficient or hybrid backbones (Bhati, Gong, et al., 2024b; Qwen Team, 2026). Taken together, these works suggest that efficient attention is a promising direction for speech processing. However, evidence remains limited in realistic long-form settings and in complex LLM-based scenarios, which motivates the present thesis.

2.4 Interview Analysis

In the field of work psychology, behavioral job interviews have been established as a best practice for assessing applicants' competencies (Janz, 1989). They often use past-behavior questions, e.g., *Please tell me about an occasion where you had to deal with an unsatisfied client*, which invite applicants to give a detailed account (i.e. a story) of a past work-related event where they took action to resolve a problematic situation (Ralston, Kirkwood, and Burant, 2003). Successful storytelling in such interviews is a key factor in interview performance and hiring decisions (Stevens and Kristof, 1995; Bangerter, Corvalan, and Cavin, 2014). However, applicants often struggle to produce stories on demand and may opt for sub-optimal responses, like generic pseudo-stories or decontextualized self-descriptions (Bangerter, Cor-

valan, and Cavin, 2014; Brosy, Bangerter, and Mayor, 2016).

In recent years, new technology-driven interview formats have emerged, such as asynchronous video interviews (AVIs), in which applicants upload recorded answers to predefined questions through an online platform (Lukacik, Bourdage, and Roulin, 2022). This new form of interview has important implications: On the one hand, due to the lack of interactivity, applicants' responses may differ from face-to-face (FTF) settings (Germanier, Bangerter, Orji, et al., 2025), and some traditional approaches to facilitate applicants' responses (e.g., probe questions) become impossible in AVIs. On the other hand, AVIs are more amenable to the recording and automatic analysis of applicants' responses. Such analysis falls within the scope of SLU and may be supported by emerging machine learning and AI approaches. Such techniques have been applied both to assisting recruiters in hiring decisions (e.g., Hickman, Bosch, et al. (2022) and Liff et al. (2024)), as well as providing feedback and training to applicants seeking to improve their interview performance (Gebhard et al., 2019; Hoque et al., 2013; Langer et al., 2016). Particularly, Bangerter, Mayor, et al. (2023) apply machine learning methods to automatically identify storytelling responses to transcripts of FTF interviews in order to identify storytelling responses to past-behavior questions, with the goal of helping interviewees improve their storytelling. Nevertheless, the use of AI for interview analysis remains relatively underexplored, especially in the context of AVIs, storytelling analysis, and end-to-end approaches that operate directly on speech recordings. As a complex understanding task on lengthy speech recordings, this task provides a natural testbed for the methods investigated in the present thesis from a technical perspective.

3 Enhancing Spoken Language Understanding via Speech Translation

End-to-end spoken language understanding (SLU) remains elusive even with current large pretrained language models on text and speech, especially in multilingual cases. A potential bottleneck is the lack of semantic supervision during pretraining, as the speech foundation models are mostly trained by self-supervised objectives on (sub-)phonetic units. Machine translation has been established as a powerful pretraining objective on text as it enables the model to capture high-level semantics of the input utterance and associations between different languages, which is desired for speech models that work on lower-level acoustic frames.

Motivated particularly by the task of cross- and multi-lingual SLU potentially valuable for our application in French interview analysis, this chapter demonstrates that the task of speech translation (ST) is a good means of pretraining speech models for end-to-end SLU on both intra-and cross-lingual scenarios. By introducing ST, our models reach higher performance over baselines on monolingual and multilingual intent classification as well as spoken question answering using SLURP, MINDS-14, and NMSQA benchmarks. To verify the effectiveness of our methods, we also create new benchmark datasets from both synthetic and real sources, for speech summarization and low-resource/zero-shot transfer from English to French or Spanish. We further show the value of preserving knowledge for the ST pretraining task for better downstream performance, possibly using Bayesian transfer regularizers.

The work in this chapter has been published as:

Mutian He and Philip Garner (Dec. 2023a). "The Interpreter Understands Your Meaning: End-to-end Spoken Language Understanding Aided by Speech Translation". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 4408–4423. DOI: 10.18653/v1/2023.findings-emnlp.291

3.1 Introduction

Modern artificial intelligence is characterized by large pretrained language models (PTLMs) with strong language capabilities to be adapted to various downstream tasks. The success of PTLMs rests on carefully-designed pretraining tasks to bestow the capability we expect on the model. Current PTLMs are mostly trained on self-supervised tasks, which started from masked language modelling (MLM) and next sentence prediction (NSP) in BERT (Devlin et al., 2019), but recently evolved into more difficult ones such as whole word (Cui et al., 2021) or span masking (Joshi et al., 2020), text infilling, and token deletion (Lewis et al., 2020). While the rather simple NSP has been replaced by sentence permutation, document rotation (Lewis et al., 2020), and sentence order prediction (Lan, Chen, et al., 2020). All those efforts introduced more challenges in the pretraining phase to mine stronger semantic supervision signals out of unlabelled data.

Such semantic-rich supervision is particularly relevant for pretrained spoken language models like wav2vec2 (Baevski et al., 2020) and HuBERT (Hsu, Bolte, et al., 2021) based on MLM on (sub-)phonetic units from lower-level audio signals, which are less informative and require models to carry out additional labor on acoustics. Therefore, their high-level capacities are more restricted. This may explain why automatic speech recognition (ASR) models fine-tuned upon them with paired data still have a role in fully end-to-end (E2E) SLU, often as a pretrained feature extractor (Seo, Kwak, and Lee, 2022; Arora, Dalmia, Denisov, et al., 2022). Unlike the cascaded SLU in which ASR produces transcripts for text processing, in such E2E systems ASR as an auxiliary or additional pretraining task provides strong supervision to explicitly link audio to representations that correspond to the denser and semantic-richer textual space, which is valuable for downstream understanding tasks.

On texts, self-supervised objectives are rather effective thanks to enormous text data with high information density, but supervised tasks are still used in many cases, machine translation (MT) being an often-seen one. A pioneer of the current PTLM paradigm, CoVe (McCann et al., 2017), is a seq2seq model pretrained on MT that achieved the then state-of-the-art on various downstream tasks. Belinkov et al. (2020) further validate language capabilities of MT on morphological, syntactic, and semantic levels, and T5 (Raffel et al., 2019) uses an ensemble of supervised tasks including MT. Furthermore, when trained with inputs of multiple languages, the model encoder may align and push representations for inputs in different languages with similar meaning together to have the same output in the target language, thanks to the guidance from paired data (Johnson et al., 2017; Schwenk and Douze, 2017). With this semantic-centric language agnosticity, such an encoder can achieve few/zero-shot transfer to another language in downstream tasks (Eriguchi et al., 2018).

Inspired by those works, we hypothesize that the counterpart of multilingual MT on speech, i.e., E2E multilingual speech translation (ST) that directly maps speech of various languages to texts in other languages, will also be effective as a pretraining task on E2E SLU, for three critical advantages:

1. It requires high-level understanding as an interpreter must "understand" the utterance before interpreting it into a different language, unlike ASR that transcribes speech verbatim and MLM on phonetic units that needs less semantic understanding.

2. It captures long-term dependency and a global view of the full input, in contrast to ASR and MLM which can often be resolved with local context.

3. It enables better cross-lingual transfer in comparison with multilingual ASR models and self-supervised PTLMs without the supervision that promotes language agnosticity.

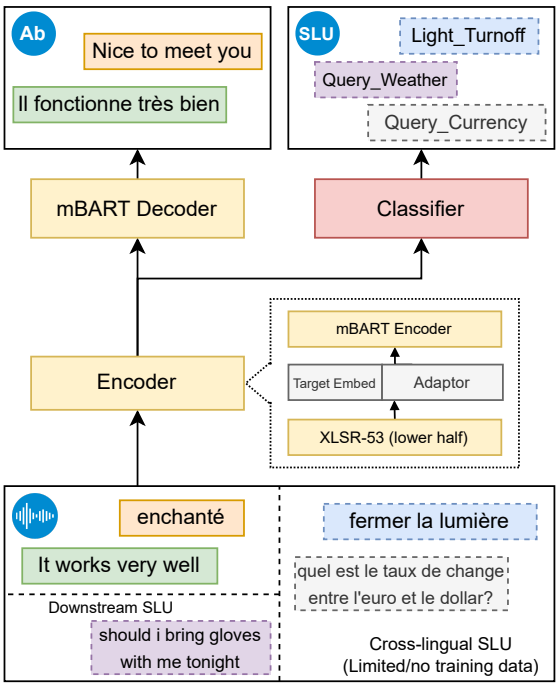


Figure 3.1: Our framework of ST-aided SLU, by connecting pretrained XLSR and mBART fine-tuned for ST following Li, Wang, et al. (2021), and then reusing the ST encoder, transferred to downstream SLU tasks like intent classification with a stacked classifier also from PTLMs.

Admittedly, ST data is only available in a limited number of language pairs, but for each covered language, there are infinite number of diverse downstream SLU tasks with only rich data in English. It is a practical need to enroll various such languages to an English-only model trained on each specific SLU task. Therefore, as shown in Figure 3.1, we may pretrain the model on speech translation between English and the target language French in both directions (i.e. En↔Fr), and then fine-tune on downstream tasks with an additional classifier, reusing the encoder. We show the benefit of our method on a variety of tasks for semantic understanding of speech, including mono- & multilingual intent classification (IC), spoken question answering (SQA), as well as speech summarization, for which we create a synthetic dataset following Huang, Rao, et al. (2022). Then we show the strong advantage on cross-lingual transfer to French. All the experiments are focused on comparing ST with other tasks like ASR as the pretraining or auxiliary task to verify our core hypothesis above. In addition, to show that our method applies to other languages as well, we also conducted experiments using Spanish as the target language. This is evaluated by creating the French and Spanish version of the English IC benchmark SLURP (Bastianelli et al., 2020), using both real and synthetic sources. On all the tasks, our approach outperforms previous baselines and ASR pretraining, often by a large margin.

Furthermore, unlike knowledge for self-supervised objectives loosely connected to target SLU tasks, knowledge to handle tasks with closer link to semantics such as ST will be more valuable, following our core hypothesis. Hence it should be helpful to preserve such knowledge instead of direct fine-tuning with the risk of catastrophic forgetting. Therefore, we introduce

multi-task learning as well as Bayesian regularizers for knowledge preservation, namely L2-SP (Li, Grandvalet, and Davoine, 2018) and EWC (Kirkpatrick et al., 2017), which show benefits especially in low-resource cases.

To summarize, our contributions are three-fold:

1. We demonstrate the effectiveness of speech translation pretraining on multiple SLU tasks, especially in cross-lingual transfer cases.
2. We confirm the value of preserving ST pretraining knowledge for downstream tasks and the capability of Bayesian regularizers to achieve that.
3. We build several new datasets for speech summarization and cross-lingual SLU.

Our code, models, and datasets are released at https://github.com/idiap/translation-aided-slu.

3.2 Model Pretraining

As in Figure 3.1, we first build a speech translator using an architecture established by Li, Wang, et al. (2021), that connects pretrained models on speech and text with a CNN adaptor: Audio signals are fed into the lower half of the multilingual wav2vec2, XLSR-53 (Conneau, Baevski, et al., 2021), to extract the (sub-)phonetic representations into a 320x-downsampled sequence. The upper half (12 layers) of XLSR is discarded for computational efficiency, as those parameters are found focused on MLM pretraining and less useful for downstream tasks (Zhu, Wang, Cheng, et al., 2022). Given the phonetic level embeddings produced by the half XLSR, the task is similar to machine translation to map them to the output text, for which we leverage the MT model based on mBART (Liu, Gu, et al., 2020). While the length of this sequence is still much longer than the corresponding text. To better align it with typical textual embeddings as in mBART inputs, a 3-layer 8x-downsampling CNN adaptor is inserted. A *target embedding* is then prepended to specify the target language or task, similar to the target token used in mBART. To promote language agnosticity, we do not indicate the source language. Furthermore, it has been found that explicitly promoting language agnosticity may help zero-shot transfer (Arivazhagan et al., 2019), hence we've also attempted to add language adversarial training on the encoder outputs during pretraining and fine-tuning, using a language classifier of a 2-layer MLP to predict the language of the input speech, with gradient reversal layer to explicitly align the representations between different languages.

Based on the architecture, we fine-tune the model using a combination of the En→Fr portion of MuST-C (Di Gangi et al., 2019), and the Fr→En portion of TEDx (Salesky et al., 2021), both derived from TED talks, plus the Fr→En portion of CoVoST2 (Wang, Wu, et al., 2021) based on general sentences in Common Voice (Ardila et al., 2020), with texts further cleaned and sentences that are too long or contain foreign characters removed. Unlike Li, Wang, et al. (2021), the whole model is fine-tuned for best pretraining results. To compare, we also experiment

ASR WER↓	TEDx	MuST-C	CoVoST2
ASR	16.58%	8.62%	13.67%
ASR+ST	15.82%	8.28%	13.62%
ST BLEU↑			
ST	29.27%	36.30%	31.34%
ASR+ST	31.19%	37.18%	31.93%

Table 3.1: Test results on the cleaned pretraining datasets given by word error rate (WER) for ASR and BLEU score for ST, with French inputs for TEDx and CoVoST2, and English inputs for MuST-C.

with pretraining on the task of ASR instead. As the data are paired with both translations and transcripts, we use the same ST dataset for ASR training to build a multilingual (En+Fr) ASR model. We've tried to jointly train on ASR+ST in a multi-task manner as well. With a total of >700 hours paired speech data, we achieve satisfactory results on the pretraining tasks as indicated in Table 3.1, and ASR+ST training shows better performance compared to the single-task ones. Starting from the ASR and ST models, we further add Spanish portion from the same set of ST datasets. As a result, we obtain an ST model supporting both En↔Fr and En↔Es (Spanish), and a tri-lingual En+Fr+Es ASR model, both with similar satisfactory results, details available in Appendix A.5.

3.3 Downstream Adaptation

3.3.1 Tasks

We then fine-tune the whole model on a variety of direct downstream tasks as follows.

SLURP is a large and challenging English SLU dataset recently proposed, with 72.2k real speech recordings and 69.3k synthetic audio for a broad range of speech commands given to voice assistants. We use its IC labels to classify the input into 18 scenarios and 46 actions.

MINDS-14 is a multilingual IC dataset for banking scenarios with 14 types of intents in 14 languages with around 600 utterances per language, and we use four subsets (en-AU, en-GB, en-US, and fr-FR) under a 3:2:5 train-dev-test split in XTREME-S (Conneau, Bapna, et al., 2022). The rather scarce training data demand data-efficient multilingual modelling.

NMSQA or Natural Multi-Speaker Question Answering is a spoken QA dataset consisting of audio for the questions and segmented context articles from SQuAD (Rajpurkar et al., 2016), with 97.6k question-answer pairs given in >300 hours of synthetic audio from 12 speakers produced by Amazon TTS, coupled with a 60-speaker real test set of 2.7 hours of recordings. In this task, the goal is similar to textual QA to predict the correct span in the spoken context audio that answers the question, and the performance is measured by Audio Overlapping

Score (AOS) (Lee, Wu, et al., 2018), defined as AOS = $X \cap Y / X \cup Y$, in which X is the predicted audio span and Y the ground truth.

Spoken Gigaword is the synthetic spoken version of the summarization or headline generation task on Gigaword (Rush, Chopra, and Weston, 2015), proposed by Huang, Rao, et al. (2022), aimed at generating a brief headline from a short piece of English spoken news. As it was not released, we follow their method to filter the data and create a synthetic dataset of 131.5 hours of audio using Google TTS from 9 neural voices in en-US, with 50k training samples, 1k validation samples, and 385 test samples, as a result of filtering out the frequent noise in the test set.

Synthetic data are used in those established datasets for training and evaluation. Despite being possibly different from real data, it is observed that they are often reliable to reflect model performances and well correlated with real cases.

3.3.2 Methods

For these downstream tasks, we reuse the encoder further pretrained on ST/ASR (with French, unless otherwise stated). It should be noted that the 12-layer mBART encoder we use is slightly smaller than half XLSR. So when connected, the total encoder size and the computational cost to fine-tune it is comparable with fine-tuning the whole original XLSR. Upon the encoder we stack a 3-layer transformer, which is also transferred from a PTLM. As for IC, we use layer 2-4 from pretrained XLM-R (Conneau and Lample, 2019) for possibly better understanding capabilities, stacked with linear classifier heads over mean-pooled outputs. Particularly, for SLURP in which the intent consists of a scenario and an action, two heads are used. As for SQA, we use layer 2-4 of pretrained Longformer (Beltagy, Peters, and Cohan, 2020), a PTLM dedicated for long utterances due to the length of each segment in the data, as in Lin, Chuang, et al. (2022). Two linear classifiers are then applied to each frame to predict the start and end of the span, along with an answer existence classifier over mean-pooled outputs to predict if the answer exists in the provided segment. We then concatenate the question audio with each segment in the spoken article as model inputs, and pick the predicted answer span from the segment with the highest answer existence likelihood. For these two tasks, the pretrained decoder is simply discarded. While speech summarization is more similar to ASR and distinct from other downstream tasks that the model first needs to capture general meaning of the speech as encoded representations, and then generate a textual summary by the decoder, which demands a seq2seq architecture identical to the ST/ASR pretraining task. Hence we reuse the whole encoder-decoder model and formulate the task as generation in an extra "target language". With the needs to both understand the general meaning and generate in the same language, we hypothesize that combining ASR and ST will lead to the best results.

Furthermore, as mentioned above, direct model fine-tuning may lead to catastrophic forgetting of the knowledge on ASR or ST and harm semantic understanding capabilities. Hence we also tried a multi-task ***joint training*** approach on both the pretraining and target task. Results

Pretraining task	Accuracy↑
ASR	87.38%
w/ Joint training	88.37%
ST	87.84%
w/ Joint training	89.35%
w/ Joint training + Es	**89.59%**
ST+ASR	87.75%
w/ Joint training	89.43%
None	84.80%
Whisper	80.39%
ESPnet-SLU (Arora, Dalmia, Denisov, et al., 2022)	86.30%
CTI (Seo, Kwak, and Lee, 2022)	86.92%
Generative IC+SF (Wang, Boumadane, and Heba, 2022)	
based on wav2vec2	87.13%
based on HuBERT	89.38%
CIF-PT (Dong, An, et al., 2023)	91.43%

Table 3.2: SLURP test results of our models, fine-tuned from wav2vec2, compared to baselines without additional supervised pretraining or reusing Whisper as the encoder, as well as results reported in literatures.

are compared between the model pretrained with ST, ASR, or both, or one directly derived from self-supervised pretraining without further supervision (***None***), plus other baselines. More, the recent Whisper (Radford, Kim, et al., 2023) is trained on multiple speech tasks including ASR and ST, which matches our idea despite not aiming at SLU. Hence we also try to fine-tune the Whisper encoder, using the *medium* version with size similar to our encoder.

Pretraining task	en-AU	en-GB	en-US	fr-FR	Average
ASR	95.7%	97.3%	96.5%	95.2%	96.2%
w/ Joint training	96.3%	98.3%	98.2%	93.7%	96.6%
ST	96.9%	**99.0%**	98.2%	97.8%	98.0%
w/ Joint training	**97.3%**	98.7%	**99.3%**	98.2%	**98.3%**
ST+ASR	95.4%	98.3%	97.5%	95.6%	96.7%
w/ Joint training	96.3%	98.3%	98.9%	**98.5%**	98.0%
XLSR (Lozhkov, 2022)	92.4%	93.2%	93.3%	94.4%	93.3%

Table 3.3: Test accuracies for models on MINDS-14 multilingual IC, comparing with directly fine-tuning the full XLSR model. Both ST pretraining and joint training show benefits.

Pretraining task	*dev*	*test*
ASR	54.6%	53.0%
ST	**58.2%**	**59.4%**
ST+ASR	57.8%	58.0%
DUAL E2E	48.5%	49.1%
- Cascaded	58.3%	57.4%
ByT5lephone E2E	-	53.3%
- Cascaded	59.2%	70.5%

Table 3.4: AOS (↑) scores for models on NMSQA, compared to baselines reported in DUAL (Lin, Chuang, et al., 2022) as well as the much larger ByT5lephone model (Hsu, Chung, et al., 2023). The pretraining tasks prove helpful, particularly ST pretraining, which may reach performance close or better certain cascaded system.

3.3.3 Results

English IC Following the previous works, we report the test accuracy on SLURP as in Table 3.2. It can be observed that the models with ST pretraining outperform those trained on ASR only, while adding ASR to ST pretraining makes limited improvements, though it gives better WER and BLEU during pretraining; it is the same case for the model with the extra Spanish ST task introduced in pretraining. However, ASR does help considering the *None* model directly fine-tuned from self-supervised PTLMs without any additional pretraining. By joint training with both the pretraining and downstream task, results are consistently improved. However, despite being a strong ASR+ST model, Whisper is found not suitable for fine-tuning on SLURP in this way as shown by the low accuracy. This might be explained by the fact that Whisper is trained on En ASR and X→En ST but not for En→X ST. Our hypothesis is that the ST pretraining on a specific language would enhance the semantic understanding capabilities of the model in that language, which may not help Whisper much on the English SLURP benchmark. Also, Whisper is trained on 30-second chunks, while SLURP contains more shorter utterances.

HuBERT, used in multiple baselines in Table 3.2, has been found stronger on various downstream tasks compared to wav2vec2. Owing to the lack of a multilingual HuBERT (large) model, we rely on the multilingual wav2vec2 as our acoustic encoder. However, we reach much better results compared to many notable baselines, including the approach of jointly generating the intents and slots (Wang, Boumadane, and Heba, 2022), with 87.13% accuracy, the highest among wav2vec2-based baselines. We also reach slightly higher accuracy than its HuBERT version, which was the previous state-of-the-art. The very recent CIF-PT (Dong, An, et al., 2023), concurrent with ours also injects more semantic signal, but by learning frame-to-token alignments on the encoder and then distilling from PTLMs, significantly pushing the state-of-the-art on this monolingual benchmark. Nevertheless the method is distinct from ours, raising the possibility of applying both methods orthogonally for further improvement, and we maintain advantages on cross-lingual transfer and possibly also generative tasks by reusing a pretrained seq2seq decoder, as elaborated below.

Multilingual IC We then report the accuracy on MINDS-14 as in Table 3.3 on four languages plus the average accuracy across languages, compared to a baseline directly fine-tuned from XLSR. The results are consistent with the monolingual case that ST pretraining can significantly improve the performance on SLU tasks, that joint training is beneficial, and that adding ASR gives limited gains.

	dev			*test*		
Joint task	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ASR	40.16	18.39	37.69	35.90	16.25	33.77
ST	40.61	18.95	38.23	36.70	16.39	34.47
ST+ASR	**41.39**	**19.50**	**38.83**	**37.63**	**17.80**	**35.20**
None	21.49	7.44	20.37	18.39	6.16	17.41
Cascaded	**42.00**	**21.42**	**39.60**	32.24	15.03	30.14

Table 3.5: ROUGE (↑) scores for models on Spoken Gigaword speech summarization. ST still proves beneficial, while best results could be obtained by combining ASR for this task of generating summaries in the same language.

Spoken QA We compare our methods with results reported by Lin, Chuang, et al. (2022), including the results from a cascaded pipeline that fine-tunes Longformer upon transcripts from wav2vec2-based ASR, and the DUAL approach that fine-tunes Longformer upon units pre-extracted by a frozen HuBERT, hence not fully end-to-end. For fair comparison, we fine-tune the classifier built by layers 2–4 of Longformer and the top 5 layers of the mBART encoder, while the rest of the model is frozen and used as a feature extractor, so that they have a comparable number of trainable parameters with the baselines. Therefore we do not conduct experiments on joint training in this task as most shared parameters are frozen. The results reported in the more recent T5lephone (Hsu, Chung, et al., 2023) including the E2E and cascaded approach are also mentioned, though they are almost twice as large as other models. All the baselines enjoy a view of the whole article, while in our experiments we use a model that works on a shorter context window with the question and each segment in the article individually, in order to have an end-to-end architecture given our computational resources that is consistent with other experiments. Therefore, the baselines possess a strong advantage over ours. However, as shown in Table 3.4, the additional pretraining stage leads to better results compared to all the E2E baselines, which further demonstrates the advantage of our approach. Particularly, ST considerably improves the performance and could successfully beat the cascaded system reported by Lin, Chuang, et al. (2022) in the more challenging *test* portion.

Speech summarization We report the results compared between different auxiliary tasks as in Table 3.5 using the ROUGE-1/2/L metrics (Lin, 2004). In the experiments, we observed that simply fine-tuning the model rapidly leads to overfitting, hence we perform joint-training only, and use a special target embedding to indicate the summarization task. ASR is helpful

	Full			100-shot			Zero-shot		
Pretrain task	*dev*	*test*	*real*	*dev*	*test*	*real*	*dev*	*test*	*real*
ASR	83.3%	84.0%	79.7%	69.6%	69.0%	71.3%	39.0%	39.8%	39.4%
ST	85.2%	**86.1%**	**84.9%**	78.1%	77.0%	79.0%	58.9%	58.9%	56.6%
ST+ASR	85.8%	85.7%	82.4%	78.0%	77.0%	78.8%	63.9%	62.6%	59.1%
ST+Adv.	**86.4%**	84.9%	84.1%	**78.3%**	**78.1%**	**80.9%**	**67.0%**	**67.7%**	**63.7%**
None	75.9%	74.0%	65.4%	57.5%	52.4%	53.5%	15.3%	16.1%	13.6%

Table 3.6: Results on SLURP-Fr cross-lingual IC transferred from SLURP with different data amounts. Comparing different supervised pretraining (or w/o additional supervision), results highlight ST and adversarial training.

		None	ASR	ST
Full	*dev*	75.6%	83.7%	85.6%
	test	75.1%	82.5%	84.5%
100-shot	*dev*	64.4%	76.5%	80.3%
	test	63.8%	75.1%	79.6%
Zero-shot	*dev*	12.3%	39.7%	55.2%
	test	12.8%	39.1%	54.4%

Table 3.7: Results on SLURP-Es cross-lingual IC transferred from SLURP with different data amounts.

on the summarization task as the ST+ASR model consistently outperforms the ST one, while the ST one is still better than the ASR-only model, signifying the importance of the semantic understanding capability brought by ST pretraining. In addition, we compare with a cascaded baseline that first transcribes the inputs with our ASR model, which introduces WER of 9.1% and 8.9% on *dev* and *test* respectively. Then we leverage a BART-based model fine-tuned on the full textual Gigaword with ROUGE-1/2/L=37.28/18.58/34.53 to produce the summaries. When applied to the relatively simple utterances in Spoken Gigaword, it reaches a higher performance on *dev*, which suggests the challenges for E2E systems in our benchmark, though the gap is narrow compared to our E2E approach with ST+ASR pretraining, and on the noisier *test* set our E2E models consistently get much better results.

3.4 Cross-lingual Transfer

For cross-lingual transfer, IC models trained on SLURP are then applied on/fine-tuned to French/Spanish data below:

Datasets A French version of SLURP, **SLURP-Fr**, is created to evaluate the cross-lingual transfer capabilities of the model, which is based on MASSIVE (FitzGerald et al., 2023), a translation of SLURP texts into multiple languages. With the same input domain and output categories, zero-

shot transfer becomes possible. We first produce the audio for the 16.5k French samples in MASSIVE with a 7:2:1 train-dev-test split using Google TTS from four different WaveNet-based speakers. Then we invite two native French speakers to read out a total of 477 randomly-selected category-balanced held-out utterances, forming the *real* test set. To mimic SLURP, we record the audio indoors with two microphones under both near-field and far-field conditions. We also define a 100-shot per category subset with 4.5k samples in total to simulate a condition with even lower resource. **SLURP-Es** is created in a way similar to SLURP-Fr with 16.5k Spanish samples in MASSIVE, though we are unable to create a *real* set.

Experiments The advantage of our method on cross-lingual transfer is evaluated under the full-data, 100-shot, and zero-shot cases, using different pretraining strategies compared to the *None* model trained on SLURP without further supervision but directly upon the multilingual self-supervised pretrained models. Hence it is noteworthy that all the compared models have been pretrained in a multilingual way. As given in Table 3.6 on French, extra multilingual ST/ASR supervision consistently leads to better results on different data amounts. ST pretraining outperforms ASR, similar to previous experiments, while ST+ASR joint pretraining brings some improvements in the zero-shot case. Notably, the gap between ASR and ST models becomes larger with fewer data, especially with zero shot, which implies the importance of ST on cross-lingual transfer. Accuracy on the real near-field speech is reported, which correlates well with those on synthetic ones, indicating that performance on synthetic speech is reliable for evaluation. The *ST+Adv.* model incorporates language adversarial training during pretraining and fine-tuning to further promote language agnosticity as mentioned above, which outperforms other models in most cases, particularly with zero shot, implying the usefulness of language adversarial training and the importance of language agnosticity of input features to the classifier. In addition, we build a cascaded system which first translates the speech into English text using our ST model, with BLEU scores of 26.08/26.34/21.56 on dev/test/real. Then a BART-based textual SLURP model with 85.7% test accuracy is used. This essentially zero-shot system gives a competitive 62.2% *real* accuracy, which poses challenges to the future development of E2E cross-lingual models. The model on Spanish, based on En+Fr+Es ST/ASR pretraining along with training on SLURP in an identical protocol, shows similar results as in Table 3.7, indicating the applicability of our approach to other languages.

3.5 Pretraining Knowledge Preservation

Methods As mentioned above, the knowledge to perform ASR/ST that connects speech and semantic-rich texts could be valuable for downstream tasks, which motivates us to use *joint training* above that maintains the performance on the pretraining tasks. This is verified by the considerable performance improvement or gap between the joint and single models. However, it is computationally intensive and requires access to the pretraining data. To alleviate the gap without joint training, we intend to explore Bayesian transfer/continual learning regularizers that limit the parameter shift by applying a prior on the parameters,

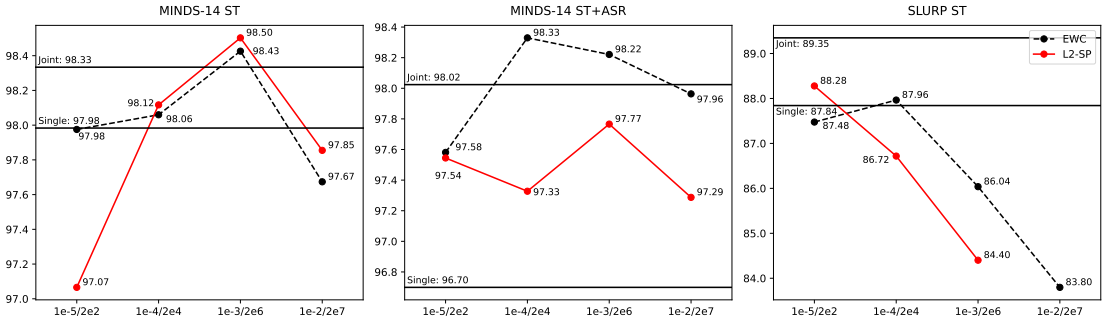


Figure 3.2: Results for Bayesian transfer regularizers when applied to different tasks, with the goal of mitigating the gap between the performance of single-task and joint-training models, indicated by the lower and the upper horizontal lines. The x-axis indicates the regularization weight for EWC/L2-SP, and the y-axis the accuracy. The regularizers bring positive effects on the data-scarce MINDS-14 task, but not on SLURP.

based on the Laplacian approximation of posterior parameter distributions in pretraining (MacKay, 1992). Particularly, the L2-SP method formulate the prior as an isotropic Gaussian distribution with the pretrained parameters $\boldsymbol{\theta}_0$ as the mean and identical variance for all parameters, which leads to an L2 regularization term with weight α centered at $\boldsymbol{\theta}_0$ in the loss for the maximum likelihood estimation of the parameters (Li, Grandvalet, and Davoine, 2018). While elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) considers the variance of each parameter θ_i decided by the Fisher diagonal F_i, which could be further estimated using squared gradients by averaging over the stochastic gradient descent (SGD) trajectory. However, for optimization we use the Adam algorithm

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha \cdot \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon), \tag{3.1}$$

that already computes $\hat{\boldsymbol{v}}_t$, an exponential moving average of squared gradients (Kingma and Ba, 2017), close to linear averaging with a smoothing parameter $\beta_2 = 0.999$. Hence we reuse them to set the per-parameter weight αF_i for regularization. For both methods, the hyperparameter α is used to control the strength of the knowledge preservation or the restraint to the parameter update. See Appendix A.1 for more theoretical explanations.

Experiments We experiment with these regularizers, targeted on ST-pretraining on SLURP and MINDS-14, plus the ST+ASR pretraining on MINDS-14 which has a considerable 1.32% accuracy gap. We try to use various weights α for L2-SP regularization ranging from 1e-5 to 1e-2. Then we inspect the distribution of the approximated F_i, which ranges from 1e-20 to 1e-5 as in Appendix A.6. For optimization stability we clamp the weight αF_i above 1e-2, and use EWC weights of 2e2, 2e4, 2e6, and 2e7 to roughly match the magnitude of those for the L2-SP regularizer.

Results are shown in Figure 3.2, and for MINDS-14 the average accuracies are reported. In the case of SLURP, it is possible that the amount of data is already sufficient that the preservation

of the pretraining knowledge could be helpful only if it is carried out in a fully adaptive way, namely joint training. Therefore, the regularizers lead to limited help or even harm to the accuracy when the weight is large. However, under the low-resource condition in MINDS-14, both regularizers are effective. As in Li, Grandvalet, and Davoine (2018), although being more flexible and adaptive, EWC doesn't necessarily lead to better transfer learning. This is consistent with our observations: Both regularizers can successfully overcome the accuracy gap or even go beyond the joint training model under an appropriate weight, while the best regularizer varies in different cases, though the more adaptive EWC has a chance to reach better results as in the MINDS-14 ST+ASR case. In this way, we demonstrate the effectiveness of Bayesian parameter-preserving regularizers for transfer learning on such large pretrained models.

3.6 Related Work

Translation as an auxiliary task It has been found that representations from MT models capture various aspects of the input utterance such as syntax (Shi, Padhi, and Knight, 2016), morphology (Belinkov et al., 2017), and also semantic inferences (Poliak et al., 2018; Belinkov et al., 2020). Hence MT has been established as a pretraining task as in CoVe (McCann et al., 2017) for various downstream tasks. But unlike our work, recent works on the direction has been focused on multilingual and cross-lingual cases, starting from attempts to reuse MT representations as sentence embeddings for text classification (Shi, Padhi, and Knight, 2016; Lu, Keung, et al., 2018), and, particularly often, for semantic similarity and bi-text mining (Schwenk and Douze, 2017; Vázquez et al., 2019; Raganato et al., 2019; Artetxe and Schwenk, 2019). As for pretraining PTLMs to be fine-tuned, MT proves effective for downstream cross-lingual tasks on few-shot and zero-shot transfer (Eriguchi et al., 2018), while often accompanied with similar tasks like translation language modelling (Conneau and Lample, 2019; Kale, Siddhant, et al., 2021), cross-lingual MLM (Chi, Dong, Ma, et al., 2021), and dictionary denoising (Reid and Artetxe, 2022). Particularly, MT has been used as an auxiliary task for cross-lingual intent classification on texts (Schuster et al., 2019; Siddhant et al., 2020; van der Goot et al., 2021), and is widely used on cross-lingual generation, including summarization (Zhu, Wang, Wang, et al., 2019; Cao et al., 2020; Xu et al., 2020; Takase and Okazaki, 2022), simplification (Mallinson, Sennrich, and Lapata, 2020), question generation (Chi, Dong, Wei, et al., 2020), and data-to-text generation (Kale and Roy, 2020).

Bayesian transfer learning Viewing the pretrained model not as a point estimation but a distribution is critical for continual learning as in EWC (Kirkpatrick et al., 2017), and the idea has been also applied to transfer learning to regularize fine-tuning as in L2-SP for image classification (Li, Grandvalet, and Davoine, 2018), though similar regularizers have been used on MT (Miceli Barone et al., 2017) and ASR (Liao, 2013). More recently, Shwartz-Ziv et al. (2022) propose to approximate the prior using SGD trajectory as in SWAG (Maddox et al., 2019) for transfer learning.

End-to-end SLU As introduced in Chapter 2.2, recently end-to-end methods and PTLM-based methods have gained popularity in SLU. Besides directly fine-tuning existing PTLMs on speech (Wang, Boumadane, and Heba, 2022; Arora, Dalmia, Denisov, et al., 2022), there are also explorations for end-to-end interface to connect pretrained models on speech and text (Saxon et al., 2021; Seo, Kwak, and Lee, 2022; Raju et al., 2022), as well as joint speech-text modelling, pretraining, or distillation (Chuang et al., 2020; Chung, Zhu, and Zeng, 2021; Kim, Kim, et al., 2021; Villatoro-Tello et al., 2023; Dong, An, et al., 2023), prompt tuning for PTLMs (Gao, Ni, et al., 2022; Chang, Tseng, et al., 2022), combining PTLM features (Cheng, Zhu, et al., 2023), and multitask learning with ASR (Huang, Rao, et al., 2022), which is most relevant to this work.

3.7 Conclusion

We confirm our hypothesis that speech translation can be a powerful pretraining and joint-training means for various end-to-end models on tasks involving semantic understanding of speech. Particularly, it benefits multilingual scenarios and cross-lingual transfer, including the zero-shot case. We also create two new datasets for the above tasks. Furthermore, we demonstrate the effectiveness of Bayesian regularizers to preserve the knowledge from pre-training for downstream tasks. This provides important insights for our research on SLU that highlights the value of semantic-heavy pretraining tasks. This also motivates our subsequent investigation on the use of the more semantically capable LLMs pretrained on text.

4 Spoken Language Understanding via LLMs

The previous chapter falls under the pretraining+fine-tuning paradigm, where the generalist model still requires further fine-tuning before it can be effectively applied to specific downstream tasks. Following GPT-3 and the release of ChatGPT in late 2022, LLMs have demonstrated strong language understanding capabilities. This is particularly reflected in their zero-shot and in-context learning abilities on downstream tasks through prompting. To assess their impact on spoken language understanding (SLU), in this chapter we evaluate several such models like ChatGPT and OPT of different sizes on multiple benchmarks. We verify the emergent ability unique to the largest models as they can reach intent classification accuracy close to that of supervised models with zero or few shots on various languages given oracle transcripts. By contrast, the results for smaller models fitting a single GPU fall far behind. We note that the error cases often arise from the annotation scheme of the dataset; responses from ChatGPT are still reasonable. We show, however, that the model is worse at slot filling, and its performance is sensitive to ASR errors, suggesting potential challenges for the application of those textual models on SLU. All of the results indicate that LLMs can be powerful for SLU, while the exact methods to leverage them remain to be explored.

The work in this chapter has been published as:

Mutian He and Philip N. Garner (2023b). "Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding". In: *Proceedings of the 24th Annual Conference of the International Speech Communication Association, Interspeech 2023*, pp. 1109–1113. DOI: 10.21437/Interspeech.2023-1799

4.1 Introduction

Gigantic pretrained language models like GPT3 with 175B parameters trained on 45TB texts have demonstrated surprisingly strong performance on various downstream language tasks with little or no data (Brown, Mann, et al., 2020). Since then, GPT3 has evolved into GPT3.5 through pretraining on code as in Codex (Chen, Tworek, et al., 2021) which powers GitHub Copilot, as well as through instruction fine-tuning that aligns the model's responses given instructions with human expectations using reinforcement learning, known as InstructGPT (Ouyang et al., 2022). When further combined with fine-tuning on dialogues in a similar way, the resulting model, ChatGPT, has gained great popularity since its release in late 2022, displaying highly human-like language understanding and generation capabilities (Kung et al., 2023; Bang et al., 2023; Qin, Zhang, et al., 2023), and has become the core of a number of AI-powered applications. Combined with the ability to utilize tools with external APIs as in Toolformer (Schick et al., 2023) as well as conducting web search as in WebGPT (Nakano et al., 2022) and New Bing, a competent and versatile AI assistant has taken shape. Therefore, a question arises: is the model capable of conducting spoken language understanding (SLU) tasks like those in current voice assistants?

Current SLU approaches are substantially different from how we use those GPT3-based models. Traditionally, SLU is carried out using a cascaded pipeline, which includes an automatic speech recognition (ASR) module taking audio as inputs, and a natural language understanding (NLU) module working on ASR transcripts, hypotheses, or lattice to predict labels for tasks like intent classification (IC) and slot filling (SF) (De Mori, 2007; Qin, Xie, et al., 2021). Recently, end-to-end approaches that directly predict labels from speech (Serdyuk et al., 2018; Haghani et al., 2018; Saxon et al., 2021) become more popular, and pretrained language and speech models are also introduced into SLU (Wang, Boumadane, and Heba, 2022; Arora, Dalmia, Denisov, et al., 2022; Seo, Kwak, and Lee, 2022). Additionally, there are works focused on low-resource or few-shot textual IC/SF (Yazdani and Henderson, 2015; Ferreira, Jabaian, and Lefèvre, 2015; Hou, Che, et al., 2020; Wu, Su, and Juang, 2021; Peng, Zhu, et al., 2021).

However, those methods are based on the paradigm of supervised training or fine-tuning with a set of possibly large-scale training data. In contrast, considering the difficulty of fine-tuning the whole GPT3 model, recent NLP research highlights a different scheme, namely **prompting** (Liu, Yuan, et al., 2023): given a fixed textual description of the task known as a *prompt* without any training data, the language model may correctly carry out the task. Furthermore, the **in-context learning** approach adds a few paired examples in the textual prompt to further direct the model towards the desired outputs. Such methods are different from traditional zero or few-shot learning in which the model parameters are fixed. It appears to be clumsy and may perform worse than fine-tuned smaller models like T5-11B at the beginning (Lester, Al-Rfou, and Constant, 2021). Additionally, larger models were believed to be unscalable to reach the desired performance given the costs (Brown, Mann, et al., 2020; Kaplan et al., 2020; Cobbe et al., 2021), However, recent explorations reveal the **emergent abilities** of larger models like GPT3-175B and PaLM 540B (Chowdhery et al., 2023): prompting shows low or

even close-to-random performance on multiple tasks until a certain scale of the model where a breakthrough emerges (Srivastava et al., 2023; Wei, Tay, et al., 2022). This breakthrough enables chain-of-thought prompting to surpass the smaller models fine-tuned on rich data (Wei, Wang, et al., 2022; Fu et al., 2023), allows reasoning using internal knowledge with results comparable to external knowledge retrievers (Yu et al., 2023), and leads to better robustness and generalization (Fu et al., 2023; Si et al., 2023).

There have been several works on SLU employing prompts, such as fine-tuning pretrained models like T5 aided by prompts (Wu, Wang, Zhang, et al., 2022; Song, Huang, and Wang, 2024), fine-tuning embeddings prepended to the inputs known as continuous prompts (Chang, Tseng, et al., 2022; Chang, Wang, et al., 2023), and end-to-end SLU by in-context learning on GPT2 with a fine-tuned audio encoder (Gao, Ni, et al., 2022). They are nevertheless distinct from the current prompting and in-context learning scheme, and have not approached the regime of emergent abilities. Hence the potential and limitations of this new type of method on SLU remain unexplored. Therefore we endeavor to undertake it by designing prompts and evaluating these models on multiple SLU benchmarks, including SLURP (Bastianelli et al., 2020) and the multilingual MINDS-14 (Gerz et al., 2021). Since these models take textual inputs, beside oracle transcripts, we also use ASR transcripts from Whisper (Radford, Kim, et al., 2023), which embodies a pipeline upon completely off-the-shelf pretrained models. Furthermore, we compare smaller models that can easily run on a common GPU, namely GPT2 (Radford, Wu, et al., 2019) and several OPT models (Zhang, Roller, et al., 2022).

As a result, we discover that the largest GPT3.5 and ChatGPT models achieve high performance on intent classification under zero-shot or few-shot in-context scenarios that are close or even better than models fine-tuned on the whole dataset, when given the oracle transcripts. This is unique to those large models as the smaller GPT2, OPT and GPT3.5 Curie models have much lower performance and are entirely ineffective under zero-shot cases. Even in the cases where the predictions differ from the labels, the predictions are mostly reasonable, often due to the ambiguity of the sentence. This raises the question that the tasks to predict intermediate IC/SF labels might not sufficiently reveal the potential of the model. However, for the slot filling task with a more complicated task definition, the performance is much worse. Additionally, the accuracy drops significantly when using ASR transcripts. We show that the models have limited awareness of word pronunciations and possible ASR errors, which poses challenges for directly deploying those models for real-world SLU. To facilitate reproduction, relevant resources and prompts are available at https://mutiann.github.io/papers/ChatGPT_SLU/.

4.2 Methods

We design prompts to be given to ChatGPT for the intent classification and slot filling tasks as in Figure 4.1. We begin by explaining the background of the task, and then provide options, like possible intents. We rewrite the names of the scenarios and actions in natural language, for example, with the underlines in the name removed. For in-context learning cases, several

examples are further appended. Following by instructions about the answer format are the questions. Several questions are asked in a batch for better efficiency. For instance, in the case of ChatGPT on SLURP, we ask 45 questions in a conversation, with the first 5 questions appended to the instructions, and the remaining 40 questions split into 2 rounds to be asked. We then collect results from each line of the answer that corresponds with each question by text matching. If the answer could not be identified, we retry up to 3 times. As for ChatGPT, we use the *legacy* (a.k.a. *default*, codenamed *text-davinci-002-render-paid*) model available on the webpage, which gives slightly different results compared to the faster *default* (a.k.a. *turbo*, codenamed *text-davinci-002-render-sha*) model. For slot filling, we further explain the meaning of each entity type. While as for GPT3.5, we adopt the largest *text-davinci-003* model as well as the smaller *text-curie-001* model[1], using a slightly different prompt to fit the task of text completion. Since GPT3.5 is called by the OpenAI API, more customization is allowed, including logit bias to ensure that only tokens that may appear in the answer could be generated. While as for smaller models, we use GPT2 large (774M) (Radford, Wu, et al., 2019), the widely-used predecessor, as well as OPT, an open-sourced reproduction of GPT3 with various sizes available (Zhang, Roller, et al., 2022). We picked the 1.3B, 2.7B, and 6.7B versions that could be easily run on a single GPU, though we have to use half-precision inference on the 6.7B one. We performed generation with an 8-beam search, using a prompt similar to GPT3.5 but perform text completion with one question at a time. Whisper-large is chosen for performing ASR (Radford, Kim, et al., 2023).

The models are evaluated on the test split of two different datasets: SLURP is a large-scale IC and SF dataset of commands to voice assistants with over 141k samples annotated with 60 different intents formulated as scenario-action pairs, as well as 56 types of entities or slots. There have been limited evaluations on a task with so many different types, as current works are mostly focused on tasks with a small label space (e.g. positive/neutral/negative in sentiment analysis), though lower performance has been found on named entity recognition (NER) (Qin, Zhang, et al., 2023) and dialogue state tracking (Bang et al., 2023) with a smaller label space but share some similarity with this task. MINDS-14 is a smaller banking scenario IC dataset of 14 intents in 14 different languages, for which we use the XTREME-S split (Conneau, Bapna, et al., 2022). Evaluations are performed on the test split, while in-context training examples are picked from the training or validation split.

User: We will show you some commands by a user to a voice assistant like Siri or Olly. Please determine that the command is under which of the following predefined scenarios [...]
Audio: volume up, volume down, volume mute, volume other

Some examples are:
1. "book a taxi to the airport for tomorrow morning": the scenario and intent is "Transport: taxi". [...]

Please give answers like: 1. Email: query contact, 2. IoT: cleaning, etc. The intent in your answer must match one of the intents for the corresponding scenario given above. If you are uncertain, choose the one that you think is the most likely.
How about the following commands:
1. event reminder mona tuesday
2. what is the exchange rate of us dollar to pound sterling [...]
ChatGPT:
1. Calendar: set
2. Question Answering: currency [...]

Figure 4.1: An example of ChatGPT doing SLURP intent classification in a conversation. The list of options, questions, and answers are partially omitted for brevity.

[1]The sizes of Davinci and Curie are estimated to be comparable to GPT3 175B and 6.7B respectively, according to Eleuther AI as in https://blog.eleuther.ai/gpt3-model-sizes/

Table 4.1: Accuracy for various GPT3-based models on SLURP intent classification task with zero or few examples.

#Examples	0	10	20	30
GPT3.5	72.86%	74.55%	77.27%	77.44%
w/ bias	75.86%	75.59%	78.31%	77.87%
Curie w/ bias	5.01%	4.91%	3.80%	3.77%
ChatGPT	79.25%	80.33%	**83.93%**	80.16%
Turbo ver.	78.98%	80.03%	81.78%	79.62%

Table 4.2: Accuracy for smaller language models on SLURP intent classification task with zero or few examples. A 30-example prompt is too long for GPT2 with a 1024 token window.

#Examples	0	10	20	30
GPT2 (774M)	6.66%	8.88%	8.31%	-
OPT-1.3B	5.58%	10.69%	17.85%	17.01%
OPT-2.7B	7.06%	28.65%	26.66%	36.97%
OPT-6.7B	4.37%	28.18%	35.14%	**42.40%**

4.3 Experiments

4.3.1 Oracle Texts

We first evaluate the models from OpenAI using oracle transcripts in SLURP for intent classification. As shown in Table 4.1, ChatGPT achieves an accuracy of 79.25% with zero shot, and 83.93% with only 20 examples. Despite there being 60 different types of intents in SLURP, ChatGPT achieves the result even without examples in many of the categories. The result is on par with, or not far from, many supervised NLU models trained on the full oracle transcripts dataset, like SF-ID with 82.25%, HerMiT with 84.84% (Bastianelli et al., 2020), and conformer deliberation with 89.0% (Arora, Dalmia, Chang, et al., 2022), or end-to-end methods like fine-tuning HuBERT with 89.38% (Wang, Boumadane, and Heba, 2022). While more training examples seem to have limited or no improvements, as shown by the results on 30 examples. This could be possibly due to the bloated length of the prompt. When we use a more verbose version of the prompt with 20 examples but with 1260 tokens, similar to the 30-example one, the accuracy drops to 81.41%. ChatGPT gives slightly better performance compared to GPT3.5 even with token bias, possibly thanks to its additional dialogue fine-tuning, while the faster Turbo version shows slightly inferior results.

The results obtained with smaller models are presented in Table 4.2 in addition to the Curie results above. It is evident that the performance falls far behind the presumed 175B GPT3.5 and ChatGPT models. The model size is apparently the most critical factor affecting the performance, while the number of examples also makes a difference. Relatively larger (e.g. 6.7B) models can better leverage the increased examples, while none of the zero-shot experiments work. This is in stark contrast to the largest models where the number of examples has limited

Table 4.3: Accuracy for intent classification on different languages from MINDS-14 with ChatGPT, compared to the supervised textual model using full data.

	en-US	fr-FR	pl-PL	ko-KR
0-shot	95.4%	97.4%	90.0%	89.2%
1-shot	97.9%	99.3%	96.1%	90.5%
LaBSE (Gerz et al., 2021)	95.1%	93.1%	89.2%	91.4%

Table 4.4: F1 score for slot filling with ChatGPT with different number of examples.

#Examples	10	20	30
ChatGPT	12.03%	13.00%	13.35%

impact and the zero-shot cases also show good results. This suggests that the largest models possess sufficient capabilities through pretraining to comprehend the task explanation and the input question without additional training examples. The role of the training examples is more to guide the model to align with the requirements of the task that are not elaborated in the task description. In contrast, in-context examples are more important for smaller models that lack sufficient internal knowledge.

We then evaluate the models on four different languages from MINDS-14 selected based on their proportion in the GPT3 training data: Among 118 languages in the data, 92.6% are in English by word count, while the 2nd-ranked French accounts for 1.8%. Other languages are exceptionally sparse: Polish (ranked 10th) for 0.16%, and Korean (28th) for 0.017%. Nevertheless, ChatGPT generalizes to all of these languages with similar or better performance compared to the supervised LaBSE (Feng, Yang, et al., 2022) reported in (Gerz et al., 2021), though using a different split. The model can almost perfectly solve the task on English and the rather low-resource French with zero shot, as well as Polish using only 14 examples or 1 shot per category to align with the task. Even on Korean with extremely sparse training data, the zero-shot and one-shot results are still satisfactory. This demonstrates that the model enjoys inherent multilinguality to generalize its strong language understanding capability to various languages.

However, when it comes to the slot filling task with a more complicated task definition, the situation is different. As in Table 4.4, the F1 score is poor and much lower than the models like HerMiT with 78.19% F1 (Bastianelli et al., 2020). Therefore, the approach to better leveraging ChatGPT on this task remains to be explored.

4.3.2 ASR Transcription

We then evaluate the model on SLURP intent classification using ASR transcripts, which reflects the real SLU scenario. The SLURP recordings are transcribed with 16.7% test WER with the off-the-shelf Whisper, which is acceptable, considering that the audio is often noisy and

Table 4.5: Accuracy for intent classification on SLURP with different number of examples using ASR transcripts.

#Examples	10	20	30
GPT3.5	64.78%	68.95%	68.64%
ChatGPT	72.89%	73.96%	72.85%

an XLSR-based ASR system adapted to the SLURP data reports 15.5% test WER (Villatoro-Tello et al., 2023). To our disappointment, the models suffer from significant performance drop processing ASR transcripts as in Table 4.5, even though the model is prompted to account for ASR errors.

Considering that GPT3 is capable of correcting language errors, we are inspired by the chain-of-thought prompts (Wei, Wang, et al., 2022) to first instruct the model to try to correct the ASR errors, and then determine the scenario and action. Such a prompt with 5 examples brings 67.15% accuracy on GPT3.5, which alleviates the issue with a 2.4% improvement compared to the 10-example prompt, although it is still far from the oracle transcript case and leads to bloated prompt. We also attempt to directly instruct GPT3.5 to correct ASR errors, characterized by the replacement of words with similar pronunciation. Given 1000 error cases, WER increases from 25.6% to 32.4% after the correction. In most cases the sentences are more fluent, but ASR errors remain. Hence we hypothesize that due to the nature of training on textual inputs with wordpiece tokenization, the model has limited awareness of phonetics, as implied by their high error rates on the IPA task (Srivastava et al., 2023). We further verify that by asking ChatGPT the names of 50 well-known places with their Romanized transliteration in Chinese and Japanese, e.g., "A city is called Nyūyōku in Japanese. What's its name in English?", which can be easily guessed by a person knowing the common pronunciation of Latin letters. However, ChatGPT can correctly answer only 46% from Japanese and 10% from Chinese [2], which further confirms its deficiency in phonetic knowledge.

4.4 Discussions

Experiments have demonstrated the capability of the model, though many errors are found. However, it is worthwhile to look into the error cases to determine if ChatGPT really failed to understand those text, especially considering that SLURP labels are relatively noisy (Villatoro-Tello et al., 2023). Therefore, we check 100 error cases in the 20-shot ChatGPT experiments, and find that 79% of them are not errors in the strict sense, as exemplified in Table 4.6. Some of the "errors" occur when the input sentence is ambiguous and open to multiple different interpretations. An example is shown in Figure 4.2, in which case we find that ChatGPT could actually provide the correct answer and explain the answer if we provide clarification that is

[2] Later we experimented with the more recent GPT4 model, and it achieves 100% accuracy on this Chinese test. It also appears to have stronger capability for correcting ASR mistakes, but errors are still often. Therefore a stronger textual model may partially alleviate this limitation, though the extent remains unclear and is left for future investigation.

Table 4.6: Examples of "errors" made by ChatGPT on SLURP intent classification and slot filling. Many of the predictions are also reasonable, but different from the labels in the dataset.

Command	Label	ChatGPT Prediction
Is there a groomer in town for cats only?	General: quirky	Recommendation: locations
In how many hours will it be midnight in London England.	Date/time: query	Date/time: convert
Please tell me news related to the stock market.	Question answering: stock	News: query
Olly I need a drink.	IoT: coffee	General: quirky
Can you give me the details of upcoming annual function on twenty sixth march?	General: quirky	Recommendation: events
How long does it take to make vegetable lasagna?	Cooking: recipe	Cooking: query
Give details of rock sand.	Question answering: definition	Question answering: factoid
Event reminder Mona Tuesday.	Event name: Mona; Date: Tuesday	Alarm type: reminder; Person: Mona; Date: Tuesday
Exchange rate of US dollar to pound sterling.	News topic: exchange rate of US dollar to pound sterling	Definition word: exchange rate; Currency name: US dollar; Currency name: pound sterling
Take out the milk from the shopping list.	List name: shopping	List name: shopping; Ingredient: milk
Increase the brightness of the lights.	-	Change amount: brightness; Device type: lights
Please send a mail to my friend Divya how are you.	Relation: friend; Person: Divya	Email address: Divya; Person: friend
Please scan my social media and tell me what's happening.	-	Media type: social media; Definition word: what's happening

not given in the command. There are also some examples for which ChatGPT gives a more accurate answer than the label. While in many other cases the issues arise due to the ambiguity of the categories, which are often specified by the annotation scheme of the SLURP dataset. For example, asking for news about stock is labeled as "Question answering: stock," not "News: query." This is a fundamental weakness of prompting: the instructions, examples, and label names in the prompt must fully reveal the goal of the task if it is not typically observed in the textual corpus, which is difficult when the task or labeling specification is complicated and subtle, even if the task itself is straightforward. Such annotation specifications could be captured by a supervised model given the whole training set, but impossible for a zero-shot or in-context model.

The phenomenon is more pronounced in the slot filling case where multiple entities with 56 different types could be extracted from a single command. The meanings of the labels often overlap with each other, and the annotations tailored to the specific scenario of the dataset.

For example, normal apps are extracted as “app name,” and companies as “business name”, but Uber as “transport agency” and Northern Rail as “transport name,” which would baffle a zero/few-shot model as in Table 4.6. The harm of such counter-intuitive labelling scheme also matches the observations in (McKenzie et al., 2022). This kind of misalignment has more impact in such a tagging task as tagging decisions need to be made on every individual word, which may explain the observed worse performance on similar tasks like NER and dialogue state tracking (Bang et al., 2023; Qin, Zhang, et al., 2023). We also find that ChatGPT may be biased towards patterns in the limited examples. For example, with “tell me about morel mushrooms” labeled as “Definition word: morel mushroom” in a training example, ChatGPT tends to label “Definition word” on other sentences like “tell me what’s happening.” All these examples illustrate the difficulties of in-context learning on a task with a complicated task goal.

User: great song for commute
ChatGPT: The scenario and intent is “Recommendation: music”
User: The user might be asking the assistant to recommend a song for the commute unknown to the user. But what if we already know what song the user is talking about, as the device is playing it at the moment?
ChatGPT: If the device is already playing a song and the user says “great song for the commute”, then the intent could be “Music: likeness” as the user is expressing their enjoyment of the current song and suggesting that it would be a good choice for a future commute.

Figure 4.2: An example for ChatGPT giving an answer different from the label (“Music: likeness”) on an ambiguous command. While ChatGPT is able to determine the answer that matches the label under an alternative situation.

In the evaluations conducted above, we direct the model to fulfill requirements of the dataset, which are intermediate tasks in a pipeline. The predictions are then utilized to run external functions (e.g., sending an email, answering a question, telling a joke) available in the specific annotated scenario using certain rules. However, ChatGPT has demonstrated its versatility as a chatbot that can communicate with users without being bound by these objectives. For instance, the model often appears to “understand” the commands in the “error” cases above, and may directly request for clarification when the input can not be understood, such as when an ASR error happens. When equipped with the ability to call external APIs (Schick et al., 2023; Nakano et al., 2022), New Bing can read returned results and reason upon them to determine the final response to users’ requests. Additional instructions regarding the functions can be also fed to the model via some system prompt in multiple rounds of conversation along with user inputs. In this way, the model’s ability on such intermediate tasks may underestimate its actual capability as an end-to-end assistant.

4.5 Conclusions

In this chapter we evaluate large language models on SLU tasks with prompting. We find that they reach results comparable to fine-tuning on intent classification of multiple languages, but not on tasks like slot filling when understanding the task itself is challenging, and being sensitive to ASR errors. This supports the potential of using LLMs for SLU, but also reveals the limitations of cascaded pipelines, and highlights the need for end-to-end approaches.

5 Linearizing Pretrained Language Models

Previous chapters have implied pretrained generalist model as an essential component for our goal on complicated semantic understanding under low-resource scenarios. However, a critical challenge under the context of this thesis is that the quadratic computational cost of such models can be prohibitive when processing long contexts. Architectures such as Linformer and Mamba have recently emerged as competitive linear time replacements for transformers. However, corresponding large pretrained models are often unavailable, especially in non-text domains. In this chapter, we present a Cross-Architecture Layerwise Distillation (CALD) approach that jointly converts a transformer model to a linear time substitute and fine-tunes it to a target task. We also compare several means to guide the fine-tuning to optimally retain the desired inference capability from the original model. The methods differ in their use of the target model and the trajectory of the parameters. In a series of empirical studies on language processing, language modeling, and speech processing, we show that CALD can effectively recover the result of the original model, and that the guiding strategy contributes to the result. Some reasons for the variation are suggested.

The work in this chapter has been published as:

Mutian He and Philip N. Garner (Apr. 2025). "Joint Fine-tuning and Conversion of Pretrained Speech and Language Models towards Linear Complexity". en. In: *The 13th International Conference on Learning Representations, ICLR 2025*

5.1 Introduction

Recent progress in language modeling from BERT (Devlin et al., 2019) to Llama3 (Grattafiori et al., 2024) has witnessed the rise of the attention mechanism and transformer architecture. Such models are able to scale up to very large model capacity and pretraining data for natural language processing (NLP), and have lead to breakthroughs in a broad range of downstream applications under either fine-tuning or zero-shot/in-context scenarios. This is surely not restricted to the text domain: Off-the-shelf speech and audio models like Wav2Vec2 (Baevski et al., 2020), HuBERT (Hsu, Bolte, et al., 2021), and WavLM (Chen, Wang, Chen, et al., 2022) are pretrained on extensive audio data; they can be fine-tuned to obtain good performance on various tasks from automatic speech recognition (ASR) to spoken language understanding (SLU). However, such large-scale models are computationally expensive; the main reason is that in the standard attention mechanism, each token interacts with all other tokens in the input sequence, forming a complete graph. The computation time therefore increases quadratically with the input length, and a linear-growing KV-cache is required. As a result, the computation requirements can become prohibitive, especially for long contexts. A large cohort of architectures and algorithms have been proposed to mitigate the problem and to reach linear complexity, using techniques such as low-rank projection, hashing, and sparse attention (Tay et al., 2022). More recently, there has been a revival of recurrent networks such as state space models (Gu, Goel, and Re, 2022; Gu and Dao, 2024), RWKV (Peng, Alcaide, et al., 2023), and RetNet (Sun, Dong, et al., 2023). By recurrently updating states, those models allow linear time and constant space inference while retaining the transformer's parallel training capability. It has been demonstrated that the latest recurrent models, notably Mamba2 (Dao and Gu, 2024), are capable of reaching performance comparative with the most well-trained transformer, even in large-scale models.

In spite of such transformer substitutes, there remain obstacles to putting them into practical use. These new models are not simply plug-and-play upgrades for existing pretrained models, but are generally pretrained from scratch again. The pretrained weights of these models are often not publicly released; the data and computational cost of repeating such pretraining for every new model can be prohibitive for users and researchers of downstream applications. In particular, most models are primarily developed for processing language or text. These models are typically general-purpose substitutes for the standard attention mechanism, also applicable to other domains such as speech and audio. However, the pretrained models on those latter domains simply never exist; this hinders those domains from utilizing the latest models and algorithms created in the larger machine learning (ML) and NLP community. This is a particular concern for research in end-to-end speech processing involving long-distance dependency in the lower information density audio form: such models are a prerequisite to efficiently process the abundant long-form but poorly segmented audio in the wild.

To address this issue, a possible solution is to make use of off-the-shelf pretrained models and convert them into the target linear-complexity model. Several previous works have explored this method to obtain a target model with much less computation. Many of the standard

transformer layers (e.g., embeddings, MLP) are still part of the target architecture. Compared to pretraining from scratch, an immediate improvement would be the reuse of those pretrained layers (Kasai et al., 2021; Choi, 2024). In addition to such parameter transfer, we can further guide the conversion by transferring model behavior using knowledge distillation (KD). Given model inputs, KD aligns the model outputs or hidden states between the original transformer (teacher) and the target model (student) using an extra loss term. In an earlier work, Tang, Lu, et al. (2019) explored conversion from BERT to RNN by aligning outputs; more recently, Zhang, Bhatia, et al. (2024) and Bick et al. (2024) tried to reproduce the attention weights of the standard attention mechanism in linear attention models. However, it has been pointed out that reproducing attention weights is not a prerequisite to reproduce performance (Mao, 2022), and even causes training inefficiency and instability (Mercat et al., 2024). More importantly, this method is restricted to target models for which we can easily determine an attention weight matrix equivalent to that of the standard one. This is not always the case: e.g., the matrix in Linformer has a lower rank (Wang, Li, Khabsa, et al., 2020). Furthermore, distillation only on outputs may not be sufficient to guide the model. The above works are restricted to NLP tasks without considering other more demanding domains, e.g., speech. Furthermore, most of these works are focused on conversion using an efficient re-pretraining stage, which is still difficult for downstream users with limited access to computational resources and pretraining data. Instead, it would be more attractive to directly perform the conversion during fine-tuning.

Given the considerations above, we explore the possibility of converting an existing pretrained transformer into a linear-complexity model on the downstream task. Our approach broadly covers different source models, target models, and target tasks. This is made possible by combining parameter transfer and layerwise distillation with hidden states from a target teacher model, i.e., a pretrained standard transformer fine-tuned for the target task. In addition to direct guidance from hidden states of the target teacher, we further investigate two modes of guidance: Recognizing the importance of the pretrained knowledge of the original transformer that can be lost during fine-tuning, we can guide the converted model by the sequence of models on the teacher's fine-tuning trajectory. Alternatively, we can use the distilled model as a good initialization for conversion, and allow deviation from the teacher's behavior in the later training phase.

We demonstrate the advantage of our novel Cross-Architecture Layerwise Distillation (CALD) approach under a broad range of scenarios, including conversion from RoBERTa to Linformer for NLP tasks, from Pythia to Mamba for language modeling, and from Wav2Vec2 to Mamba2 for speech tasks including ASR, SLU, and speaker identification (SID). In all of these scenarios, our converted models demonstrate minor to no performance loss compared to the standard transformer, much better than naive parameter transfer. Code and resources to reproduce our work are available at https://github.com/idiap/linearize-distill-pretrained-transformers.

5.2 Background and Related Work

Linear-Complexity Models Various linear-complexity models have been introduced in Chapter 2.3, including Mamba and Mamba2 used in this chapter. In addition, this chapter also leverages a different type of effcient attention: Linformer Wang, Li, Khabsa, et al. (2020) finds that the attention matrix $\boldsymbol{A}$ is inherently low-rank, and projects $\boldsymbol{K}$ and $\boldsymbol{V}$ into a lower k-dimensional space using $k \times N$ matrices $\boldsymbol{E}$ and $\boldsymbol{F}$. The two matrices can be shared, or even shared between all layers. Then, the $O(N^2)$ matrix multiplication can be avoided, as $\boldsymbol{A}$ can be provably approximated by

$$\tilde{A}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}(\boldsymbol{E}\boldsymbol{K})^T}{\sqrt{d}})\boldsymbol{F}\boldsymbol{V}. \tag{5.1}$$

Knowledge Distillation Distilling knowledge from teacher models to a student one by matching the output probabilities is a standard approach to compress ML models, including pretrained language models (LMs) (Hinton, Vinyals, and Dean, 2015; Sanh et al., 2020). Intermediate hidden states of the teacher and student can be further aligned by layerwise distillation (Sun, Cheng, et al., 2019; Aguilar et al., 2020). In addition, Brown, Williamson, et al. (2023) have attempted to distill and compress existing pretrained linear-complexity LMs. In these works, parameters from the teacher are also transferred to the student when possible, hence also represent the direct parameter transfer approach.

Model Conversion Distinct from model compression, there is also a series of works aimed at converting (a.k.a. uptraining) existing models into new, more efficient ones. Parameter transfer is generally applied, e.g., in the conversion from standard attention to sparse attention (Zaheer et al., 2020), RNN (Kasai et al., 2021), linear attention (Mao, 2022; Mercat et al., 2024), and multi/grouped-query attention (Ainslie, Lee-Thorp, et al., 2023). Distillation, or behavior transfer, from the standard transformer as the teacher has been also introduced, starting from early attempts to convert transformers to RNN using teacher outputs or generations (Senellart et al., 2018; Tang, Lu, et al., 2019). Zhang, Bhatia, et al. (2024) further tried to reproduce the attention matrix of standard attention in linear attention with a loss to align the teacher and student attention weights. Bhati, Gong, et al. (2024a) also used output distillation for pretraining Mamba-based speech models, but without parameter transfer. A concurrent work then combined the distillation on attention weights, outputs, and hidden states (Bick et al., 2024), and managed to convert large-scale LMs into Mamba with limited computation costs and performance loss. These works are nevertheless focused on conversion during pretraining; by contrast, we focus on conversion during fine-tuning, which is generally cheaper and more accessible. Choi (2024) investigated conversion to RetNet or Mamba during fine-tuning on commonsense reasoning tasks, but using only parameter transfer. Zhang, Bhatia, et al. (2024) also considered conversion after fine-tuning by attention weight alignment. A concurrent work converts Llama3 into Mamba using distillation on outputs and generations during instruction fine-tuning (Wang, Paliotta, et al., 2024). Meanwhile, LoSparse jointly fine-tunes

and compresses linear layers into LoRA + sparse matrix (Li, Yu, et al., 2023). Our work is nevertheless distinct in that we investigate different modes of distillation under a general framework of joint fine-tuning and conversion into efficient architectures based on layerwise distillation, which allows a broader range of models and tasks as discussed above.

5.3 Methods

We aim at building a broadly applicable framework to convert an existing pretrained model into a task-specific linear-complexity one. To achieve this goal, we combine parameter transfer and distillation to best reproduce the performance of the **target teacher model**, a standard transformer model fine-tuned on the target task from the pretrained **source teacher model**. For parameter transfer, we replace only the attention layers in the teacher with the more efficient sequence-mixing modules we intend to use, forming the **student model** as a result; most other layers including embeddings, feed-forward, and layernorm will be preserved. Following this approach, the teacher parameters can be used to initialize the student model as much as possible. This constitutes our baseline **unguided** approach, in which we directly fine-tune the student model with transferred parameters. Instead, by posing loss on the difference of hidden states, we enforce constraints on the deviation of the student's hidden states from the teacher. In this way, we directly guide each layer in the student to reproduce the behavior of the corresponding teacher layer. In addition, we consider different ways or modes of such guidance, introduced below. These modes are also illustrated in Figure 5.1, and pseudo-codes for the different modes of training algorithms are available in Appendix B.2.

- **Target Guided**. We can directly transfer the parameters from the fine-tuned teacher target model and distill from it. More specifically, given the model inputs for training on a target classification task, we denote the hidden states from the student and the teacher target as $\boldsymbol{H}^{(s)}$ and $\boldsymbol{H}^{(t)}$, each with m vectors; outputs (class probabilities) of the student and the teacher target as $\boldsymbol{y}^{(s)}$ and $\boldsymbol{y}^{(t)}$; and one-hot labels as $\boldsymbol{y}$. Then the loss terms are written as

$$\mathscr{L}_{\mathrm{CE}}(\boldsymbol{y}^{(s)}, \boldsymbol{y}) = -\sum_i \boldsymbol{y}_i \log(\boldsymbol{y}_i^{(s)}) \tag{5.2}$$

$$\mathscr{L}_{\mathrm{KD}}(\boldsymbol{y}^{(s)}, \boldsymbol{y}^{(t)}) = \sum_i \left(\frac{\boldsymbol{y}_i^{(t)}}{\beta}\right) \log\left(\frac{\boldsymbol{y}_i^{(t)}/\beta}{\boldsymbol{y}_i^{(s)}/\beta}\right) \tag{5.3}$$

$$\mathscr{L}_{\mathrm{LD}}(\boldsymbol{H}^{(s)}, \boldsymbol{H}^{(t)}) = \frac{1}{m}\sum_{i=1}^{m} \left(\boldsymbol{H}_i^{(s)} - \boldsymbol{H}_i^{(t)}\right)^2 \tag{5.4}$$

$$\mathscr{L} = \alpha_{\mathrm{CE}}\mathscr{L}_{\mathrm{CE}} + \alpha_{\mathrm{KD}}\mathscr{L}_{\mathrm{KD}} + \alpha_{\mathrm{LD}}\mathscr{L}_{\mathrm{LD}} \tag{5.5}$$

  We set the temperature $\beta = 2$ following Hinton, Vinyals, and Dean (2015). The final loss $\mathscr{L}$ is a combination of the cross entropy $\mathscr{L}_{\mathrm{CE}}$, the KL divergence $\mathscr{L}_{\mathrm{KD}}$ between teacher and student outputs in standard KD, and an MSE loss $\mathscr{L}_{\mathrm{LD}}$ between hidden states for layerwise distillation. The three terms are weighted by a set of hyperparameters α.

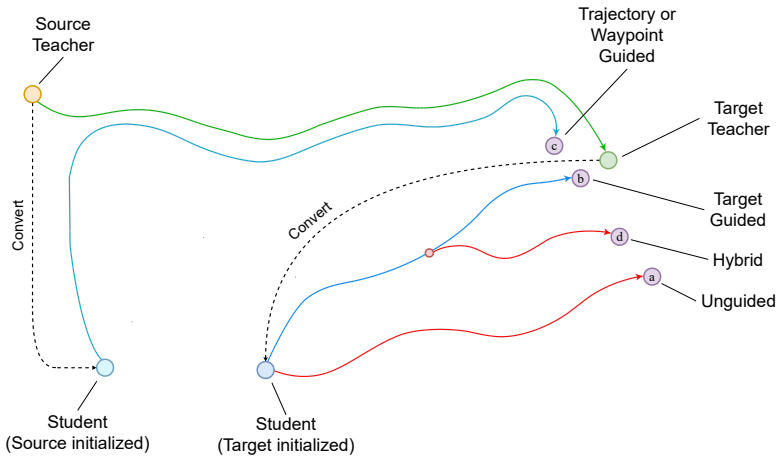


Figure 5.1: A conceptual illustration of the hidden states shift during training under different modes of distillation. Given the trajectory (green line) of the hidden states during the fine-tuning from the source teacher model (i.e. pretrained transformer) to the target teacher model (i.e. transformer fine-tuned on the target task), we consider: a) Unguided: parameter transfer without any distillation; b) Target Guided: distill from the target teacher; c) Trajectory/Waypoint Guided: gradually distill from a series of models on the trajectory; d) Hybrid: distill from the target teacher until a certain step.

Instead, the unguided mode uses $\mathscr{L}_{\text{CE}}$ only. More specifically, as the target sequence-mixing module (e.g., RNN) is nevertheless distinct from standard attention, it can be more difficult to reproduce the hidden states right after the sequence-mixing layer. Instead, we extract the hidden states after the processing of the following feed-forward layer as $\boldsymbol{H}$ for layerwise distillation. As for tasks other than classification, we will replace the $\mathscr{L}_{\text{CE}}$ term with the corresponding loss function, e.g. CTC loss for ASR tasks.

- **Trajectory guided**. Models initialized from pretrained parameters perform much better than random initialization when trained on downstream tasks. This can be explained by the rich knowledge internalized into the pretrained model, which may be lost during fine-tuning. It has been found that preserving the knowledge during fine-tuning can be helpful for downstream performance (Li, Grandvalet, and Davoine, 2018; He and Garner, 2023a). Therefore, it can be sub-optimal to only reproduce the hidden states produced by the fine-tuned teacher target model, which has already lost certain knowledge. We are further inspired by Li, Xiong, et al. (2019) who preserve certain hidden states from shifting during fine-tuning for better downstream performance, as well as classical homotopy or continuation methods that solve an optimization problem by tracing a series of surrogate problems that gradually lead to the goal (Allgower and Georg, 1990; Lin, Yang, et al., 2023). Hence we first initialize the student with source teacher parameters, and then simultaneously fine-tune the teacher and the student. In each step, the distillation loss is not computed against the hidden states of the final teacher target model, but the current teacher model instead. In this way, the student model

will reproduce the trajectory of the hidden states during the whole teacher fine-tuning process. As a result, we can better capture the pretraining knowledge present at the early stage of fine-tuning, and adapt to the target task in a way more similar to the teacher. As the converted architecture is different from the original one and may require multiple steps to approximate the current teacher hidden states, we also consider updating the teacher at a slower pace, e.g., every T_u steps.

- **Waypoint guided**. Joint fine-tuning of the teacher and student will essentially double the computation. Instead, we can make use of the checkpoints in teacher fine-tuning. That is to say, we mark the teacher checkpoint of every T_w steps as the guiding *waypoint*. During conversion, we use the nearest waypoint ahead of the current training step as the current teacher. In this way, we approximate the trajectory guided approach in a more coarse-grained way but without extra computation compared to the target guided approach.
- **Hybrid**. The target architecture is after all distinct from the standard transformer, and hidden features in an optimal transformer can be sub-optimal for the target architecture. Therefore, a hybrid scheme between the target guided and unguided modes can be useful. Similar to KD annealing (Jafari et al., 2021), we apply the target guided mode at the initial phase of fine-tuning. We later switch to the unguided mode, which allows the model to train without the distillation constraint. In this way, the initial distillation provides a good initialization for the following optimization.

5.4 Empirical Studies

5.4.1 Model Configuration

We carry out experiments on several scenarios that are representative to demonstrate the advantages of our approach, including

- Converting RoBERTa to Linformer, fine-tuned on language processing tasks, and
- Converting Wav2Vec2 to the latest Mamba2, fine-tuned on speech tasks.

Although not our focus, we also carry out an extra experiment to convert a widely-used open-source LM of various sizes, namely Pythia (Biderman et al., 2023), into Mamba for language modeling by retraining on a small subset of the pretraining corpus.

As mentioned above, we build the target architecture that allows parameter transfer as much as possible. In this way, the architecture may not be identical to the standard Linformer, Mamba, or Mamba2, as specifically described below:

- Language processing by RoBERTa → Linformer: We only add the E and F matrices to project the hidden features, which are randomly initialized by $\mathcal{N}(0,1)$ to preserve the

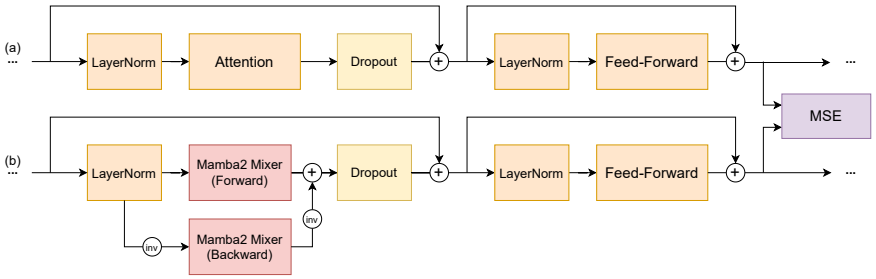


Figure 5.2: Encoder layers in the student bidirectional Mamba2 model (b), converted from Wav2Vec2 encoder layers in the teacher model (a). In each encoder layer, the attention layer is replaced by two new forward and backward Mamba2 mixers. Inputs and outputs of the backward mixer are timewise inverted. Hidden states after the feed-forward layer are extracted for layerwise distillation.

scale of features after projection. All other parts of the model are left unchanged and initialized with pretrained RoBERTa parameters. We follow the reported best Linformer configuration (KV or layer-shared E and F, with rank $k = 256$).

- Language modeling by Pythia → Mamba: We only replace each attention layer with a Mamba mixer of the same hidden size. The mixer is initialized using the specific Mamba state-space scheme to best preserve the past memory (Gu and Dao, 2024).

- Speech processing by Wav2Vec2 → Mamba2: Wav2Vec2 is a bidirectional model, while Mamba2 is unidirectional. Hence we build a bidirectional Mamba2 similar to the one by Zhang, Zhang, et al. (2024). Each attention layer is replaced by two Mamba2 mixers of the same hidden size, but with expand factor = 1, as shown in Figure 5.2. The mixer is similarly initialized with the specific Mamba2 state-space scheme (Dao and Gu, 2024).

In this way, all the models have roughly the same number of parameters after conversion. Classification tasks are carried out by adding a linear prediction head after mean pooling. The modeling and training configuration under each scenario is further introduced below, and more details including hyperparameters are provided in Appendix B.1. Additional details on the computational costs are given in Appendix B.3.

5.4.2 Language Processing

We follow the settings in the Linformer paper to fine-tune the models and report the validation accuracy on a set of widely-used benchmarks: QNLI (Rajpurkar et al., 2016) for natural language inference, QQP (Iyer, Dandekar, and Csernai, 2017) for text similarity, as well as SST2 (Socher et al., 2013) and IMDB (Maas et al., 2011) for sentiment classification; the former three being part of the GLUE benchmark (Wang, Singh, et al., 2018). The models are converted from and compared with pretrained RoBERTa-base. In this way, we are able to directly compare with the reported results of the fully re-pretrained Linformer, which is not publicly available.

Table 5.1: Performance of Linformer models converted from RoBERTa, measured by accuracy on benchmarks reported by the original Linformer paper (Wang, Li, Khabsa, et al., 2020), in comparison with the reported results of a fully re-pretrained Linformer. The best results are bolded.

	QNLI	QQP	SST2	IMDB	Average
Pretrained Linformer	91.2%	90.8%	93.1%	94.1%	92.3%
Std. RoBERTa	92.4%	91.8%	95.3%	95.7%	93.8% +1.3
Unguided	69.4%	84.3%	83.6%	82.6%	80.0% -12.3
CALD					
- Target Guided	89.0%	91.8%	93.3%	92.3%	91.6% -0.7
- Src. init.	88.5%	91.7%	93.1%	92.3%	91.4% -0.9
- Trajectory Guided	**91.2%**	**91.9%**	**94.0%**	**93.1%**	**92.5%** +0.2
- Waypoint Guided	89.9%	**91.9%**	93.7%	92.8%	92.1% -0.2
- Hybrid	86.8%	90.8%	91.4%	90.5%	89.9% -2.4

We fine-tune all models for 10 epochs, and a waypoint is taken after each epoch.

The results are given in Table 5.1. Compared with our fine-tuned standard RoBERTa, the re-pretrained Linformer has a slight drop in performance (1.3% on average). This is typical and can be explained by the reduced expressivity compared to the standard attention. While without extensive pretraining, unguided conversion with only parameter transfer leads to much lower accuracy and severe training instability. Despite much lower computational costs, such performance regression is unacceptable. Fortunately, the regression can be alleviated using CALD. By simply using the target guided approach, the performance drop can be reduced to merely 0.7% on average compared to pretrained Linformer. With the trajectory guided approach, the CALD model can even reach better results (+0.2% on average) but without any pretraining. The approximated waypoint guided approach leads to only slightly lower results, but still better than the target guided one. The hybrid approach leads to lower results compared to target guided, indicating that the guidance is helpful even after early training updates, which matches the rather low unguided results; this is further discussed below in Chapter 5.4.4. In addition, we try to use the target guided approach but initialized from the teacher source parameters. Since the distillation should be easier if the student model is initialized with parameters exactly from the teacher model (i.e. the teacher target), these *Src. init.* models show lower results as we expected. All these results indicate that the behavior transfer by CALD is effective and critical for the joint conversion and fine-tuning in this scenario to minimize the performance regression. Particularly, the trajectory or waypoint guided approaches are helpful.

5.4.3 Language Modeling

We then present the results of experiments on converting Pythia to Mamba for language modeling. This is nevertheless not our focus as recent pretrained LMs are typically directly

Table 5.2: Zero-shot performance of downstream evaluation tasks reported by Pythia (Biderman et al., 2023) on Mamba-based language models converted from Pythia-1B, using a small subset of Pile.

Model	Lambada	PIQA	Winog.	WSC	ArcE	ArcC	SciQ	LogiQA	Avg.
Pythia-1B	0.562	0.707	0.535	0.670	0.570	0.244	0.839	0.221	0.544
0.5% Pile / 1.5B tokens									
Unguided	0.394	0.671	0.493	0.542	0.502	0.211	0.778	0.246	0.479
Tgt. Gd.	0.410	0.686	0.504	0.608	0.538	0.210	0.775	0.230	0.495
Hybrid	0.432	0.683	0.507	0.608	0.537	0.209	0.788	0.238	0.500
1% Pile / 3.0B tokens									
Unguided	0.453	0.673	0.514	0.571	0.525	0.217	0.794	0.217	0.495
Tgt. Gd.	0.449	0.689	0.518	0.626	0.535	0.220	0.798	0.218	0.507
Hybrid	0.479	0.693	0.520	0.648	0.531	0.224	0.808	0.247	0.519
2% Pile / 6.0B tokens									
Unguided	0.475	0.687	0.535	0.612	0.543	0.222	0.802	0.237	0.514
Tgt. Gd.	0.459	0.697	0.527	0.656	0.547	0.218	0.817	0.235	0.520
Hybrid	0.485	0.708	0.510	0.645	0.556	0.224	0.829	0.244	0.525

evaluated by zero or few-shot performance in downstream tasks without fine-tuning. Therefore, we will essentially need to do the re-pretraining under this scenario. Yet we try to convert the model by training only on a tiny (0.5%, 1%, or 2%) subset of the pretraining corpus, i.e. the deduplicated Pile (Gao, Biderman, et al., 2020; Biderman et al., 2023). Under this scenario, we can only compare the unguided, target guided, and hybrid approaches. Models are trained for 2 epochs. Due to limitations in computational resources, we perform experiments only on Pythia-1B.

We evaluate the models using a set of benchmarks based on those reported in the Pythia paper as shown in Table 5.2. Compared with the standard Pythia-1B, there remains a performance gap. Nevertheless, we can find that the gap gets limited using 2% data. More importantly, the CALD models consistently get better results than the unguided ones, especially with the hybrid method. We also observe that the unguided training is rather unstable with frequent NaNs. All the results support the effectiveness of our CALD approach.

5.4.4 Speech Processing

We then convert the Wav2Vec2-large models into the latest Mamba2, using a bidirectional architecture specified above. We evaluate them on three common speech tasks with corresponding common benchmarks: ASR with TED-LIUMv3 (Hernandez et al., 2018), intent classification, a standard SLU task, with SLURP (Bastianelli et al., 2020), along with speaker ID with VoxCeleb1 (Nagrani et al., 2020), following the SUPERB configuration (Yang, Chi, et al.,

Table 5.3: Performance (%) of Mamba2-converted Wav2Vec2 models: word error rate for ASR on TED-LIUM, as well as accuracy for intent classification on SLURP and speaker ID on VoxCeleb1. The best results are bolded.

	ASR WER ↓	IC Acc. ↑	SID Acc. ↑
Std. Wav2Vec2	6.24	91.70	96.09
Unguided	11.29	79.68	84.24
CALD			
- Target Guided	6.56	90.43	**96.56**
- Waypoint Guided	6.92	90.32	96.16
- Hybrid	**6.41**	**91.23**	96.41

2021). Specifically, ASR training is performed by CTC loss instead of cross entropy as L_{CE}. The converted models are compared with standard fine-tuned Wav2Vec2 models. We take a waypoint every 10,000 steps during fine-tuning. Test word error rate (WER) and accuracy are reported respectively.

Results are given in Table 5.3, which are consistent with previous findings. There is a significant gap between the performance of standard Wav2Vec2 and the unguided conversion. This can be largely alleviated by CALD using the target guided approach, reaching results close to or even better than standard Wav2Vec2. The hybrid approach further brings slight improvements on ASR and IC.

However, the waypoint guided approach is found to be not helpful or even harmful; we choose not to proceed with the more precise trajectory guided approach. Our assumption is that the trajectory guided approach helps retain the knowledge in the hidden states of the pretrained model. Hence We hypothesize that the hidden states of the Wav2Vec2 model undergo significant shift during fine-tuning. As a result, the hidden states of the original pretrained model or those in the early phase of fine-tuning are less useful for the target task. To examine the hypothesis, for each model, we compute the average cosine distance of the hidden states produced by the source teacher and the waypoints, using 100 random data samples. As shown in Figure 5.3, as for fine-tuning RoBERTa on NLP tasks, the cosine distance is relatively small and increases gradually. While the distance is much larger during the fine-tuning of Wav2Vec2 on speech tasks, even at the early phase of fine-tuning. In fact, the distance has already reached 0.372 after only 2,000 steps of fine-tuning on TED-LIUM. Therefore, our hypothesis is supported that the trajectory guided approach works better in the cases when there is limited hidden state shift during fine-tuning, and when the pretrained model features remain relevant for the target task. While the condition is not satisfied for speech tasks, in which case allowing the model to deviate more from the initial behavior using the hybrid approach will be beneficial.

In addition, we have also attempted to train the models that are unguided and without any parameter transfer. Such a large Mamba2-based speech model trained on the target task from

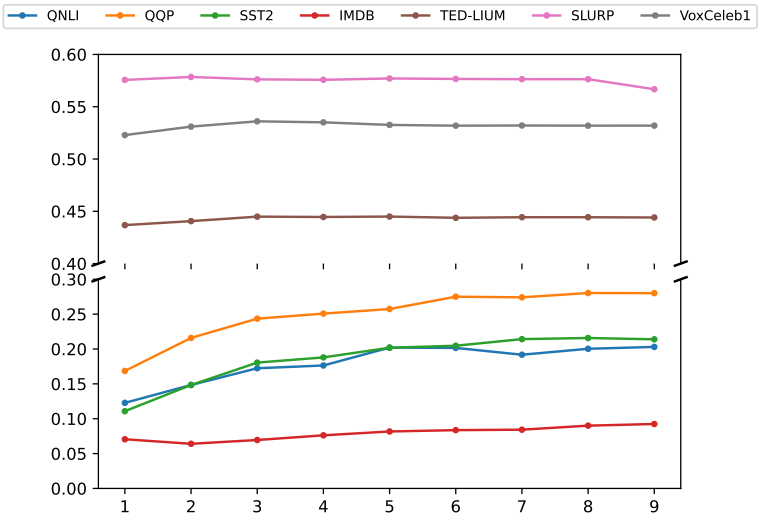


Figure 5.3: Average cosine distance between the hidden states produced by the initial model and the checkpoints during fine-tuning after each epoch (for NLP tasks) or every 10,000 steps (for speech tasks). Unlike NLP models, features produced by the fine-tuned speech models are far from the initial ones since the early phase of fine-tuning.

scratch is rather unstable and fails to converge well. This highlights the necessity of pretraining in such large models.

5.4.5 Hidden State Trajectory

We further visualize the trajectory of hidden states during fine-tuning on the QNLI dataset using different guidance modes by applying t-SNE to concatenated hidden states of 20 random samples in the training set. Hidden states produced by every half-epoch checkpoint are included, hence a total of 20 points during training are plotted for each model. As shown in Figure 5.4, the actual trajectory matches our expectation conceptualized in Figure 5.1 that the target guided model approaches the fine-tuned target teacher model (the end of the green trajectory), while the waypoint or trajectory guided models further mimic the trajectory, which is found to be beneficial to the final results. In sharp contrast, the unguided model will completely deviate from the teacher, which leads to training instability and failure to reproduce the original performance.

5.5 Conclusion

We investigate the task of converting various existing pretrained transformer models into efficient linear-complexity architectures without the need to redo the whole pretraining. For this goal, we propose a novel and broadly compatible Cross Architecture Layerwise Distillation

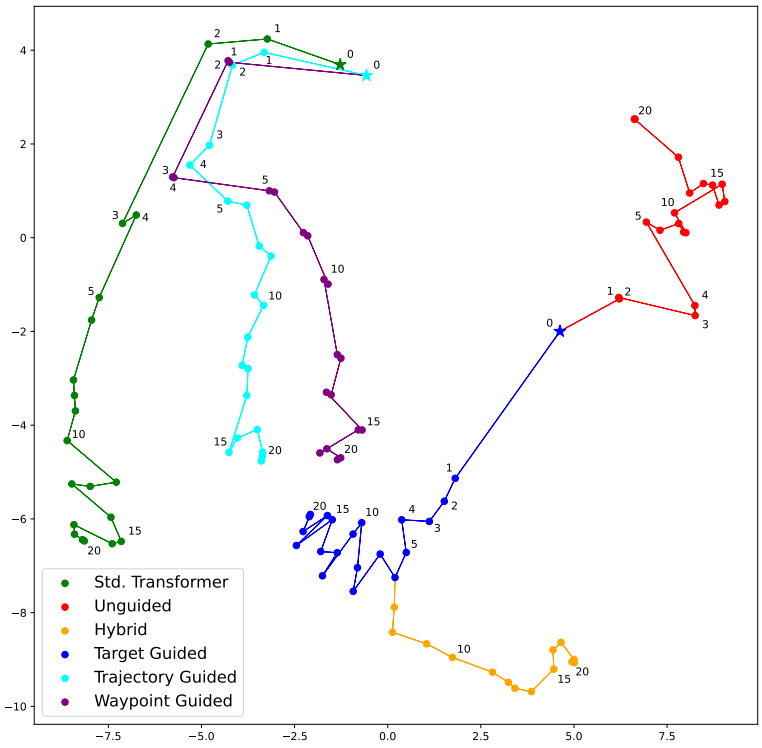


Figure 5.4: The trajectory of hidden states during training under different modes of guidance, visualized using t-SNE.

(CALD) approach, and further enhance CALD with a trajectory-based guidance or a hybrid approach. Through our empirical experiments, we confirm that CALD can effectively convert the transformer into an efficient linear-complexity model with performance close to or on par with the standard transformer, much better than directly converted models. We also verify the versatility of CALD by experiments on multiple tasks with various speech and language models. We further identify the tasks where the enhancement approaches are effective in improving the results. Taken together, these findings offer a practical path toward building linear-complexity models under constrained computational budgets, and contribute to the broader study of efficient attention, which is a central theme of this thesis.

6 Hybrid Linear-Sparse Attention

The previous chapter provides a cost-effective approach to build efficient linear attention models, but to put such models into practical use, there is still a critical bottleneck. Linear attention models compress the entire input sequence into a fixed-size recurrent state offer an efficient alternative to Transformers, which induces forgetfulness that harms retrieval-intensive tasks. To mitigate the issue, this chapter explores a series of hybrid models that restore direct access to past tokens. We interleave token mixers of intermediate time and space complexity between linear attention and full attention, including the query-aware native sparse attention, and sparse attention with token eviction. We further propose a novel learnable token eviction module. Combined with sliding-window attention, an end-to-end trainable lightweight CNN-based eviction module aggregates information from both past and future adjacent tokens to adaptively retain a limited set of critical KV-pairs per head, maintaining linear attention's constant time and space complexity. Empirical evaluations on retrieval-intensive benchmarks support the effectiveness of our approaches.

6.1 Introduction

Large language models (LLMs) highlight the effectiveness of the transformer architecture (Vaswani et al., 2017). By explicitly scoring pairwise token interactions, transformers excel in utilizing information from the distant past, and scale efficiently through parallel training on modern GPUs. These advantages enable transformers to supplant recurrent neural networks (RNNs). However, the blessing of pairwise attention is also a curse: compute time scales linearly with context length per query (i.e. quadratically over a sequence), and the memory use of key-value (KV) cache grows linearly. These costs become especially problematic for settings such as agents with ultra-long accumulated context. This renewed interest in hardware-efficient recurrent or linear attention token mixers such as Mamba (Gu and Dao, 2024) and DeltaNet (Yang, Wang, Zhang, et al., 2024). Like classical RNNs, these models compress the entire history into a fixed-size memory recurrently updated for each new token (Figure 2.1 (a)), yielding $O(1)$ time and space per step while remaining competitive with transformers on many language tasks. However, this also comes with a cost: a variable-length history can never be losslessly compressed into a fixed-size state. As time elapses, information from distant tokens inevitably decays, resulting in diminishing performance on retrieval-intensive tasks compared to standard transformers, even at modest context lengths (Wen, Dang, and Lyu, 2025; Jelassi et al., 2024; Arora, Eyuboglu, Timalsina, et al., 2024; Arora, Eyuboglu, Zhang, et al., 2024).

Several approaches attempt to alleviate this forgetfulness. A prominent direction is to improve the update rule itself to increase memory expressiveness and update selectivity (Gu and Dao, 2024; Yang, Kautz, and Hatamizadeh, 2024; Du et al., 2025), without altering the recurrent nature. Others interleave full and linear attention layers to create hybrid models (Lenz et al., 2025; Waleffe et al., 2024; MiniMax et al., 2025). While this combination or compromise of two worlds allows more effective direct retrieval of past memory, it also reintroduces higher complexity. This motivates a natural question: Can we directly access past tokens for better retrieval, without much sacrifice in computational complexity?

Other than a complete fallback from linear to full attention, a broad class of sparse attention of intermediate time and space complexity lies between the two extremes. A promising approach is query-aware sparse attention (Figure 2.1 (c)) like Native Sparse Attention (NSA) (Yuan et al., 2025). Such models have a lightweight *probing* step to first compare the current query against all past KVs, then perform actual attention only with a few selected KVs. This motivates our hybrid **laNSA** model that interleaves **l**inear **a**ttention and **NSA** layers to enable direct retrieval of past tokens while largely preserving favorable time complexity.

This nevertheless introduces the overheads of the probing stage, and requires access to all past tokens and thus $O(N)$ memory, undermining linear attention's constant-space appeal. Ideally, the model would instead *predict* whether the current token will matter for future queries, and proactively *evict* irrelevant tokens in advance, retaining only $O(1)$ tokens in the working memory (Figure 2.1 (b)), much like people jotting down brief notes when listening to a long speech. This allows direct access to the distant past, but suffers from the risk of forgetting:

once evicted, a token is no longer available to any future query. An accurate importance metric is henceforth of utmost importance, but most prior work relies on fixed and handcrafted heuristics. By contrast, we introduce a novel contextualized learnable token eviction (LTE) mechanism, which combines sliding-window attention (SWA) and a lightweight convolution neural network (CNN) to predict per-head KV importance from both past and future adjacent context. Trained end-to-end under sparsity regularization, LTE allows each sparse attention layer to retain the most relevant KVs under a learned per-head cache budget. We then build our **laLTE** model with interleaved linear and LTE-sparse attention layers. We further approach actual speedup with a dedicated KV cache layout, fused Triton kernels for SWA+KV-sparse attention, and a lazy scoring scheme. In this design, laLTE preserves the constant time and space complexity of linear attention.

We pretrain language models at 0.4B and 1.4B scale, and evaluate them on several short-context and long-context/retrieval-intensive benchmarks. Both laLTE and laNSA outperform baselines including hybrids with SWA or heuristic eviction, and approach full transformer performance. To summarize, our contributions are: 1) propose a novel contextualized learnable token eviction mechanism for sparse attention under strict time and space constraint; 2) build hybrid laNSA and laLTE architectures with efficient implementation; 3) examine the performance of hybrid linear attention models across the token mixer hierarchy. Our source code and models are publicly available at https://github.com/idiap/hybrid-linear-sparse-attention.

6.2 Related work

The general background on linear and sparse attention has been introduced in Chapter 2.3. Below we further introduce specific methods relevant to our work, and how they relate to our proposed approaches.

Token eviction As introduced in Chapter 2.3.2, query-agnostic sparse attention or token eviction strategies develop criterion to determine the importance of a token/KV pair without knowing future queries. In addition to fixed sparsity patterns like SWA, there are many methods to dynamically measure token importance. For example, as high past attention score (i.e. more contribution to past outputs) often correlates with future importance, a common strategy is to retain tokens with top-K accumulated attention scores over the full history (Wang, Zhang, and Han, 2021; Zhang, Sheng, et al., 2023; Jo and Shin, 2024), in a window (Liu, Desai, et al., 2023; Zhao, Wu, and Wang, 2024; Oren et al., 2024; Li, Huang, et al., 2024), or to specific query tokens (Chen, Wang, Shang, et al., 2024; Kim, Shim, et al., 2024). Layer- and head-wise variation of required budget further motivates non-uniform budget heuristics (Yang, Han, et al., 2024; Cai et al., 2024; Qin, Cao, et al., 2024) or global allocation (Feng, Lv, et al., 2025; Zhou et al., 2025). These methods are training-free and largely heuristic. However, some content can be *typically* important despite lacking high local importance or attention, such as infobox in Wikipedia. Furthermore, extracting attention scores that are not materialized in Flash Attention (FA) (Dao,

Fu, et al., 2022) adds implementation complexity and overheads.

In contrast to attention score heuristics, a more potent but underexplored alternative is learnable token eviction. An early attempt, ColT5 (Ainslie, Lei, et al., 2023), selects tokens with a linear scorer, taking each token as input without considering the context. Anagnostidis et al. (2023) use attention score criteria but add a trainable per-layer bias term. Zeng et al. (2024) train a separate attention layer for LTE, which ingests full context but incurs heavy quadratic computation and is limited to prefilling stage. More recently, Łańcucki et al. (2025) fine-tune LLMs with a global compression ratio constraint, which allows many layers to remain uncompressed and contradicts our $O(1)$ complexity goal. Piękos, Csordás, and Schmidhuber (2025) train small scale LMs under predetermined cache budget, without considering layer and head variations. Shi, Wu, et al. (2025) predict importance scores that modulate attention scores directly, departing from standard attention and increasing implementation complexity. More importantly, these works are limited to single-layer and per-token LTE scorers. To our knowledge, this chapter is the first to introduce a lightweight contextualized LTE variant.

Query-aware sparse attention Methods above try to predict future relevance without observing future queries, which can be fallable. If we relax space constraints and retain all past KVs, we can select $\mathscr{I}_i = \text{select}(\boldsymbol{q}_i, \boldsymbol{K}, \boldsymbol{V})$ in a $\boldsymbol{q}_i$-aware manner using an $O(N)$ but lightweight probing step before actual attention, as introduced in Chapter 2.3.2. Early methods use online hashing and clustering (Kitaev, Kaiser, and Levskaya, 2020; Roy et al., 2021). GPU-efficient blockwise approximation is highlighted in more recent approaches, such as **Mixture of Block Attention** (MoBA) (Lu, Jiang, et al., 2025) which is trained end-to-end during pretraining. We further adopt the stronger and more sophisticated **Native Sparse Attention** (NSA) (Yuan et al., 2025), which combines SWA, attention to compressed KV blocks, and MoBA-like sparse attention. Other trained variants directly predict block- or token-level attention scores (Gao, Zeng, et al., 2025; DeepSeek-AI, 2025), approximate scores in lower rank (Singhania et al., 2024; Tan et al., 2025), combine NSA branches (Zhao, Zhou, et al., 2025), or insert per-block landmark token per block as $\boldsymbol{k}^B$ (Mohtashami and Jaggi, 2023). Many other methods target training-free setting and use heuristic block summaries such as mean-pooling as $\boldsymbol{k}^B$ (Tang, Zhao, et al., 2024; Chen, Wang, Cao, et al., 2024; Jiang, Li, Zhang, et al., 2024; Xiao, Zhang, et al., 2024; Lu, Zhou, et al., 2024; Zhang, Xiang, et al., 2025; Lee, Park, et al., 2025); while we focus on the trained setting to best fit the model to sparse context.

Hybrid attention As discussed in Chapter 2.3.4, a direct way to mitigate the forgetfulness of linear attention is to put attention back. A small number of full attention layers can substantially improve retrieval performance (Arora, Eyuboglu, Timalsina, et al., 2024; Glorioso et al., 2024; Waleffe et al., 2024; Wang, Zhu, Abreu, et al., 2025; Yang, Rezagholizadeh, et al., 2025; Lenz et al., 2025; Jamba Team et al., 2024; Bae et al., 2025; MiniMax et al., 2025; Tencent Hunyuan Team et al., 2025; Qwen Team, 2025; Kimi Team et al., 2025a; Ling Team et al., 2025) but at higher computational cost. In contrast, we prioritize preserving time and space complexity, by

introducing token mixers of intermediate costs. Interleaving SWA layers has been also shown to markedly improve linear attention performance (Ren, Liu, Lu, et al., 2024; Zuo, Liu, et al., 2024; Yang, Kautz, and Hatamizadeh, 2024; Arora, Eyuboglu, Zhang, et al., 2024; Cabannes et al., 2025; Behrouz, Zhong, and Mirrokni, 2024; De et al., 2024; Lan, Sun, et al., 2025; Irie, Yau, and Gershman, 2025; Dong, Fu, et al., 2025; Fang, Yu, et al., 2025). Closer to laLTE, Ren, Liu, Wang, et al. (2023) and Nguyen, Nguyen, et al. (2024) study hybrid models that learn to perform SWA on selected query tokens. These works nevertheless target a distinct regime: While SWA can be viewed as an eviction policy, it does not grant access to distant tokens, even when stacked together (Xiao, 2025). Only few concurrent or very recent works explore the combination of linear and other sparse attention: SE-Attn (Nunez et al., 2025) converts attention layers in hybrid Mamba2 into query-aware sparse attention, MiniCPM-SALA (MiniCPM Team et al., 2026) converts transformer models into hybrid of query-aware sparse attention and linear attention models, LoLA (McDermott, Jr, and Parhi, 2025) and hybrid associative memory (Lufkin et al., 2026) keeps an inference-time side memory of tokens difficult for linear attention to memorize, HAX (Zhan et al., 2025) combines Mamba with hashing-based sparse attention and learned key selection, and RWKV-X (Hou, Huang, et al., 2025) combines RWKV with MoBA-like attention and block eviction heuristics; none of them explored learnable contextualized token eviction.

6.3 Methods

6.3.1 Learnable Token Eviction

LTE module

Our LTE module predicts a per-token, per-head retention score $r_{j,h}$ for each KV pair. At decoding step i and head h, attention is restricted to retained KVs, the SWA window, and the attention sink: $\mathscr{I}_{i,h} = \{j : j \leq i, (j \leq s) \vee (r_{j,h} > 0.5) \vee (j \geq i - w)\}$, which yields an A+column-shaped attention pattern in Figure 6.1. To determine retention accurately and efficiently, we anchor our LTE module design around four principles:

1. **Fine-grained**: Attention patterns vary substantially across layers and heads: some are sharply focused on a few tokens, some form local clusters, and others are more dispersed. We therefore make eviction decisions at the level of individual KV pairs and heads, rather than by block or layer, to better utilize each head's budget and capture head-specific behavior.

2. **Contextualized**: To better understand each token and estimate its importance, we consider not only each token's own KV but also its local past and future context. Since future is unavailable to causal/recurrent layers, we introduce a 1D CNN with total receptive field $R = 13$ tokens, which possesses local look-ahead capabilities thanks to SWA: recent KVs in a window $w \gg R$ are cached, hence we can defer computing r_j by $R//2$ steps during decoding until all KVs within the receptive field are readily available.

Figure 6.1: Left: Illustration of our eviction scheme, combining LTE, an attention sink (size $s = 2$), and SWA ($w = 12$), with 4 KV heads. A CNN stack predicts a per-token, per-head retention score $r_{j,h}$ when its receptive field is fully covered by the SWA window, which determines whether a KV pair is evicted once it leaves the window. With recent KVs cached by SWA, the CNN can read short-range past and future context. Right: The resulting per-head A-plus-column sparsity pattern; colored tiles denote tokens selected at each step.

3. **Independent**: Inspired by ReMoE (Wang, Zhu, and Chen, 2024), we evaluate each token independently rather than enforcing global scheduling, such as selecting top-K tokens per head as in other eviction methods. This reduces computational overheads, and more importantly yields more adaptivity, flexibility, and fine-grained budgeting than manually allocating a fixed per-head budget K.

4. **Lightweight**: We prioritize efficient and parallel inference, hence LTE uses a small 3-layer 1D CNN with progressively narrowing width, adding only about 1% parameters.

Figure 6.1 sketches the full eviction scheme. The LTE module consumes, for each head, the concatenated key and value vectors prior to rotary positional encodings (RoPE) (Su et al., 2024) for better position generalization. Convolutions are applied independently to each head and parallelized across heads via grouped 2D convolutions, producing a scalar in the end. We then apply sigmoid to obtain $r_{j,h}$, unlike ReMoE for better training stability, and binarize with a threshold of 0.5. We also force the first $s = 4$ tokens to retain as attention sinks. During training, we use FlexAttention (Dong, Feng, et al., 2024) for efficient sparse attention.

Adaptive training

Training LTE is nontrivial as it outputs only r scores that determine a discrete attention mask, so the gradient cannot be back-propagated directly to LTE parameters. Instead, we leverage straight-through estimator that conceptually applies an all-one mask m on the value vectors $v'_{j,h} := v_{j,h} \cdot m_{j,h}$, and use gradients on $m_{j,h}$ as a surrogate gradient to $r_{j,h}$:

$$\frac{\partial \mathscr{L}}{\partial r_{j,h}} := \frac{\partial \mathscr{L}}{\partial m_{j,h}} = \left\langle \boldsymbol{v}_{j,h}, \frac{\partial \mathscr{L}}{\partial \boldsymbol{v}'_{j,h}} \right\rangle, \tag{6.1}$$

Under this formulation, the gradient on $r_{j,h}$ is tied with the contribution to the loss of the underlying $v_{j,h}$ it controls, and we find that it enables stable end-to-end training.

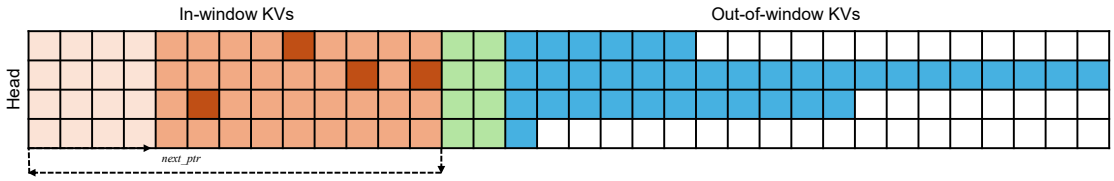


Figure 6.2: Illustration of the KV cache layout in LTE inference, with the same color scheme as Figure 6.1. It contains a circular SWA segment for in-window KVs and a capped out-of-window segment for LTE-retained KVs and the attention sink. Retention scores are computed in batch and deferred as much as possible.

Some heads mainly focus on a few tokens, and their attention patterns naturally become sparse under this mechanism alone: For unimportant $v_{j,h}$, the all-one $m_{j,h}$ is pulled to zero by a negative gradient, which is copied to $r_{j,h}$. Other heads exhibit more diffuse attention and benefit from an explicit sparsity prior. Inspired by ReMoE (Wang, Zhu, and Chen, 2024), we impose a dynamically weighted L1-style penalty to encourage sparsity of retained tokens:

$$\mathscr{L}_{sparse} = \sum_h \lambda_h \sum_i \mathrm{ReLU}(r_{i,h} - 0.5) \tag{6.2}$$

We target a per-head retention cap $b = 512$, i.e. $c_h = \sum_i \mathbf{1}[r_{i,h} > 0.5] \leq b$. During training, λ_h is adjusted by feedback: it increases when the moving average of c_h exceeds b, and decreases when it falls below $0.95 \cdot b$, with an upper bound $\lambda_h \leq 1$ for stability. We observe that many heads quickly drive λ_h to zero, suggesting that LTE gradient signals alone suffice to induce adequate sparsity. In this way, the memory budget of each head emerges from end-to-end training using gradient signals on the values, without the need of handcrafted budget allocation rules (e.g. setting a per-layer K). More details are provided in Appendix C.

Inference implementation

Capped capacity Our models are trained on fixed-length sequences ($N = 4096$ in our experiments). Although the sparsity penalty regularizes the number of retained tokens, the actual retention pattern remains input-dependent and no hard cap is enforced during training. At inference time, however, we expect predictable time and space complexity for arbitrary inputs of arbitrary length. We therefore resort to a hard global budget: at most b out-of-window KVs are cached per head.

Cache prefilling We first allocate and fill a fixed-size cache, illustrated in Figure 6.2, consisting of two segments: one holds the recent w KVs within the SWA window, and other gathers out-of-window KVs retained by LTE or forced by the attention sink. This yields contiguous KVs amenable to tiled FlashAttention, avoids memory management intricacies due to cache size imbalance between heads, and prevents worst-case slowdowns when a head is unusually dense. Specifically, since we store post-RoPE KVs, physical positions in memory are irrelevant,

allowing arbitrary reordering and compaction. We therefore store the SWA segment in a circular buffer with a maintained `next_ptr`, and compact retained out-of-window KVs into a contiguous span, similar to Pagliardini et al. (2023). If the capacity happens to exceed on a certain head, we keep those with top-b scores. We then compute SWA+sparse attention by a FlashAttention-style Triton kernel. For each query tile, attention is computed in two stages: (1) SWA over original KV tiles; and 2) sparse attention using compacted tiles. To avoid double counting, compacted KVs that fall inside the SWA window are masked out, and tiles completely inside the window or violating causality are skipped.

Cached decoding During decoding, each new KV is pushed into the SWA buffer and the leftmost KV is popped. If its $r > 0.5$, it is copied into the next free slot of the out-of-window segment. If the segment is also full, it replaces the current lowest-score entry only if it scores higher. We implement this cache-replacement step as a fused Triton kernel for efficiency. Now there is only one query token, and when the SWA buffer is full all its relevant KVs are contiguous. In this case we can directly utilize the existing highly-optimized `flash_attn_with_kvcache` API that supports `cache_seqlens` during decoding.

Lazy scoring We need not compute r_j as soon as it becomes available, i.e. when current N reaches $j + R//2$. LTE scoring can be deferred and batched as long as the entire receptive field remains inside the SWA window. Accordingly, we trigger LTE scoring only when an uncomputed r_j depends on a token about to leave the window, i.e. when $N - w = j - R//2$. At that point we rotate the buffer so that *next_ptr*=0, and score all tokens from $N - w + R//2$ to $N - R//2$ in a batch. A further complication is that LTE consumes pre-RoPE keys, whereas KV cache stores post-RoPE keys. We therefore undo RoPE on cached keys by applying the inverse rotation. With RoPE implemented using cached sinusoids, this is equivalent to reapplying RoPE with negated sine values at offset $N - w$. Using this trick, scores are computed every $w - R$ steps in parallel, incurring minimal overheads.

6.3.2 Hybrid modeling

We use Gated DeltaNet (GDN) as both the baseline and linear attention backbone thanks to its relatively strong retrieval performance. Following the GDN+SWA design of Yang, Kautz, and Hatamizadeh (2024) (GDN-H1), we interleave GDN layers and alternative token-mixer layers in a 1:1 ratio for a simplified comparison. We use $w = 1024$ for SWA and $w = 768$ for LTE. Although the LTE cache is capped at $b = 512$, the observed average number of retained tokens per head is about 256, making the effective total memory comparable.

Table 6.1: Results (perplexity and accuracy%) on short-context tasks; best numbers bolded.

Model	Wiki. ppl↓	LMB ppl↓	LMB acc↑	PIQA acc↑	Hella acc_n↑	Winog. acc↑	ArcE acc↑	ArcC acc_n↑	BoolQ acc↑	SciQ acc↑	SIQA acc↑	Avg.↑
0.4B Params.												
GDN	27.56	36.06	31.9	66.4	39.1	50.9	56.8	25.9	58.4	71.0	37.5	48.7
GDN+SWA	26.99	33.87	33.7	66.4	39.6	51.5	56.4	27.9	60.2	72.4	38.5	49.6
laLTE	26.65	**30.03**	34.4	66.1	**40.2**	**52.8**	57.2	26.3	**61.8**	**75.6**	38.1	**50.3**
laNSA	26.84	36.19	33.2	66.1	39.9	51.8	**58.0**	**28.3**	61.6	73.1	37.7	50.0
GDN+Attn.	**26.61**	33.50	**34.6**	**66.5**	39.8	51.6	57.4	27.0	55.8	74.9	38.2	49.5
NSA	27.70	39.33	33.5	66.4	38.7	50.8	55.8	28.1	59.7	69.4	37.3	48.8
Transf.++	27.97	40.36	32.8	65.9	38.3	51.1	56.8	28.0	61.2	71.0	**38.6**	49.3
1.4B Params.												
GDN	18.31	15.27	43.2	70.9	52.4	54.6	67.8	34.4	58.0	**83.9**	40.4	56.2
GDN+SWA	18.53	14.77	43.7	70.4	51.7	55.0	66.8	35.2	61.4	81.5	40.2	56.2
laLTE	**17.99**	14.79	43.9	71.1	**54.0**	**55.2**	**68.4**	**35.8**	60.6	83.4	39.4	**56.9**
laNSA	18.16	**14.73**	**44.9**	**71.2**	52.4	54.6	67.3	35.5	59.9	82.4	**40.8**	56.6
GDN+Attn.	18.68	16.10	43.2	69.6	51.0	53.5	66.3	32.8	53.8	81.5	40.1	54.6
NSA	19.53	18.48	42.2	68.8	50.1	53.0	65.6	34.0	59.1	80.5	39.1	54.7
Transf.++	20.68	20.17	39.1	68.1	47.2	52.0	63.8	33.5	**62.5**	80.5	38.8	54.0

6.4 Empirical studies

We evaluate hybrid attention models that interleave GDN with token mixers spanning the hierarchy in Figure 2.1: sliding-window attention (GDN+SWA), learnable token eviction (laLTE), native sparse attention (laNSA), full-attention (GDN+Attn.), along with several ablated variants of LTE. We also compare against pure GDN, pure NSA, and a strong transformer (Transf.++) following Gu and Dao (2024). With limited computational resources, we train 0.4B- and 1.4B-sized models with 10B and 30B tokens from FineWeb-Edu (Penedo et al., 2024) from scratch. We implement LTE layers, hybrid models, and inference kernels for LTE and NSA on top of Flash Linear Attention (Yang and Zhang, 2024). Below we leverage both short-context and long-context/retrieval-intensive benchmarks to assess models above and to demonstrate the effectiveness of our laLTE and laNSA approaches, and compare the prefilling and decoding efficiency. More details are available in Appendix C.

6.4.1 Short-context benchmarks

We first evaluate zero-shot commonsense tasks. As these inputs are typically shorter than the SWA window, denser attention will not show their advantage in long-term retrieval, and performance differences reflect only knowledge capacity and training stability. This is confirmed by results in Table 6.1. At 0.4B GDN slightly lags behind Transf.++, but at 1.4B, GDN slightly outperforms Transf.++, while NSA and GDN+Attn. fall in between. GDN+SWA performs similar or slightly better, while adding LTE or NSA yields small gains, with laLTE showing best average accuracy. Overall, our observations corroborate Yang, Kautz, and Hatamizadeh (2024) that linear attention and hybrid models match or slightly exceed transformers on these tasks, further demonstrated by our laLTE and laNSA.

Table 6.2: Results (%) on recall-intensive EVAPORATE tasks; best linear-time results bolded.

Model	FDA	SWDE	NQ	SQUAD	TQA	DROP	Avg.
0.4B Params.							
GDN	4.17	8.37	10.90	27.95	47.69	17.73	19.47
GDN+SWA	5.08	18.27	9.34	**34.52**	49.11	20.51	22.81
laLTE	15.52	19.26	**11.15**	32.98	48.99	**23.67**	25.26
laNSA	**17.33**	23.13	10.71	31.60	**49.17**	21.56	**25.58**
GDN+Attn.	29.31	26.19	11.53	30.53	46.92	18.69	27.19
NSA	7.99	**26.28**	10.01	33.91	47.16	19.74	24.18
Transf.++	28.58	26.46	11.09	31.94	46.68	17.30	27.01
LTE-MLP	12.70	18.63	11.15	31.27	49.76	20.17	23.95
Pure LTE	14.25	18.72	7.44	33.31	46.15	19.45	23.22
1.4B Params.							
GDN	20.96	25.83	**17.96**	35.02	56.46	20.75	29.50
GDN+SWA	26.68	27.99	16.53	41.09	58.95	**23.29**	32.42
laLTE	28.68	28.98	17.68	**43.00**	**60.60**	22.62	33.59
laNSA	**31.85**	**36.99**	17.20	38.07	58.06	20.60	**33.80**
GDN+Attn.	46.64	41.13	18.09	40.21	58.47	22.57	37.85
NSA	25.59	34.83	13.91	40.65	56.28	23.05	32.38
Transf.++	38.84	34.92	16.19	37.60	55.92	21.66	34.19
TOVA	20.96	30.78	13.49	40.18	58.59	22.62	31.10
Unif.+SWA	15.97	24.21	11.82	22.32	57.17	15.81	24.55

6.4.2 Retrieval-intensive benchmarks

We then evaluate on single needle-in-a-haystack (S-NIAH) from RULER (Hsieh et al., 2024) and the EVAPORATE suite (Arora, Eyuboglu, Zhang, et al., 2024). S-NIAH spans tasks from retrieving numbers in synthetic texts (S1) to the more difficult extraction of long UUIDs from real essays (S3). We evaluate contexts of 1K–4K where models at our scale can attain non-trivial accuracy. EVAPORATE, a benchmark challenging for linear attention models and standard in the area, comprises realistic QA-style retrieval over passages up to 4K tokens. Since these tasks require retrieving from distant past, we expect hybrid models with more direct and accurate access to past tokens to perform better; under tight time and/or space constraints, this should favor laLTE and laNSA. Meanwhile, access to past tokens does not always guarantee successful retrieval, higher computational complexity does not monotonically improve accuracy, and outcomes remain task-dependent as observed by Wang, Zhu, Abreu, et al. (2025).

Results in Table 6.2 and Table 6.3 confirm the effectiveness of hybrid models, laLTE in particular. Specifically, pure GDN excels on S-NIAH and, at 1.4B, approaches full transformers, echoing Yang, Kautz, and Hatamizadeh (2024). However, GDN lags on the more realistic EVAPORATE QA tasks. GDN+SWA improves substantially on EVAPORATE, yet perform much worse on S-NIAH, especially beyond its 1K window size and at 1.4B. GDN+Attn. achieves the best EVAPORATE results across scales and remains competitive with full transformers and pure

Table 6.3: Results (%) on RULER single needle-in-a-haystack tasks; numbers >80% bolded.

Model	S1-1K	S1-2K	S1-4K	S2-1K	S2-2K	S2-4K	S3-1K	S3-2K	S3-4K	Avg.
0.4B Params.										
GDN	**99.8**	**92.8**	45.2	**100.0**	**94.8**	32.4	0.2	0.2	0.0	51.7
GDN+SWA	**100.0**	52.0	27.0	61.2	67.2	27.4	75.4	64.8	17.4	54.7
laLTE	**99.8**	**86.8**	36.4	**99.8**	74.2	25.0	**87.2**	42.8	17.8	63.3
laNSA	**100.0**	**100.0**	65.0	**88.6**	**100.0**	37.6	**97.6**	73.0	11.8	74.8
GDN+Attn.	**100.0**	**100.0**	77.0	**100.0**	**99.6**	52.6	**95.4**	**87.0**	12.2	**80.4**
NSA	**100.0**	**98.8**	46.6	**100.0**	**95.8**	36.4	**94.4**	64.6	18.6	72.8
Transf.++	**100.0**	**99.6**	50.8	**100.0**	**100.0**	55.4	62.8	71.4	35.8	75.1
LTE-MLP	**95.6**	50.0	23.0	**100.0**	56.6	24.4	25.4	32.0	16.2	47.0
Pure LTE	**94.4**	37.8	21.4	74.2	54.6	22.6	**85.8**	37.6	15.0	49.3
1.4B Params										
GDN	**100.0**	**100.0**	**100.0**	**93.6**	**99.0**	**88.0**	**93.0**	**77.6**	49.0	**88.9**
GDN+SWA	**100.0**	52.0	29.6	**93.4**	**83.2**	34.2	**92.8**	57.0	19.4	62.4
laLTE	**100.0**	**99.2**	**95.0**	**100.0**	**98.0**	**81.4**	**85.4**	55.6	33.2	**83.1**
laNSA	**100.0**	**99.8**	57.8	**80.6**	**100.0**	68.2	60.4	**94.2**	34.0	77.2
GDN+Attn.	**99.8**	**100.0**	78.4	**100.0**	**100.0**	**90.6**	**83.8**	**71.0**	58.0	**86.8**
NSA	**100.0**	**93.8**	36.2	**100.0**	**99.4**	40.2	**97.8**	**78.6**	26.8	74.8
Transf.++	**100.0**	**100.0**	**98.4**	**100.0**	**100.0**	**85.6**	**89.4**	**84.2**	64.8	**91.4**
TOVA	**99.8**	**81.6**	41.0	**100.0**	11.4	8.0	**83.8**	0.0	0.8	47.4
Unif.+SWA	**94.2**	37.6	20.2	**100.0**	54.4	24.2	**84.8**	37.2	23.0	52.8

GDN on S-NIAH, but its computational cost undermines our efficiency goals.

As for our models, laNSA achieves the best EVAPORATE performance among linear-time methods, approaches full transformers on 0.4B S-NIAH, and outperforms pure NSA. However it trails full transformer and GDN+Attn. on EVAPORATE, and weaker on 1.4B S-NIAH despite theoretical access to the full history. It also underperforms laLTE on shorter-context retrieval (e.g. S3-1K). Possible reasons include interference between different attention branches (Hu et al., 2025), and its small number of KV heads: NSA requires at least 16 group size for grouped-query attention. In our models with relatively small hidden sizes and head counts, NSA is effectively reduced to near multi-query attention with only 2 KV heads, which can be a performance bottleneck and a factor of instability. LaLTE stands out against this backdrop: Despite similar time and space complexity as GDN and GDN+SWA, it outperforms both on EVAPORATE, and nearly matches laNSA on EVAPORATE and 1.4B GDN+Attn. on S-NIAH despite limited KV cache. This highlights laLTE's cost-effectiveness and versatility for long-context retrieval under tight complexity constraints.

We further ablate laLTE against GDN+following token mixers with constant time and space complexity: **LTE-MLP** replaces CNN with a 2-layer MLP with no access to the future context; **Pure LTE** is a pure transformer with all layers sparsified by LTE; **Unif.+SWA** retains out-of-window tokens at a uniform stride; **TOVA** is one of the state-of-the-art eviction heuristics

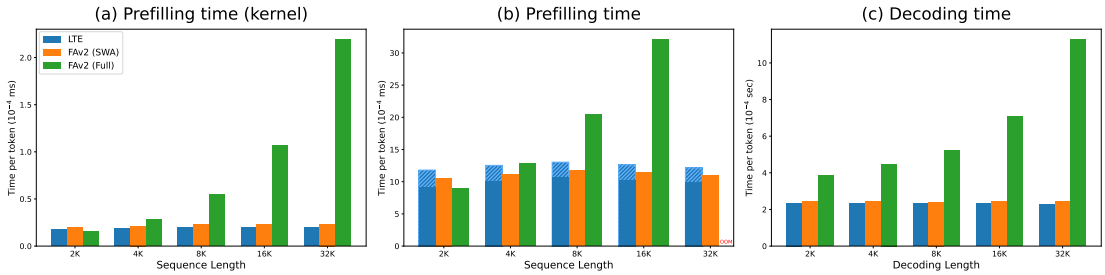


Figure 6.3: Runtime comparison of LTE, SWA and full attention with FlashAttention v2, on (a) attention kernels prefilling with artificial KVs; and hybrid layers with real inputs during (b) prefilling and (c) decoding with a 4K-token prompt, under 1.4B configurations with batch size=32. The shaded portion of the LTE bar in (b) indicates LTE scoring overheads. Note that decoding is substantially more time-consuming than prefilling. Overall, LTE and SWA have similar runtime and much faster than full attention.

based on accumulated attention scores (Oren et al., 2024). The latter two are training-free, and evaluated by applying them to pretrained GDN + Attn. for which we use an identical 1024 total memory size. Although LTE-MLP and TOVA achieve moderate results on EVAPORATE, better than pure GDN, all of them underperform laLTE, especially on S-NIAH; while Unif.+SWA performs much worse than all other models. This confirms the importance of an accurate eviction rule, and underscores the effectiveness of our design of contextualized and learnable eviction. The observation is similar on Pure LTE, which highlights the complicacy of the synergy between sparse and linear mechanisms: Improving a hybrid attention model cannot be reduced to improving each of the linear and softmax attention components, and LTE works much better when combined with linear attention, possibly thanks to the compressed memory to compensate for evicted tokens. To summarize, the results underscore the value of direct and accurate access to historical context for retrieval-intensive tasks. From GDN, GDN+SWA/laLTE, laNSA, to GDN+Attn., increasingly complete access to past tokens generally improves long-context performance, although gains remain task-dependent. In particular, laLTE offers a strong performance-efficiency trade-off under rigorous time and space constraints.

6.4.3 Efficiency

We next benchmark the actual efficiency of LTE on a single H100-SXM GPU. Figure 6.3 (a) shows the runtime of prefilling kernels at different sequence lengths, compared with SWA and full attention using the highly-optimized FlashAttention v2 (Dao, 2024) kernels. For LTE, we use an artificial half-full out-of-window KV-cache, with 1024 tokens matching the SWA window size. Our SWA+sparse attention kernel exhibits runtime scaled linearly with the input length, comparable to SWA and much faster than full attention. While for LTE, eviction is input dependent and actual retention rates vary by layer and head; LTE scoring and caching also introduce overheads. We therefore also measure the actual model prefilling and decoding time on real inputs. To better isolate the relevant costs, we only time the hybrid layers, excluding

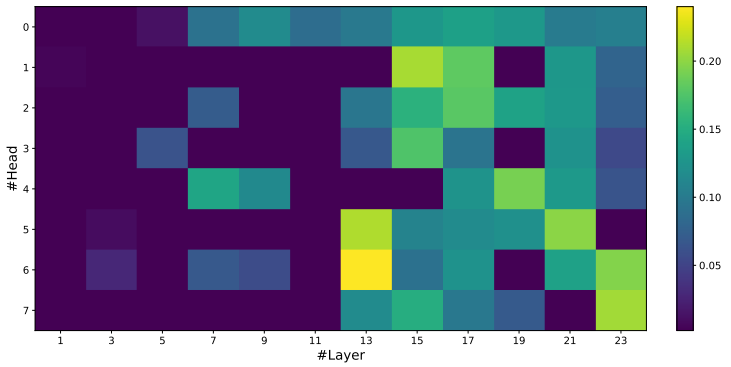


Figure 6.4: Average retention rate per head per layer produced by a laLTE model.

other layers like GDN. Figure 6.3 (b, c) again confirms the efficiency of LTE. During prefilling, scoring adds some overhead, but this is largely offset by faster attention thanks to layers and heads with few retained KVs. During decoding, which dominates inference time, lazy scoring makes the overhead negligible, and LTE runs slightly faster than SWA. Furthermore, LTE uses a constant-size KV-cache equal to 1280 tokens, i.e. 2.5MiB per sample per layer in our 1.4B setting. Relative to full KV cache with longer context, this enables larger batch sizes and higher token throughput given limited memory. Taken together, these results support adopting LTE in lieu of SWA for better performance without much efficiency degradation.

6.4.4 Retention pattern

Figure 6.4 shows the average memory retention rate per head per layer from the out-of-SWA part of 16 samples drawn from the training data produced by our l.4B laLTE model. We find that retention rates are highly varied across layers and heads, while the higher layers are often denser. This is similar to findings in other works, e.g. Łańcucki et al. (2025), except that dense heads do not present in lowest layers; The interleaved GDN layers may have a role in it. The observation also supports the necessity of having a fine-grained cache budget different from head to head and from layer to layer.

6.5 Conclusion

We study hybrid models that improve the retrieval ability of linear attention by interleaving token mixers with different time-space trade-offs. We propose two variants: laNSA, which interleaves query-aware native sparse attention and achieves sub-quadratic time complexity when memory capacity permits; and laLTE, which performs token eviction with a learnable contextualized module, preserves constant space complexity, and supports an efficient decoding algorithm. Empirical results confirm the effectiveness of our approaches, particularly highlighting laLTE as a strong candidate for hybrid linear-sparse attention with improved retrieval performance. Although these methods are not evaluated directly on SLU tasks, they contribute to a central theme of the thesis: developing architectures for efficient and effective

modeling of extended sequences.

7 Application to Interview Analysis

This chapter returns to the original motivation of interview analysis and looks into a specific aspect, namely storytelling. More specifically, structured interviews often include past-behavior questions inviting applicants to recount a past work experience. While optimal responses to these questions should take the form of a story, applicants struggle to produce them extemporaneously. Asynchronous video interviews (AVIs) present new opportunities for job interview coaching, which can incorporate artificial intelligence to analyze audio-recorded responses and deliver personalized feedback. We explore the potential of audio-based deep-learning models to the specific SLU task of identifying storytelling and other, sub-optimal responses (pseudo-stories, decontextualized self-descriptions) from interview audio recordings. Using data from 254 mock interviews featuring three past-behavior questions, we developed models to determine the utterance type, considering different scenarios and labeling schemes of varying granularity. We further applied multiple techniques to improve the model accuracy. Findings show that our models achieve satisfactory performance when enhanced with audio information and enriched with longer context. However, providing paralinguistic cues from the audio recordings brings limited benefit to the models' performance in our experiments. We discuss the results, implications, and future research directions.

The work in this chapter has been published as:

Elisabeth Germanier, Mutian He, et al. (Aug. 2025). "Identifying storytelling in job interviews using deep learning". In: *Computers in Human Behavior Reports* 19, p. 100688. DOI: 10.1016/j.chbr.2025.100688

7.1 Introduction

Job interviewing is a common procedure for recruiters to assess applicant characteristics such as personality and competencies (Huffcutt, Conway, et al., 2001). Behavioral interviews are a best practice in interviewing. They often feature past-behavior questions, which are valid predictors of future job performance. Past-behavior questions invite participants to describe their behavior in past work-related situations (Campion, Palmer, and Campion, 1997; Janz, 1989). Applicants' responses to such questions should thus take the form of a narrative (i.e., storytelling) (Ralston, Kirkwood, and Burant, 2003). Indeed, storytelling is a good way to create a positive impression, providing an engaging and potentially more memorable response (Bangerter, Corvalan, and Cavin, 2014; Stevens and Kristof, 1995), thus increasing hireability ratings (Bangerter, Corvalan, and Cavin, 2014). Practical interview guides often advise applicants to use the STAR mnemonic to organize their stories: Start with a brief description of the situation (S), explain the task and actions undertaken (TA), and describe results obtained (R) (Kessler, 2006). For example, regarding the past behavior question:

> Can you tell me about a situation where one of your employees didn't complete a task on time? How did you react to this problem?

a storytelling response from our corpus, translated from French, is (see Table 7.1 for abbreviations and definitions):

> *then uh yes it did happen uh during a project to be handed in uh for uh the courses* (SS)/ *and um what happened is that I uh discussed with uh the various uh collaborators who hadn't handed in their work on time* (STA)/ *and so we sent an e-mail uh* [laughs] *to the professor* (STA)/ *to extend the deadline* (STA)/ *to find a way* (STA)/ *basically we used uh I tried to communicate the best I could with them* (STA)/ *and tell them what was wrong* (STA)/ *and what solutions we could perhaps uh put in place* [laughs] (STA)

This example constitutes a story because it describes a unique episode (*it did happen uh during a project*) in the past, one marker of this being the use of the past tense. In this example, the response has been segmented into utterances and each utterance has been labeled as being about the Situation (SS), Task and Actions (STA), or Results (SR). Besides the words constituting each utterance, the response transcript also shows evidence of filled pauses ("uh") and paraverbal behavior ("laughs").

Applicants do not always spontaneously provide storytelling responses when asked past-behavior questions, and often provide sub-optimal responses, such as general descriptions of generic situations (pseudo-stories, P), often using the present tense without any individualizing information (e.g., specific time or location, specific actions or outcomes). Another

Table 7.1: Overview of Utterance Types

Utterance Type	Abbreviation	Definition
Story	S	Description of a specific past event
Situation	SS	Description of a specific situation
Task-Action	STA	Description of specific tasks and actions
Result	SR	Description of specific results obtained
		Generic description of events
Pseudo-Story	P	
Situation	PS	Generic description of situations
Task-Action	PTA	Generic description of tasks and actions
Result	PR	Generic description of results usually obtained
Value/Opinion	VO	Assertions about one's values and opinions
Self-description	SD	Assertions about one's traits
Justification	JS	Justifications of one's actions
Other	OR	Other types of utterances

type of sub-optimal response consists of decontextualized assertions about oneself, including values/opinions (VO), self-descriptions (SD), or justifications (JS); examples are given in Appendix D.1.

This does not only illustrate the difficulties applicants face in producing stories in response to past-behavior questions, but also highlights the variability in the types of responses given. While storytelling allows applicants to present themselves positively by attributing desirable characteristics to themselves in their telling (Ralston, Kirkwood, and Burant, 2003; Silvester, 1997), it also involves descriptions of concrete past behaviors that help recruiters evaluate them. In contrast, pseudo-stories offer descriptions of generic typical behaviors and decontextualized self-descriptions fail to provide any evidence of what applicant has done in the past. This variability in response content may constitute an important factor affecting the validity of the interview procedure (Huffcutt and Murphy, 2023). Coaching applicants to tell stories can help reduce the variability in response type, thus limiting variance in evaluation that is unrelated to the competencies targeted by the interview questions.

Recently, in-person job interviews are being increasingly complemented by asynchronous job interviews (AVIs). Applicants connect to a platform and audio-/video-record their responses to questions which are displayed on-screen or asked by an avatar, without much dialogue or interactions between the interviewer and applicant as in FTF interviews. Recruiters, or, more recently, algorithms, evaluate applicants' responses based on the video-recording (Lukacik, Bourdage, and Roulin, 2022). Since AVIs constitute different interactional situations, applicants' responses to past behavior questions are likely to differ. Compared to FTF interviews, applicants' storytelling responses in AVIs feature similar proportions of STAR narrative elements (Germanier, Bangerter, Renier, et al., 2023). However, applicants talk more about their values and opinions in AVI settings (Germanier, Bangerter, Orji, et al., 2025). Such expressions

of personal values may not constitute relevant responses to past-behavior questions, resulting in lower chances of being hired (Orji et al., 2024). To help with this, video-based training has been developed, by explaining for instance what constitutes good behaviors leading to higher performance, including storytelling (Roulin, Pham, and Bourdage, 2023). An alternative training approach for effective storytelling could involve AI-powered systems that automatically provide tailored feedback to applicants based on recordings of their responses.

Along with the advent of AVIs, significant progress has been made in machine learning (ML). Machine learning has been applied in AVIs to predict applicants' personality traits or hiring recommendations based on automatically extracted verbal, paraverbal and visual features as well as human annotations (e.g., Naim et al. (2015), Nguyen, Frauendorfer, et al. (2014), Chen, Zhao, et al. (2017), Hickman, Bosch, et al. (2022), Rupasinghe et al. (2016), Holtrop et al. (2022), and Liem et al. (2018)). In the context of job interview training, ML applications can provide applicants with automatic feedback to improve their interview performance. One study developed an application where interviewees interact with a conversation avatar coach to get feedback (Hoque et al., 2013). Another study implemented automatic feedback on interviewees' behavior into serious games for job interview training (Gebhard et al., 2019). Langer et al. (2016) also investigated how real-time feedback on nonverbal behaviors in avatar training job interviews improves interviewee performance.

The above systems deliver feedback on nonverbal or paraverbal aspects of applicants' responses. One system delivers feedback on "weak language" (i.e., filler words) (Hoque et al., 2013). Currently, no systems we are aware of are capable of delivering feedback tailored to the substantive content of participants' responses. This is an important gap because semantic content is one of the most important factors affecting interview performance, being more important than nonverbal or paraverbal behavior (Rasmussen, 1984; Riggio and Throckmorton, 1988). Further, content is particularly important in evaluating or improving responses to past-behavior questions. The analysis of semantic content from an interview recording can be broadly considered as a specific spoken language understanding (SLU) task, which is the subject of this thesis. In a first step in this direction, Bangerter, Mayor, et al. (2023) used ML to identify storytelling in applicants' responses in FTF interviews. Based on interview transcripts, they automatically extracted semantic features (i.e., LIWC; Pennebaker et al. (2015)), word counts, and TF-IDF. Using ML algorithms (e.g., random forest), they predicted whether a response consisted of a story or not considering the amount of different STAR narrative elements in responses. Their findings suggested that it is feasible to identify a particular utterance type, storytelling, in applicants' responses using ML. A limitation of that study is the reliance on transcribed speech, which restricts the applicability of the approach to a real training platform setting. Indeed, it is not feasible to manually transcribe the audio recordings prior to ML analyses in a scalable application.

More sophisticated deep learning (DL) techniques may enable progress. These techniques have also been used in AVI settings, e.g., to predict personality using convolutional neural networks (Suen, Hung, and Lin, 2019), or hirability using hierarchical gated recurrent units

(Hemamou, Felhi, Vandenbussche, et al., 2019), though still using handcrafted features, with attention weights used to identify critical timespans (Hemamou, Felhi, Martin, et al., 2019). Particularly, the high cost of collecting realistic data (as in the case of interview responses) becomes a restriction of the dataset scale. Hence recent large-scale pretrained models on the text and speech domain, as introduced in Chapter 2, can be relevant. Attempts have been made to incorporate pretrained models into interview analysis, such as using them to process interview transcripts for predicting hirability (Rahman et al., 2021). However, there is still much to explore about the potential of end-to-end models with pretrained models in different modalities (including both language and speech) in analyses of applicants' response content.

Building on these studies and the recent progress in ML, this study aims to develop neural models based on PTLMs, capable of automatically identifying storytelling as well as other, sub-optimal types of response from audio recordings, using models that can easily be adapted to various languages. To summarize, our contributions are (1) building a realistic machine learning dataset on interview responses and storytelling; (2) analyzing the content of participants' responses; (3) evaluating performance of neural models under the pretraining+fine-tuning paradigm; and (4) working directly from audio recordings in either cascaded or end-to-end manner, which brings us a step closer to practical implementation in real application scenarios. Our findings pave the way for developing models to implement in AI-powered training platforms or in AVIs that provide automatic feedback on applicants' responses to past-behavior questions from the audio recordings.

7.2 This Study

This study used DL models with the goal of predicting storytelling components (S/TA/R) and other utterance types such as pseudo-story components (S/TA/R), values/opinions, self-descriptions, and justifications from audio recordings of participants answering past-behavior questions in mock job interviews. Participants' responses were labeled by human annotators. These labels were used as the ground truth for training models.

This study goes beyond previous work (e.g., Bangerter, Mayor, et al. (2023)) in several technical respects. First, we leverage large-scale pretrained models, which is essential in the current situation where the training set of participants' responses is relatively scarce and linguistically complex. Specifically, we used the multilingual version of the recent pretrained textual model RoBERTa (Liu, Ott, et al., 2019) and speech model Wav2Vec2 (Baevski et al., 2020), in combination with the state-of-the-art pretrained speech recognizer Whisper (Radford, Kim, et al., 2023). Second, we predicted a large set of labels (i.e., 10) that constitute a comprehensive and fine-grained array of applicants' potential responses to past-behavior questions. Third, we implemented a range of different modeling techniques to identify the impact of various factors.

Using these pretrained DL models, our modeling followed a three-stage development. In the first stage, we conducted *baseline assessments*. We compared the performance of the models

under the evaluation scenarios using human transcripts (i.e., *transcription-based* scenarios), or using automatic speech recognition (ASR) models that generate transcripts to mirror practical cases when only the audio is available (i.e., *audio-based* scenarios). In addition to the original labeling scheme with 10 labels, we also assessed two higher-level labeling schemes with fewer categories. We did this because prediction of labels that were less frequent (See Figure 7.1) or that were difficult to discriminate may be challenging and less accurate. This is even more problematic with limited context, i.e., when there is no information from the surrounding utterances from the same interview to help the model correctly classify a targeted utterance. More importantly, our objective is to detect storytelling, and it could be useful to consider simplified schemes that are more focused on this goal.

The second stage consisted of *information enrichment* in training of audio-based models. We considered the effectiveness of the model trained upon the human transcripts only, versus models trained with additional information from the audio. Such techniques include using transcripts produced by ASR during training, as well as directly introducing the audio as part of the model input. We explored these scenarios because human transcripts, although more accurate and generally used for model training, are not available in real cases based on interview recordings where we can only access ASR transcripts. Furthermore, paraverbal cues (e.g., intonation, prosody) might be also informative for this semantic-centric classification, but are not well represented in transcripts. For example, narratives (stories and pseudo-stories) involve (specific or generic) descriptions of past experiences that often lead to narrative transport (Green and Appel, 2024), which can result in more dynamic vocal behaviors, such as pitch variations.

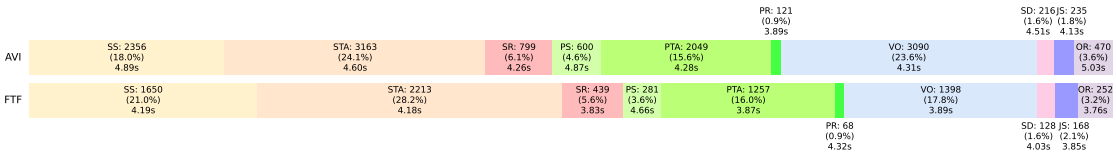
The third stage consisted of *context enlargement*. Given the significant variation in utterance length and the prevalence of relatively short utterances, it is often challenging to accurately determine the utterance type. For instance, an utterance describing an action may be classified as STA if the exact time of the event is mentioned in the previous utterances but as PTA otherwise. Therefore, incorporating more information from the surrounding utterances of the target utterance into the models can enhance their performance. Similar benefits have been also observed in other tasks like speech translation (Zhang, Titov, et al., 2021) and ASR (Chen, Kano, et al., 2024). Hence we created two different techniques to incorporate more context, namely coalescence and expansion. Coalescence involves merging adjacent utterances of the same type, while expansion directly enlarges the context visible to the model.

7.3 Dataset Preparation

7.3.1 Data Collection

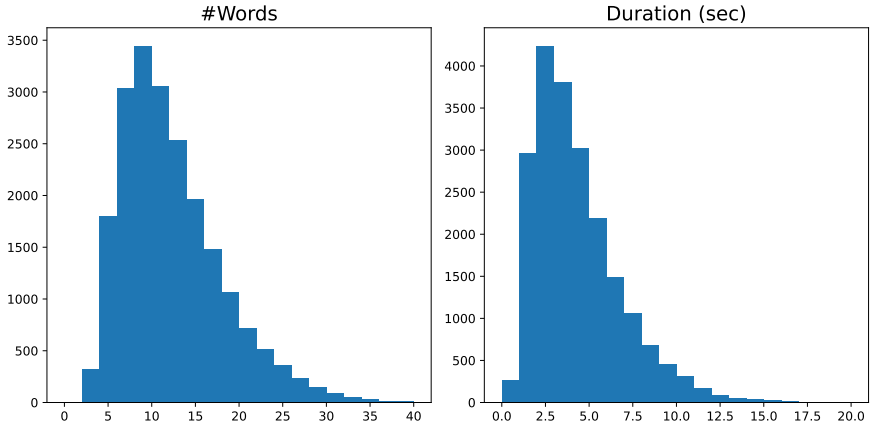
We recruited 254 French-speaking undergraduates into our study as participants, and engaged them into a two-stage lab experiment. In the first stage, they performed a work sample (i.e., two role plays) adapted from Richter et al. (2016). About one week later, participants returned

Figure 7.1: Number of Utterances, Percentage, and Average Length (sec) of Different Labels for Utterances from FTF Interviews and AVIs.



Note. The number, percentage, and average length (in seconds) of utterances for each type are indicated in the figure. The proportion differs between the two interview conditions, while for both the categories are highly imbalanced and labels like PR and SD account for little proportion of the data.

Figure 7.2: Histogram of the Word Counts and Utterance Duration in Our Dataset.



Note. The distribution leans to the left, indicating a large number of short utterances.

for a mock job interview for a managerial job position, randomly assigned to either FTF or AVI conditions. During the interview, all participants were asked the same four questions, including one self-presentation question and three past-behavior questions on managerial skills, including one targeting the role play of the previous week. Audio recordings of interview are collected and used in this study. More details of the data collection process are available in Appendix D.2.

7.3.2 Annotation and Statistics

All participants' responses to the past-behavior questions were transcribed manually, including disfluencies (*uh, um, hmm*), hesitations, repetitions, and word truncation. Their responses were then segmented into shorter units, here utterances corresponding to a subject, verb, and object structure (Bangerter, Corvalan, and Cavin, 2014).

We compared utterances and interview lengths between FTF interviews and AVIs using the Kolmogorov-Smirnov test, as the data were not normally distributed. Regarding the utterance lengths, there was a small but significant difference between FTF interviews and AVIs (Kolmogorov-Smirnov $D = 0.022$, $p = .016$, w.r.t. word counts): Each utterance contained

on average M=11.68 words (SD=5.67) or M=4.06 sec (SD=2.59) of speech for FTF interviews, and M=11.96 words (SD=6.02) or M=4.52 sec (SD=2.69) for AVIs. Thus, utterances were often quite short (see Figure 7.2). Also, there was a significant difference between the two interview conditions for the total length of an interview (Kolmogorov-Smirnov $D = 0.235$, $p = .002$, w.r.t. word counts): For each FTF interview, there were M=1250.25 words (SD=423.53) or M=7.27 min (SD=2.31) min of non-silent audio on average, while for AVIs the number was M=1511.26 words (SD=644.69) or M=9.69 min (SD=3.82).

Each utterance contained in participants' responses to past-behavior questions was annotated using the categories proposed by Bangerter, Corvalan, and Cavin (2014), as provided in Table 7.1; more details are provided in Appendix D.2. Two trained experts in personnel selection were involved in this annotation task. Reliability was established by double-annotating 30 transcripts (Fleiss' Kappa = .99, $p < .01$). The proportion of utterances for each category is demonstrated in Figure 7.1.

7.3.3 Alignment

Interview audio files collected in this study are first transcribed, and the type of each utterance is then annotated from the transcript. To directly analyze the interview speech, it is necessary to associate the annotated utterance type with the audiorecording of the corresponding segment, so that we can train DL models to predict the label given audio only. To achieve that, we use the standard technique known as *forced alignment* in speech processing, which aligns the transcript with the audio by identifying the exact start and end timestamps within the audio for each word in the transcript. Then it is straightforward to identify the audio segment or time interval in the recording corresponding to each utterance, and the label can be assigned to each segment accordingly.

Typically, forced alignment is carried out using acoustic models, which are also a key component of ASR systems; multiple out-of-the-box toolkits are available. However, in our data the recordings are long-form French dialogues. There are many filler words, variations in the recording quality, long silences, and sometimes overlapping speech between speakers; also the target utterances are often rather short. Long silences due to, e.g., participants thinking about their response, are particularly disturbing as they interfere with acoustic modeling. But identifying them, known as voice activity detection (VAD), is not trivial, and simple energy-based VAD (e.g., identifying a time interval as silence when the amplitude is small enough compared to other parts of the audio) fails due to variations in the dialogue and recording environment, as well as bursts of environmental noise (e.g., knocking at the table). We tried several well-known out-of-the-box toolkits for forced alignment, including Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) and WhisperX (Bain et al., 2023), but both of them failed to address these unique challenges and reach satisfactory results. Therefore, we had to build a specialized forced alignment pipeline using a combination of multiple pretrained speech models and heuristics.

We mainly leveraged the forced alignment toolkit in Massive Multilingual Speech (MMS) (Pratap et al., 2024), an ASR system focused on multilingual scenarios. The toolkit is based on a multilingual Wav2Vec2-based encoder-only model with Connectionist Temporal Classification objective (Graves et al., 2006). Such standard acoustic models allowed us to obtain the exact timestamp of phonemes, and the toolkit could then assign timestamps to the words and achieve forced alignments. In particular, the toolkit is able to identify silent frames, which allowed us to jointly perform VAD and forced alignment precisely. Nevertheless, long silence was generally harmful to the model performance. Therefore, we pre-processed the audio with the toolkit, and removed all the long silences (>4 sec), with 1.6 sec safe margin to allow for some error. All the following processes are based on this new version of recordings with silence removed. Using this toolkit we then obtained the final forced alignment and segmented the recordings into audio utterances. Additional margins on the left and right of the segment were added if the pause between words permitted. We found the results generally satisfying.

7.3.4 ASR Transcription

We then produced the ASR transcripts of our recordings for the training and evaluation procedures. Since the ASR system that produces the transcripts is an essential part in our pipeline, we need an ASR system that accurately recognizes our French speech data and avoids speaker biases (Hickman, Langer, et al., 2024). We thus chose OpenAI Whisper (large-v2) (Radford, Kim, et al., 2023), which is a sequence-to-sequence ASR model pretrained on an enormous amount of online available speech data from diverse speakers, styles (e.g., accent), and languages. Although it is not suitable for forced alignment, it stands among recent ASR systems with the lowest error rates (Hickman, Langer, et al., 2024). There were discrepancies in the text normalization scheme [1] between Whisper transcripts and ours, which contributed to extra "errors". By running Whisper on the audio segments, we nevertheless obtained a total character error rate of 16.0% between the ASR and human transcripts. This low error rate (close to the 13.9% error rate on the widely used yet challenging Common Voice benchmark, as reported by Radford, Kim, et al. (2023)) further verifies the forced alignment results, since the spoken content of each segmented clip will match the transcript only when the segmentation (decided by the starting and ending timestamp of each utterance) is accurate.

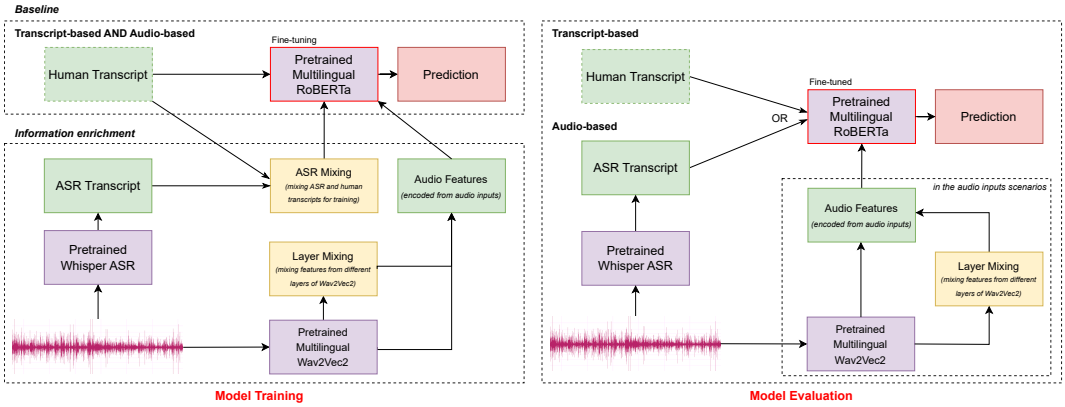
7.4 Modeling

7.4.1 Model Building

We formulated this interview analysis as a multiclass classification task under supervised ML, based on a pipeline combining multiple pretrained DL models. We followed standard ML

[1] The same speech may be transcribed in multiple slightly different ways, due to, e.g. whether numbers are transcribed in words or digits, whether filler words are transcribed, how is the text punctuated, etc. Text normalization refers to the process of specifying a normalized form of ASR transcription, which may affect downstream text processing.

Figure 7.3: Training and Evaluation Pipeline for Predicting Utterance Types from Interview Recordings.



Note. In the transcript-based modeling, the ground-truth human transcripts are used as the sole input for fine-tuning as well as evaluating the pretrained multilingual RoBERTa model; while in the audio-based modeling, human transcripts are still used in training, but transcription by the ASR system, Whisper, becomes the basis of evaluation. Furthermore, in the information enrichment with ASR Mixing, ASR transcripts are also used during training. In addition, the audio inputs may be directly processed by the neural model via Wav2Vec2 as an audio encoder.

practice to partition our data randomly into 8:1:1 training, validation, and testing subsets based on interviews and their conditions. Utterances from the same interview were grouped in the same subset, and FTF interviews and AVIs were assigned to the subsets in a balanced ratio. For the classification of utterance types, we focused on predicting the label of each utterance individually. Therefore, the task was simplified as standard supervised learning by fine-tuning pretrained models to predict the labels of certain shorter text or audio utterances. This task was developed in three stages, *baseline*, *information enrichment* and *context enlargement*. An overview of the pipeline is illustrated in Figure 7.3.

Baseline Assessments

We distinguish two types of modeling, depending on the availability of ground truth transcripts. Human transcript-based models are trained and evaluated using human transcripts, while audio-based models are trained with human transcripts but evaluated using audio data converted to ASR transcripts. We then created three different labeling schemes: The first scheme involves *fine-grained labeling*, which distinguishes the ten human-annotated categories (i.e., SS, STA, SR, PS, PTA, PR, VO, JS, SD, and OR). The second scheme, *combining non-narratives*, merges VO, JS, SD, and OR labels into a generalized OR category, reducing the total number of categories into seven. The third labeling scheme, *coarse-grained labeling*, merges the SS/STA/SR into a single S category and PS/PTA/PR into PS, resulting in the most simplified three-way classification.

Information Enrichment

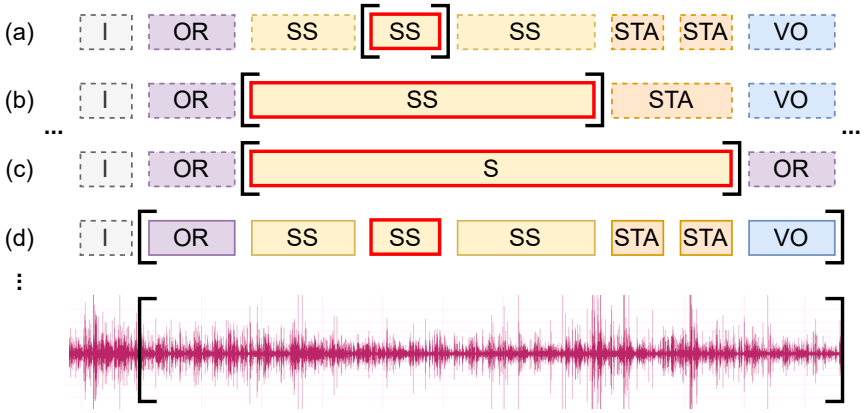
We used two techniques to enrich the training of audio-based models with audio information. The first technique is ASR mixing, which involves mixing samples of human transcripts and ASR transcripts during training. Although Whisper ASR (large-v2) is fairly accurate, the automatic system can produce errors compared to the ground truth human transcripts, which will mislead the model. Even if we do not count the errors, there are still differences in transcribing style (e.g. regarding punctuation). These discrepancies lead to *domain mismatch* of the model inputs between training and evaluation. Such mismatch may reduce performance, but can be alleviated by training the model on both the source (human) and the target (ASR) transcripts to better adjust the model to tolerate ASR transcription inputs. The second technique used audio inputs, directly injecting the corresponding audio segment extracted by forced alignment into the model, creating a *joint speech-language model*, so that paraverbal or nonverbal cues missing in the transcripts can be accessed by the model. In this case, we leveraged the pretrained Wav2Vec2 as an audio encoder to extract audio features which are then concatenated with the text inputs as the input sequence to the text model. Whilst the use of paraverbal cues is naturally related to the performance of storytelling, note that there is prior research evidencing that linguistic information is critical in predicting applicants' personality or interview outcomes (Koutsoumpis et al., 2024). Indeed, in some cases, combining multimodal predictors (linguistic, paralinguistic, nonverbal) can even increase bias without improving performance (Booth et al., 2021). The use of audio inputs hence amounts to examine whether paralinguistic information improves model performance in the storytelling classification task.

Context Enlargement

To enlarge utterance context, we tested two techniques: coalescence and expansion. Coalescence involves merging adjacent utterances of the same type. Since utterance segmentation is often relatively fine-grained, we can merge adjacent short utterances (<2 sec) in each interview based on the forced alignment results. To do this, we locate the shortest utterance at each instance using a priority queue and then coalesce it with the neighboring utterance (with <1-sec gap) of the same type on the left or right side. This allows for the absorption of a short utterance into a longer one or the merger of multiple short utterances into a longer one, resulting in more information being available within each new "utterance" (see Figure 7.4, b-c). Under the original fine-grained labeling scheme, there are limited adjacent utterances with the same label. Hence, we limit this approach to the coarse-grained labeling case. In this way, the mean length of the utterances can be increased from M = 11.86 words (SD = 5.89) or M = 4.35 sec (SD = 2.66) to M = 35.50 words (SD = 18.70) or M = 12.78 sec (SD = 6.28). Nevertheless, it should be noted that it requires pre-known information about the type of each utterance, which would not be available in real situations.

Expansion is an alternative way to feed the context directly into the model. We provide the model with a longer segment of the interview comprising the target utterance and a short part of context before and after it. We use a pair of special tokens <s> (Begin-of-Sequence)

Figure 7.4: Examples of Context Enlargement Techniques with Different Labelling Schemes.



Note. (a) Only the target utterance (indicated by the utterance with the red border, with label Story: Situation) is provided; (b) Coalescence: adjacent utterances of identical label are coalesced and classified together; (c) In addition to (b), a more coarse-grained labeling with only three different types (Story/Pseudo-Story/Other Responses) is used, hence the SS and STA utterances can be coalesced; (d) Expansion + Audio inputs: longer context around the target utterance with various labels is provided; in addition, the corresponding audio segment is also provided. I indicates an utterance spoken by the interviewer, which may serve as a boundary between questions.

and </s> (End-of-Sequence) to enclose the target utterance within the input to specify the actual target to be classified. With this method, a broader context containing various types of utterances can be included (see Figure 7.4, d). Specifically, for each target utterance, we included the adjacent utterance before or after the target utterance, starting with the shorter one, so as to cover a context of 15~25 sec with at least 5 sec on each side. FTF interviews involved dialogues between participants and the recruiter, thus also including the interviewer's speech. In this case, for consecutive utterances from one speaker, the exact speaker was indicated to the model (e.g., "CA" for participants). Also, multiple questions were included in FTF interviews and AVIs. When the target utterance is part of the answer to a specific question, the part regarding other questions might not be relevant and should not be included in the context. Therefore, we assumed that in a FTF interview a new question starts whenever the interviewer makes a long statement introducing the next question. As for AVIs, we knew exactly when a new question started as it was on the screen. In both cases, the iterative expansion on either the left or the right side stops when the presumed question boundary is touched, so as to avoid including responses to other questions. In this way, prediction on each utterance may leverage an expanded context of various types of utterances, with $M = 19.10$ sec ($SD = 2.70$) or $M = 58.83$ words ($SD = 11.24$). Furthermore, the expansion technique does not depend on manual transcription once trained, since the models are able to transcribe audio responses on their own and make predictions about each specific utterance, leveraging the enlarged context determined on their own. As such, this approach allows the model to classify any interview segment with minimal dependence on pre-known information about how the utterances are segmented and labeled.

7.4.2 Model Implementation

We use current pretrained language models in our system. For handling text inputs, we used XLM-R (Conneau and Lample, 2019), the multilingual version of RoBERTa (Liu, Ott, et al., 2019), trained upon data in 100 languages, French included. The model is fine-tuned using Adam optimizer (Kingma and Ba, 2017), the widely used efficient accelerated stochastic gradient descent approach based on moment estimation. The learning rate is set to 2E-5 by grid search, and follows a schedule of linear warm-up in 500 steps followed by inverse-square learning rate decay. To accommodate the model in a single 24GB RTX3090 GPUs while avoiding inefficient padding, we use dynamic batching with around 44 samples per batch in average. The models are trained for at most 7500 steps, which can already ensure good convergence in our experiment. On a single RTX3090, a single training run will take around 1 hour for a model using only text inputs, and 2.5 hours for a model with audio inputs. We then chose the checkpoint with best validation accuracy, and report the validation and test results.

In the case of audio-based modeling, the large-v2 version of Whisper is used as the ASR system to transcribe the recordings for its reported superior performance. Furthermore, instead of relying on ASR transcripts (i.e., indirectly using the audios), we may also allow the model to directly process the audio recordings. To handle audio inputs, we chose the Wav2Vec2 (Baevski et al., 2020) as the basis of our audio encoding module.In particular, we leveraged its multilingual version XLSR (Conneau, Baevski, et al., 2021) to handle French speech. We observed that the training failed to converge when we tried to fine-tune both the audio encoder and the stacked text model (i.e., RoBERTa). Therefore, we used the audio encoder as a feature extractor and to additionally concatenate the audio features after the embeddings of the textual inputs as an audio prompt for the text model. The speech models (i.e., Wav2Vec2 and Whisper) are frozen during training, and only the RoBERTa-based text model is updated. We also found out that it is difficult to train the models initialized from the original pretrained RoBERTa. Instead, we fine-tune the model initialized from the text-only response classification built above. Following common BERT-like practice, we leveraged the final layer features of the audio encoder; nevertheless, it has been found that higher layer features represent more semantic cues but lose paralinguistic ones from lower layers, which motivates the technique of Layer Mixing (Pepino, Riera, and Ferrer, 2021): Instead of using only the final layer, we combine the features from different layers by taking a weighted average. Initialized to be equal to each other, the weight for each layer is a learnable parameter further optimized during the fine-tuning. This allows the features from the most relevant layer to be automatically extracted, which improves the performance on speech classification tasks.

In sum, we build two types of models. The first are transcript-based models, which are trained and evaluated using only human transcripts. They serve as a baseline for performance comparison. The second type are the audio-based models, which are trained using human transcripts (and ASR transcripts in some scenarios) but evaluated only with ASR transcripts (see Table 7.2). As such, audio-based models, once trained, do not require manual transcription to identify the response type in new audio responses.

Table 7.2: Transcript- versus Audio-Based Models inputs

	Training	**Validation and Test**
Transcript-based	human-transcript	human transcript
Audio-based	human-transcript	ASR transcript
+ ASR Mixing	human or ASR transcript	ASR transcript
+ Audio inputs	human-transcript + audio	ASR transcript + audio
+ ASR Mixing	(human or ASR transcript) + audio	ASR transcript + audio
+ Layer mixing	human-transcript + audio	ASR transcript + audio

7.5 Results

We present our results on the validation and test set after training the models for various techniques. We first report the accuracy of the model.However, as shown in Figure 7.1, there is a class imbalance issue in our task. A naive model that simply outputs the most frequent category (i.e. STA) for any input can already achieve nearly 30% accuracy. To better reflect the model performance on predicting less frequent categories, we also report the macro-F1 score. In this way, the performance of the model on all categories is better represented. The evaluation results are reported in Table 7.3.

7.5.1 Baseline Assessments

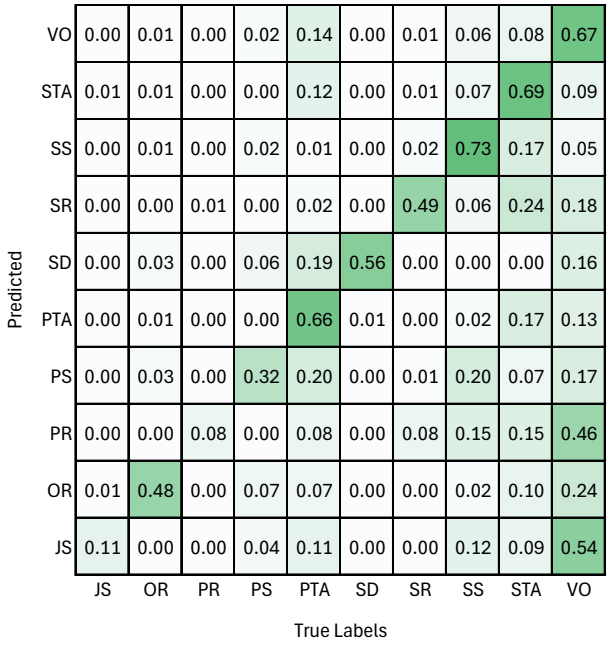
Results for human transcripts showed satisfying performance in the test set on all three labeling schemes, including the most challenging 10-way fine-grained labeling with 63.65% accuracy and 51.41% macro-F1. The performance was further increased when combining non-narratives (accuracy difference: +2.32%; macro-F1 difference: +5.5%) and even more with the coarse-grained labeling scheme (accuracy difference: +8.69%; macro-F1 difference: +17.82%). The confusion matrix in the human transcript-based model with the fine-grained labeling scenario (see Figure 7.5) confirmed that the model had trouble distinguishing several categories (e.g., VO vs JS, VO vs OR, STA vs SR). The comparison between human transcript-based and audio-based models showed that the performance of audio-based models was consistently lower across all labeling schemes (accuracy difference range: from -1.73% to -1.86%; macro-F1 difference range: from -0.39% to -1.34%). However, the gap is narrowed as the performance of audio-based models improved considerably from fine-grained labeling to combining non-narratives (accuracy difference: +2.19%; macro-F1 difference: +4.71%) and further to coarse-grained labeling (accuracy difference: +9.54%; macro-F1 difference: +17.98%).

Table 7.3: Validation and Test Accuracy/F1 Scores For Baseline Assessment, Information Enrichment and Context Enlargement.

	val. acc.	*val. F1.*	*test acc.*	*test F1.*
HUMAN TRANSCRIPT-BASED				
Fine-grained Labeling	59.83	50.39	63.65	51.41
Combining Non-narratives	62.25	52.61	65.97	56.91
Coarse-grained Labeling	70.16	65.23	72.34	69.23
AUDIO-BASED				
Fine-grained Labeling	56.77	50.24	61.92	50.86
+ ASR Mixing	**57.35**	**51.28**	**62.25**	**53.52** ↑
+ Audio Inputs	**57.06**	**50.43**	61.79	50.76
+ ASR Mixing	57.30	49.40	**62.39**	52.81
+ Layer Mixing	56.87	**51.15**	**61.92**	**51.24**
Combining Non-narratives	59.29	50.25	64.11	55.57
+ ASR Mixing	**59.49**	**51.56**	**64.90**	**55.59**
+ Audio Inputs	**59.44**	**52.07**	**64.76**	**55.69**
+ ASR Mixing	59.24	**52.40** ↑	64.53	55.49
+ Layer Mixing	59.34	51.85	**64.85**	55.61
Coarse-grained Labeling	68.12	63.88	71.46	68.84
+ ASR Mixing	67.25	62.82	71.32	68.70
+ Audio Inputs	68.12	**64.17**	**74.09** ↑	**72.16** ↑
+ ASR Mixing	66.47	62.28	72.38	70.04
+ Layer Mixing	68.12	63.88	71.87	69.32
W/ COALESCENCE				
Coarse-grained Labeling	**69.19**	**63.97**	**72.56**	**69.96**
+ ASR Mixing	**70.27** ↑	**65.21**	**71.87**	**69.58**
+ Audio Inputs	**69.05**	63.20	73.12	70.85 ↑
+ ASR Mixing	**71.35** ↑	**65.63**	**72.56**	**70.28**
W/ EXPANSION				
Fine-grained Labeling	**65.36** ↑	**51.35**	**68.62** ↑	**53.16** ↑
+ ASR Mixing	**65.26** ↑	50.81	**67.60** ↑	**53.71** ↑
+ Audio Inputs	**65.31** ↑	**53.40** ↑	**69.35** ↑	**54.12** ↑
+ ASR Mixing	**65.60** ↑	**50.91**	**67.21** ↑	**53.16** ↑
Combining Non-narratives	**66.91** ↑	**55.58** ↑	**70.57** ↑	54.72
+ ASR Mixing	**66.57** ↑	**54.56** ↑	**71.46** ↑	**58.92** ↑
+ Audio Inputs	**66.62** ↑	**55.30** ↑	**70.14** ↑	54.62
+ ASR Mixing	**66.23** ↑	**54.32** ↑	**71.12** ↑	**59.42** ↑
Coarse-grained Labeling	**72.97** ↑	**68.74** ↑	**77.22** ↑	**74.77** ↑
+ ASR Mixing	**73.80** ↑	**70.35** ↑	**77.45** ↑	**75.39** ↑
+ Audio Inputs	**72.63** ↑	**68.84** ↑	**76.56** ↑	**74.39** ↑
+ ASR Mixing	**74.14** ↑	**70.42** ↑	**77.67** ↑	**75.56** ↑

Note. Results are in %. In each scenario and labeling scheme, the highest results across different information enrichment techniques are underlined. We also **bold** the results for a technique when it is better compared with the corresponding baselines: In the Audio-based experiments, the results for a specific technique is compared with the one without it (e.g.,*+Audio Inputs* + ASR Mixing is compared with *+Audio Inputs*). In the context enlargement (coalescence/expansion) experiments, one is compared with the Audio-based results with the same enrichment technique but without context enlargement. We further put an ↑ for models with >2% improvements compared to the baseline under the scenario and labeling scheme without any extra enrichment or enlargement. Abbreviations: val. = validation, acc. = accuracy, F1. = macro-F1.

Figure 7.5: Confusion Matrix Predicted vs True Labels for Human Transcript-Based Fine-Grained Labeling.



Note. Distinguishing labels like VO and JS is challenging.

Figure 7.6: Normalized Attention Weights in Fine-Grained + Audio Input Model.

	[CLS]	Text	Audio
[CLS]	10.90%	87.72%	1.38%
Text	4.64%	89.18%	6.17%
Audio	0.50%	48.32%	51.18%

Note. Averaged over layers and heads from different part of the inputs on each row to other parts in each column, with [CLS] the classifier token that is used to produce the final prediction. The [CLS] token and the text part puts little attention weight on the audio part, and most attention weights for the [CLS] are placed on the text part of the inputs.

7.5.2 Information Enrichment

Applying ASR mixing in audio-based models during training led to slight improvements in the more challenging fine-grained labeling schemes compared to the audio-based fine-grained labeling model trained on human transcripts only (accuracy difference: +0.33%; macro-F1 difference: +2.66%). But it was less effective with combining the non-narrative (accuracy difference: +0.79%; macro-F1 difference: +0.02%) and coarse-grained labeling (accuracy difference: -0.14%; macro-F1 difference: -0.14%) cases compared to their respective audio-based models without ASR mixing. Enriching the model with audio inputs in the training phase improved performance in the more coarse-grained labeling case compared to the coarse-grained labeling audio-based model without audio inputs (accuracy difference: +2.63%; macro-F1 difference: +3.32%). However, it did not improve performance in the fine-grained labeling (accuracy difference: -0.13%; macro-F1 difference: -0.10%) and combining

non-narratives labeling (accuracy difference: +0.65%; macro-F1 difference: +0.12%) when compared to the audio-based models without direct audio inputs under their respective labeling scheme. Since the prediction is made upon the features corresponding to the [CLS] token which aggregates the information from the whole inputs using an attention mechanism with an attention weight assigned to each part of the input, we were able to determine the importance of each part of the input for making the prediction by the weight. As given in Figure 7.6, the normalized attention weights extracted from this model confirmed that the [CLS] token and the transcript part (human or ASR) placed little attention weight on the audio part (see Figure 7.4). Instead, most attention weights for the classifier token [CLS] were placed on the text part of the input. Moreover, the attention weight on the audio part gradually decreased during the training process, indicating that, for making the prediction, the model learned to put more focus on the texts and less focus on the audio. Therefore, the information learned from the audio by the model was of limited relevance for making the prediction, and a pipelined analysis via ASR transcripts can already achieve performance very close to or even better than the model directly processing audios.

We continued to investigate whether combining the audio inputs with other enrichment approaches (i.e., ASR mixing or layer mixing) leads to better performances. Combining the audio inputs with ASR mixing in the training phase seldom improved the models' performance, regardless of the labeling schemes. When compared to their respective labeling schemes audio-based models with only audio inputs enrichment, the accuracy differences range from -1.71% to 0.60% and macro-F1 difference range from -2.12% to +2.05%. Performance when combining the audio inputs with layer mixing in the training phase never deviated much from the audio-based models with audio inputs under their respective labeling schemes but without layer mixing (accuracy difference range: from -0.51% to 0.32%; macro-F1 difference range: from -1.57% to 0.12%). To summarize, enriching the model with ASR Mixing or Audio Inputs may improve the performance, depending on the labeling scheme, though combining them shows limited help to the model.

7.5.3 Context Enlargement

Coalescence improved performance in all scenarios of coarse-grained labeling compared to the original audio-based models trained without context enlargement (accuracy difference range: from +0.18% to +1.1%; macro-F1 difference range: from +0.24% to +1.12%), except for model with audio inputs solely (accuracy difference: -0.97%; macro-F1 difference: -1.31%). The performance of models trained with expansion consistently outperformed almost all the scenarios compared to the original audio-based models (accuracy difference range: from +2.47% to +7.56%; macro-F1 difference range: from -1.07% to +6.69%), with the most significant improvement in fine-grained labeling cases. Specifically, the models reached the best results when either audio inputs, ASR mixing, or both were leveraged, with as high as 69.35%, 71.46%, and 77.67% test accuracy and respective 54.12%, 58.92% and 75.56% macro-F1 on the three labeling schemes respectively.

7.6 Discussion

This chapter examined interview-response analysis as a realistic SLU task and showed that pretrained models can successfully identify storytelling-related utterance types from French interview recordings, despite limited labeled data. Beyond the feasibility of the task itself, the results provide several insights that are relevant to the broader questions of this thesis.

The results suggest that this particular classification task depends primarily on lexical-semantic information rather than on paraverbal cues. Directly incorporating audio into the model yielded only small and inconsistent improvements, indicating that the signal from the audio input is too weak, noisy, or misaligned to the task for the model compared to the text input. Notably, using direct audio inputs will considerably increase the model complexity and computational costs, which overshadows the necessity of this technique. This does not imply that audio information is generally irrelevant for interview analysis, but rather that, for utterance-type classification as defined here, the relevant signal is largely semantic and can already be captured through the transcript. After all, the storytelling analysis task can be annotated based merely on the text transcripts without paralinguistics. Nevertheless, there remains space for the paraverbal signal to be relevant for interview-related tasks, possibly when it is more aligned with the task at hand, and when models better at extracting the signal can be used.

Enlarging the available context leads to consistent performance improvements, especially with the expansion technique. Indeed, stories are not a property of isolated sentences but rather are intrinsically durative and structured as a sequence of clauses that mirrors the chronological order of past events (Labov and Waletzky, 1997; Bruner, 1991). In this respect, the interview-analysis task provides a concrete application-level validation of a broader claim developed throughout this thesis, namely that access to longer contextual spans rather than isolated short segments can be important to effective spoken language understanding in realistic settings.

At the same time, the gains from context enlargement also expose the practical limitation of current Transformer-based approaches. Enlarging the context improves prediction, but it also increases computational cost. This trade-off is particularly relevant here because interview recordings are long, utterances are short, and only part of the surrounding material is useful for each local decision. The present results therefore reinforce the motivation of the previous chapters: long-context spoken language understanding requires not only semantically strong pretrained models, but also mechanisms that use context efficiently. In that sense, this application chapter complements the methodological parts of the thesis by showing, in a realistic task, why long-context modeling is not merely an architectural concern but a practical requirement.

7.7 Conclusion

This chapter applied pretrained models to the analysis of French interview responses and showed that storytelling-related utterance types can be identified from speech recordings with promising accuracy, even in a low-resource setting. We also demonstrate the benefits of using expanded context as well as various information enrichment techniques, though direct audio cues bring limited help. The results support the value of foundation models in SLU, especially for the practical use of interview analysis.

8 Conclusions and Future Work

8.1 Conclusions

This thesis investigated how pretrained and generalist models can be effectively applied to spoken language understanding (SLU), especially in long-form and low-resource settings where both semantic complexity and computational constraints are central. Motivated by the analysis of long-form French interview recordings in the SteADI project, the work addressed a broad research question at the intersection of model capability, model efficiency, and real-world application: how to obtain strong semantic understanding of spoken input while remaining feasible in settings characterized by limited labeled data, long temporal context, and practical deployment constraints.

To answer this question, the thesis progressed through three complementary phases.

The first phase examined the semantic capability of pretrained and generalist models for SLU. It first studied the impact of introducing stronger semantic supervision to speech foundation models trained by low-level self-supervised acoustic objectives. We confirmed the effectiveness of speech translation as a means of improving the semantic utility of pretrained speech models, as demonstrated by improved performance on downstream SLU tasks. We also showed that this enhanced the cross-lingual transferability of the models. The same study also demonstrated that adaptation must be performed carefully: preserving pretrained semantic knowledge during fine-tuning, especially through Bayesian regularization, proved beneficial for downstream tasks compared to allowing the model to freely overwrite its pretrained knowledge. We then evaluated the semantic understanding capabilities of large language models (LLMs) in zero- and few-shot prompting settings. We found that LLMs can achieve strong performance on intent detection as part of a cascaded SLU system with a few or even zero examples. However, their performance degraded when handling tasks with complex definitions, or when there are ASR errors. To conclude, this phase indicates that pretrained generalist models have semantic capabilities to handle SLU tasks, while techniques like end-to-end modeling and proper supervsion, adaptation, and prompting techniques remain important to fully leverage their potential.

The second phase of the thesis focused on model efficiency, especially in the context of long-form speech processing. We show that our Cross Architecture Layerwise Distillation (CALD) framework is capable of training student models that can process long sequences with efficient attention mechanisms while retaining the performance of pretrained transformer-based teacher models, using only downstream data without the need for additional re-pretraining. We also demonstrated that leveraging intermediate models during the teacher's fine-tuning process can be helpful for distillation on language tasks. We then showed that hybrid linear-sparse attention can mitigate the forgetfulness of linear attention and improve the retrieval of distant context. Particularly, we found that our contextualized lightweight learnable token eviction approach is particularly effective at improving the retrieval of distant context without any sacrifice of theoretical and practical efficiency compared to standard linear attention. These findings suggest that efficient attention mechanisms can be a promising direction for long-form SLU, while our proposed methods address their limitations in training costs and forgetfulness.

Finally, the third phase applied these methodological advances back to the motivating application of interview analysis. We demonstrated that speech foundation models can be effectively adapted to the specific domain of French interview recordings to identify storytelling and other sub-optimal responses. We also observed the benefits of incorporating more contextual information for this task, while noting the limited help of audio clues in our study. These findings indicate that recent advances in SLU using foundation models can be valuable for the practical use of interview analysis.

Taken together, the findings of this thesis suggest that pretrained and generalist models are a viable foundation for SLU in long-form and low-resource settings, but only when carefully used. Strong performance requires improved semantic supervision, architectures that can handle extended speech efficiently, as well as accommodation to specific applications.

8.2 Future Work

This thesis explored a broad range of directions on using generalist models for SLU, and observe its strength and limitations across several settings. The results also point to a number of open questions that remain important for future research.

The first phase of the thesis highlighted the importance of richer semantic supervision for speech models, as well as the promise of LLMs for downstream spoken language understanding. Recent speech LLMs, such as the Qwen-Omni family, are typically built by post-training a combination of speech encoder and LLM using a cohort of speech tasks, including ASR, speech translation, speech QA, speaker recognition, etc. This corroborates one of the conclusions of this thesis: low-level acoustic objectives are not sufficient for learning strong semantic representations. A natural next step is therefore to develop more principled approaches to the design of such training pipelines, including the choice of auxiliary tasks, the balance between them, and the overall adaptation strategy. More broadly, an important long-term question

concerns the gap between speech and text pretraining paradigms. In contrast to the relatively unified framework of causal language modeling for LLMs, speech model pretraining remains more heterogeneous, and the integration of speech with language models is often deferred to a later, isolated stage. Bridging this gap and developing a more unified pretraining paradigm for data in different modalities may provide a more native way to capture both low-level and high-level structure and to make better use of data and computation.

The second phase of the thesis demonstrated the value of efficient attention mechanisms, and in particular the potential of hybrid linear-sparse designs for long-context modeling. At the same time, these results suggest that the design space of hybrid attention remains far from fully explored. Existing architectures often combine different attention mechanisms in relatively simple ways, for example by interleaving layers, yet the best strategy for coordinating their complementary strengths is still unclear. An important direction for future work is therefore to investigate more adaptive forms of cooperation between components, such as mechanisms through which the linear attention component can proactively delegate difficult long-range dependencies to a sparse or full-attention component. A second key direction concerns the sparse component itself. Although hybrid linear-sparse models offer clear efficiency advantages, there remains a substantial performance gap relative to the hybrid of linear and full attention. Reducing this gap will likely require more accurate token selection mechanisms, so that important information can be retained more reliably. A third key direction is to apply such hybrid attention to actual speech processing task. A brief exploration of dynamic chunking and compression of speech based on hybrid attention is presented in Appendix E, featuring promising but still preliminary results, and more in-depth investigation is critical. Progress on these questions could be important steps towards future LLM architectures beyond Transformer, and may be especially relevant for multimodal LLMs.

The third phase of the thesis examined the application of speech foundation models to interview analysis. Several important questions remain open in this setting. A first direction is to better understand how audio information can be exploited for this task, since the present study found only limited benefits from acoustic cues beyond the spoken content itself. A second direction is to evaluate more directly how efficient long-context modeling techniques can be integrated into this application when the complete context (e.g. the whole recording) is leveraged. A third challenge is to determine how recent speech LLMs can be adapted to this setting under an intermediate data regime: the available dataset is substantially larger than a typical few-shot prompting setup, yet still much smaller than the scale usually required for conventional fine-tuning. A fourth concern is due to the limitations of the dataset used in our study, covering a limited set of questions in a specific laboratory setting. This calls for investigation for the generalization of techniques with more diverse, realistic, and possibly noisier data. Beyond storytelling identification, it will also be important to extend these methods to broader interview-related constructs, including, ultimately, signals relevant to hirability and self-presentation quality. In the longer term, these advances may contribute to the development of practical systems such as coaching platforms with real-time feedback that support interviewees and interviewers through more reliable automatic analysis of spoken

responses.

A Details and Additional Results for Enhancing Spoken Language Understanding via Speech Translation

A.1 Bayesian Transfer Learning

As in the standard machine learning configuration, we determine the parameters by optimizing loss with an L2 regularizer, i.e. minimizing $\mathscr{L}(D;\boldsymbol{\theta}) + \alpha\|\boldsymbol{\theta}\|_2^2$ for the parameters $\boldsymbol{\theta} \in \mathbf{R}^N$ given data $D = \{(x, y)\}$ and the hyperparameter α, in which the cross-entropy loss $\mathscr{L}$ corresponds to the negative log-likelihood $-\log p(y|\boldsymbol{\theta})$ of the label upon model outputs. This can be formulated as maximum likelihood estimation (MLE) of $\boldsymbol{\theta}$ by maximizing $\log p(\boldsymbol{\theta}|D)$, which is equal to $\log p(D|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(D)$ by Bayes' theorem. With constant $p(D)$ and a zero-mean isotropic Gaussian prior $\mathscr{N}(0, \sigma^2\boldsymbol{I})$ on $\boldsymbol{\theta}$ with scalar σ, the optimization objective corresponds to

$$\log p(\boldsymbol{\theta}|D) \propto \log p(D|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \tag{A.1}$$

$$= \log p(D|\boldsymbol{\theta}) + \log(\mathscr{N}(\boldsymbol{\theta}; 0, \sigma^2 I)) \tag{A.2}$$

$$\propto -\mathscr{L}(D;\boldsymbol{\theta}) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\theta_i^2 \tag{A.3}$$

Hence L2 regularization can be viewed as giving an isotropic zero-mean Gaussian prior to the model parameters that assigns higher probability to close-to-zero parameters, with a larger α indicating a smaller scalar σ^2. While instead of zero, L2-SP (Li, Grandvalet, and Davoine, 2018) proposes to limit the parameter shift from the pretrained ones during fine-tuning by assigning a Gaussian prior $\mathscr{N}(\boldsymbol{\theta}_0, \sigma^2\boldsymbol{I})$ centered at pretrained parameters $\boldsymbol{\theta}_0$, which has been found to lead to better downstream performance.

Nevertheless, it is an over-simplification of the prior as different parameters are unequal and some parameters are more critical for the performance on the pretraining task than others. The importance of a parameter can be represented by posterior distribution $p(\boldsymbol{\theta}|D_p)$ near $\boldsymbol{\theta}_0$ on the pretraining data D_p that corresponds to the pretraining loss $\mathscr{L}(D_p;\boldsymbol{\theta}) \propto p(D_p|\boldsymbol{\theta})$. In this way, elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) assigns a Gaussian prior $\mathscr{N}(\boldsymbol{\theta}_0, \boldsymbol{\sigma}^2\boldsymbol{I})$ with diagonal covariance σ_i according to the estimated posterior distribution (i.e.

loss landscape) of θ_i on the pretraining task. A parameter θ_i with larger impact to the $\mathscr{L}(D_p;\boldsymbol{\theta})$ will have sharper $p(\theta_i|D_p)$ and smaller $\sigma_i^2 = 1/(\alpha F_i)$, thus less flexibility in fine-tuning, lower variance in the fine-tuning prior, and higher weight for its L2 regularizer under the goal of preserving knowledge for the pretraining task.

To estimate this posterior distribution or loss landscape on the pretraining data D_p, we can perform Taylor expansion for the log likelihood $\log f(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|D_p)$ near the parameters after pretraining, namely $\boldsymbol{\theta}_0$, which is assumed to be near to the optimum, making $\nabla \log f(\boldsymbol{\theta}_0) \approx 0$. Hence,

$$\log f(\boldsymbol{\theta}) = \log f(\boldsymbol{\theta}_0) + \nabla \log f(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T H_{\log f}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \cdots \tag{A.4}$$

$$\approx \log f(\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T H_{\log f}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) \tag{A.5}$$

Therefore, through a second-order expansion, $p(\boldsymbol{\theta}|D_p)$ is approximated by a Gaussian distribution corresponding to the negation of the quadratic term above, with $\boldsymbol{\theta}_0$ being the mean and the Hessian matrix corresponding to the inverse covariance. To estimate the Hessian matrix, we use Bayes' theorem and take a flat prior on D_p, forming

$$\begin{aligned} H_{\log f}(\boldsymbol{\theta}) &= \frac{\partial^2 \log p(\boldsymbol{\theta}|D_p)}{\partial \boldsymbol{\theta}^2} \\ &= \frac{\partial^2 \log p(D_p|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} \\ &= \mathrm{E}_{x \sim p(x|\boldsymbol{\theta})}[\frac{\partial^2 \log p(x|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}] \end{aligned} \tag{A.6}$$

While the following matrix is referred as an empirical Fisher matrix, and can be estimated by squared gradients as in Pascanu and Bengio (2014):

$$F = -\mathrm{E}_{x \sim p(x|\boldsymbol{\theta})}[\frac{\partial^2 \log p(x|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}], \tag{A.7}$$

Therefore, the posterior distribution of the parameter $\boldsymbol{\theta}$ on the pretraining task is approximated by a Gaussian distribution with the mean $\mu = \boldsymbol{\theta}_0$ and the inverse covariance $\Sigma^{-1} = F$. EWC further simplifies it by only considering diagonal terms.

A.2 Implementation Details

We pretrain the model following the common settings in the field on single 24GB V100 GPUs using the Adam optimizer with a learning rate schedule of 20k linear warmup steps from 0 to 1e-4, followed by an inverse-sqrt decay to 3e-5. Models are selected and early stopping is performed according to the WER or BLEU on the *dev* set. 5-beam search is used during evaluation. The PTLMs we use are the 24-layer "large" versions provided by Hugging Face. A

SLURP Script	Label	ASR	ST	ST+ASR
is there a meeting on my calendar this afternoon	calendar,query	calendar,set	calendar,query	calendar,query
look for apple pie recipe	cooking,recipe	qa,stock	qa,recipe	cooking,recipe
can you really see russia from alaska	qa,factoid	alarm,remove	qa,factoid	general,quirky
are there morning shows available	calendar,query	weather,query	recommendation, events	lists,query

Table A.1: Examples of SLURP IC benchmark and predictions produced by different models.

dynamic batching strategy is adopted to accommodate input utterance with different lengths. Accompanied with gradient accumulation, an average batch size of ~25 with ~500 target tokens per step is used. The wav2vec2 part is frozen for the first 10k steps, and utterances shorter than 0.1s or longer than 10s are not used during the first 20k steps. The L2 regularization with α=5e-3 is applied to the weights, except the Bayesian transfer learning experiments. The setting is similar for the fine-tuning cases except that the encoder is frozen during the initial steps, and for joint-training models a 1:3 ratio between data for the pretraining and target task is used. While for smaller datasets including MINDS-14, SLURP-Fr, and Spoken Gigawords, the data ratio, dropout rate, and learning rate schedule are further tuned to avoid overfitting. We also build and compare with several cascaded pipelines based on our ST model, for which we directly use the model outputs with beam search without external LM as the model already leverages a strong language model. More details could be found from the source code.

A.3 Examples

Several examples in the SLURP IC benchmark and the predictions from different models are provided here in Table A.1 for a more direct demonstration for the understanding capability of the models.

A.4 Dataset Details

Three new datasets are introduced in this work. Among them, SLURP-Fr is our main dataset for experiments on cross-lingual transfer, while we additionally carry out a series of experiments on Spanish (Es) to show that our methods work on more than one language. For the synthetic portion of SLURP-Fr/Es, we built the dataset based on MASSIVE, textual translation of SLURP, each using 4 speakers from Google TTS, with total 11.3H and 13.9H audio respectively. Therefore the contents are identical to MASSIVE. While for the real portion of SLURP-Fr, we leverage two native French speakers to read the held-out samples from MASSIVE. The dataset size and mean length (in seconds) of the utterances are given in Table A.2.

As for speech summarization, we follow MTL-SLT (Huang, Rao, et al., 2022) to build the

	train	dev	test	real
#Samples	11514	2033	2974	477
Avg. sec (Fr)	2.47	2.44	2.44	2.35
Avg. sec (Es)	3.03	3.01	3.01	/

Table A.2: Statistics of the SLURP-Fr/Es datasets.

	train	dev	test
#Samples	50000	1000	385
Mean length (sec)	9.21	9.30	9.24
Article word count	23.9	24.0	24.0
Headline word count	7.8	8.1	8.0

Table A.3: Statistics of the Spoken Gigaword dataset.

synthetic spoken version of Gigaword (Rush, Chopra, and Weston, 2015), using 9 speakers from Google TTS, with total 131.5H audio. We follow the data split in the original Gigaword dataset with a small and noisy test split (which we further filtered) and randomly sample from the train and dev split. The resultant size and mean length of the utterances are given in Table A.3.

A.5 Spanish Experiments

Similar to the default type of models using only data with English and French, we first introduce the Spanish data to the En+Fr ASR model or the En↔Fr ST model to pretrain a En+Fr+ES ASR model as well as a En↔Fr+En↔Es ST model, with satisfactory results as in Table A.4. Both ASR and ST models are then fine-tuned with joint training on SLURP, reaching 87.63% and 89.59% accuracy respectively. Then they are used for cross-lingual transfer to SLURP-Es.

A.6 EWC Weight Distribution

The distributions of the log estimated Fisher diagonals for each weight matrix or bias vector are illustrated in Figure A.1. It can be observed that most weights are concentrated around

	TEDx	MuST-C	CoVoST2
ASR WER↓	15.05%	8.94%	11.34%
ST BLEU↑	25.84%	30.06%	33.90%

Table A.4: Test results of the model with Spanish data added on the cleaned pretraining datasets given by word error rate (WER) for ASR and BLEU score for ST, with Spanish inputs for TEDx and CoVoST2, and English inputs for MuST-C.

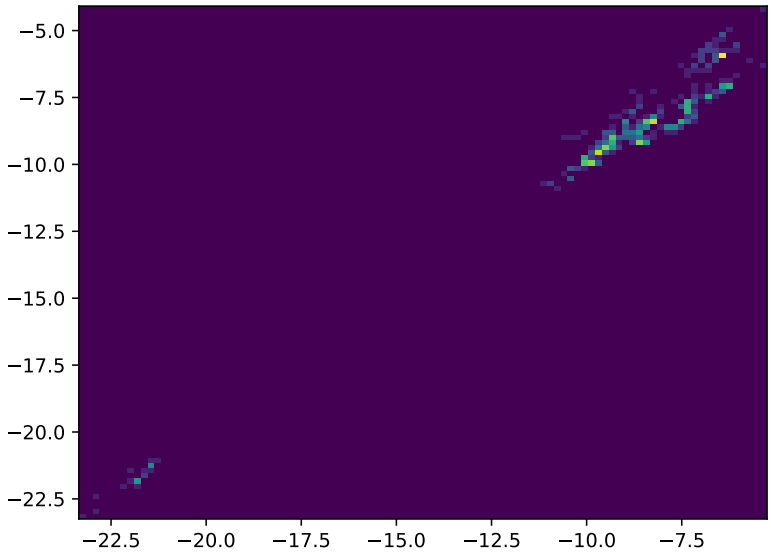


Figure A.1: The distribution of the estimated Fisher diagonals shown in heat map, with x-axis for the means of the log squared gradients of each weight or bias, and y-axis the standard deviation.

1e-5 to 1e-10, and they are close to each other as the standard deviations are at the similar magnitude. Hence with α=1e-7, most weights will reach the 1e-2 clamping threshold. The exceptions are the biases for key projection in attention modules, which correspond to the lower-left cluster and have much smaller weights.

B Details and Additional Results for Linearizing Pretrained Language Models

B.1 Implementation Details

We build the models as mentioned above, and train the models using the AdamW optimizer. On Mamba/Mamba2 models, we also find it important to apply layer normalization on hidden states prior to computing the layerwise distillation loss, i.e. we take the hidden states after the layernorm operation at the following layer. Otherwise the hidden states will have a rather wide range and bring training instability. We only use output distillation by setting $\alpha_{KD} = 1$ in ASR experiments. On other experiments, we find that the distillation on output probabilities actually leads to suboptimal results, and thus set $\alpha_{KD} = 0$. We always use $\alpha_{CE} = 1$. As for the hybrid mode of distillation, we start with the optimal target guided configuration, but switch to unguided mode when the loss is close to converge, generally at around 30% of the total training steps.

For each target task, we train the models based on its common training recipe for standard transformers, and further search and determine hyperparameters and the model checkpoint to use according to the validation performance. Details for each task are listed below, and more specific details can be accessed from our source code.

- Language processing: We train the model using a learning rate (LR) schedule of linear decay with 6% warmup steps. A total batch size of 32 is used. Other hyperparameters are listed in Table B.1.
- Language modeling: We train the model using a learning rate of 3e-4 and a schedule of cosine decay with 3% warmup steps and a 3e-5 minimum learning rate. We use a total batch size of 128, each with 2048 tokens. We also apply the weight decay of scale 0.1 on Mamba parameters following the Mamba training scheme, along with gradient norm clipped to 1. We set $\alpha_{LD} = 15$ in the experiments. FP16 mixed precision training is adopted.
- Speech processing: We train the model using a learning rate schedule of exponential

Table B.1: Hyperparameters for language processing tasks.

		QNLI	QQP	SST2	IMDB
Std. RoBERTa	LR	3e-5	3e-5	3e-5	1e-5
Unguided	LR	1e-5	1e-5	3e-5	1e-5
	Share mode	KV	Layer	KV	KV
Target Guided	LR	1e-4	1e-4	1e-4	3e-4
	α_{LD}	25	20	10	15
	Share mode	Layer	Layer	KV	KV
Traj. Guided	LR	3e-4	1e-4	1e-4	3e-4
	α_{LD}	30	25	30	40
	Share mode	Layer	KV	KV	KV
	T_u	5	1	3	3
Waypoint Guided	LR	1e-4	1e-4	1e-4	3e-4
	α_{LD}	10	20	5	25
	Share mode	KV	Layer	KV	KV

decay with 5000 warmup steps and 1.5e-6 minimum learning rate. We use a dynamic batch strategy that leads to a batch size of approximately 64 in average. We also apply the weight decay of scale 5e-3, along with gradient norm clipped to 1. For ASR, IC, and SID, we set α_{LD} to 15, 15, and 30, respectively. BF16 mixed precision training is adopted. We find that these hyperparameters work well across different modes of distillation.

B.2 Distillation Algorithms

Here we provide the pseudo-code to describe the training algorithms under different modes of distillation to facilitate understanding. Please refer to Chapter 5.3 for the notation scheme we use, including the model output terms and loss functions.

B.3 Computational Costs

The proposed method is presented as a more efficient alternative to re-doing the whole pretraining process on the target architecture, as the model pretraining is notoriously expensive and generally infeasible with academic computation. For example, Wav2Vec2-large pretraining took 5.2 days on 128 V100 GPUs (Baevski et al., 2020), while our target guided CALD experiment to convert Wav2Vec2-large into Mamba2 for ASR takes only 1.6 days on a single RTX3090 with just 24GB memory. Without the computation on the teacher model, the unguided training will be roughly 40% faster, but the accuracy degradation is unacceptable.

Algorithm 1 CALD training under the target-guided or hybrid mode

Require: `mode` (training mode: `Hybrid` or `TargetGuided`), T_h (number of guided steps in hybrid mode), T (total number of training steps), `dataloader`, $\alpha_{\text{CE}}, \alpha_{\text{KD}}, \alpha_{\text{LD}}$ (loss scaling hyperparameters)

Build and initialize the teacher model M_T and the student model M_S with the teacher target model
Replace the attention layers in M_S
Build the optimizer for M_S
for $i = 1$ to T **do**
    $(x, y) \leftarrow$ `next(dataloader)`
    $(y^{(s)}, H^{(s)}) \leftarrow M_S(x)$
    if `mode` = `Hybrid` **and** $i > T_h$ **then**
        $\mathscr{L} \leftarrow \alpha_{\text{CE}} \mathscr{L}_{\text{CE}}(y^{(s)}, y)$
    else
        $(y^{(t)}, H^{(t)}) \leftarrow M_T(x)$
        $\mathscr{L} \leftarrow \alpha_{\text{CE}} \mathscr{L}_{\text{CE}}(y^{(s)}, y)$
            $+ \alpha_{\text{KD}} \mathscr{L}_{\text{KD}}(y^{(s)}, y^{(t)})$
            $+ \alpha_{\text{LD}} \mathscr{L}_{\text{LD}}(H^{(s)}, H^{(t)})$
    end if
    $\mathscr{L}$.`backward()`
    Update the parameters of M_S using the optimizer
end for

Algorithm 2 CALD training under the trajectory-guided mode

Require: T_u (interval between teacher model updates), T (total number of training steps), `dataloader`, $\alpha_{\text{CE}}, \alpha_{\text{KD}}, \alpha_{\text{LD}}$ (loss scaling hyperparameters)

Build and initialize the teacher model M_T and the student model M_S with the teacher source model (the original pretrained model)
Replace the attention layers in M_S
Build the optimizer for M_S and M_T
for $i = 1$ to T **do**
    $(x, y) \leftarrow$ `next(dataloader)`
    $(y^{(t)}, H^{(t)}) \leftarrow M_T(x)$
    if $i \bmod T_u = 0$ **then**
        $\mathscr{L}_T \leftarrow \mathscr{L}_{\text{CE}}(y^{(t)}, y)$
        $\mathscr{L}_T$.`backward()`
        Update the parameters of M_T using the optimizer
    end if
    $(y^{(s)}, H^{(s)}) \leftarrow M_S(x)$
    $\mathscr{L} \leftarrow \alpha_{\text{CE}} \mathscr{L}_{\text{CE}}(y^{(s)}, y)$
        $+ \alpha_{\text{KD}} \mathscr{L}_{\text{KD}}(y^{(s)}, y^{(t)})$
        $+ \alpha_{\text{LD}} \mathscr{L}_{\text{LD}}(H^{(s)}, H^{(t)})$
    $\mathscr{L}$.`backward()`
    Update the parameters of M_S using the optimizer
end for

Algorithm 3 CALD training under the waypoint-guided mode

Require: W (list of waypoint models), T_w (interval between waypoint switching), T (total number of training steps), `dataloader`, $\alpha_{\text{CE}}, \alpha_{\text{KD}}, \alpha_{\text{LD}}$ (loss scaling hyperparameters)

Build and initialize the teacher model M_T with W_1; set $w_i \leftarrow 1$
Build and initialize the student model M_S from the teacher source model (the original pretrained model)
Replace the attention layers in M_S
Build the optimizer for M_S
for $i = 1$ to T **do**
    $(x, y) \leftarrow$ `next(dataloader)`
    $(y^{(s)}, H^{(s)}) \leftarrow M_S(x)$
    $(y^{(t)}, H^{(t)}) \leftarrow M_T(x)$
    $\mathscr{L} \leftarrow \alpha_{\text{CE}}\mathscr{L}_{\text{CE}}(y^{(s)}, y)$
        $+ \alpha_{\text{KD}}\mathscr{L}_{\text{KD}}(y^{(s)}, y^{(t)})$
        $+ \alpha_{\text{LD}}\mathscr{L}_{\text{LD}}(H^{(s)}, H^{(t)})$
    $\mathscr{L}$.`backward()`
    Update the parameters of M_S using the optimizer
    if $i \bmod T_w = 0$ **then**
        $w_i \leftarrow w_i + 1$
        Re-initialize M_T with W_{w_i}
    end if
end for

The hybrid approach is faster owing to more than half of the training steps being under the unguided mode, while reaching better performance than the target guided one. The waypoint guided model doesn't involve any extra computation compared to the target guided one, and hence takes roughly the same time. In NLP experiments, the trajectory guided models enjoy better performance at the cost of more computation. For example, on QNLI with layer-shared Linformer projection, the target or waypoint guided training takes around 3.4 hours under our training configuration with a single RTX3090, while the trajectory guided training takes 3.5 hours ($T_u = 5$) to 4.0 hours ($T_u = 1$). The unguided training takes around 3.0 hours. To sum up, the CALD approach saves much computation compared to re-pretraining, and the extra cost compared to the unguided training is limited.

We also provide a brief evaluation on the inference speed of transformer-based versus Mamba2-based ASR models we used in Chapter 5.4.4, as shown in Figure B.1. With asymptotic linear time complexity, the Mamba2 model will be much faster given sufficiently long inputs compared to the quadratic time transformers, although the actual speedup depends on the exact implementation.

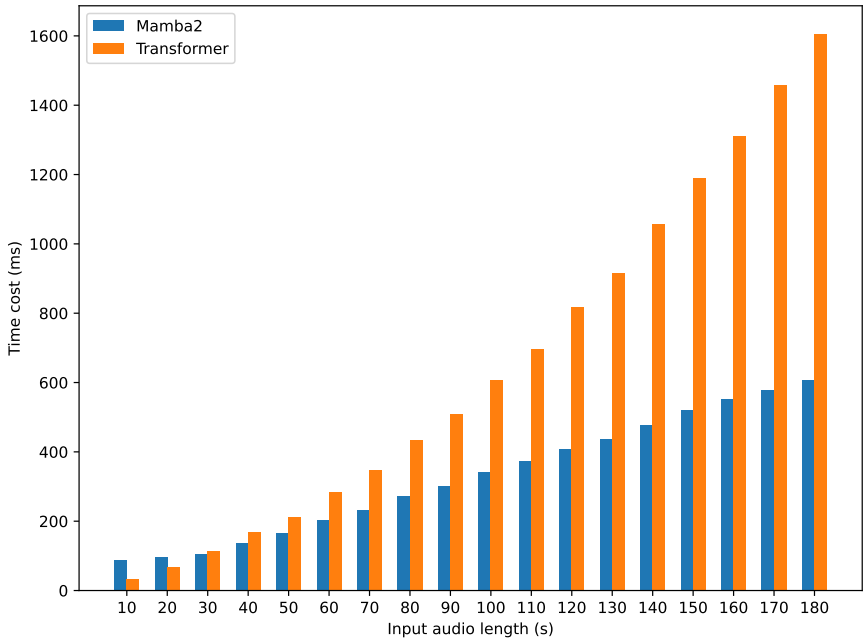


Figure B.1: Time costs for Mamba2 and transformer-based models performing ASR on audio samples with different lengths, averaged by 5 runs on a single RTX3090.

C Details for Hybrid Linear-Sparse Attention

We train our models using Flame (Zhang and Yang, 2025) based on TorchTitan (Liang et al., 2024) and refer to their training recipes, while adapted to our computational resources available. All the models are implemented based on Flash Linear Attention (FLA). Particularly, FLA implements the prefilling kernel for NSA, and we develop the decoding kernel based on it. It is noteworthy that NSA implemented by FLA is a simplified version that directly use mean pooling to produce the compressed $\boldsymbol{k}_m^B$ and $\boldsymbol{v}_m^B$, which may further limit its capability. We also develop the LTE prefilling kernel upon the FLA attention kernel. Both the 0.4B and 1.4B models have 24 layers, with hidden size = 1024 and 2048 respectively. As for GDN layers, we use 8 heads and expand_v = 1. As for attention layers, we set number of heads $H_q = H_{kv} = 16$ at 0.4B, and grouped-query attention (Ainslie, Lee-Thorp, et al., 2023) with $H_q = 32$, $H_{kv} = 8$ at 1.4B. Similar configurations are also adopted by Wang, Zhu, Abreu, et al. (2025). The exception is that NSA requires at least 16 group size due to hardware constraints, hence we use $H_q = 32$, $H_{kv} = 2$. Token embeddings are tied on 0.4B models.

The LTE module consists of a 3-layer 1D grouped convolution stack, using kernel size = 3 and dilation = 2, yielding a receptive field $R = 13$. Each layer halves the channel width. Each layer is followed by a Swish activation and a dropout layer with $p = 0.1$. It then passes linear projection per head implemented as grouped convolution with kernel size = 1, and produces the retention score r after sigmoid. We implement directly with Conv2D to avoid tensor manipulation overheads. The number of groups equals the number of KV heads to process each head independently and efficiently. Inputs are zero-padded on the left, and thanks to the deferred computation with SWA, padding is not necessary on the right. The LTE module will add ~1.7% and ~0.7% parameters on 0.4B and 1.4B models respectively.

We use the AdamW optimizer with learning rate = 3e-4 under cosine decay. At 0.4B/1.4B, we use global batch sizes of 0.5M/1.0M tokens, context length 4096, for 20480/30720 steps with 1024/512 warmup steps. As for LTE training, we initialize the regularization weight matrix λ to 1e-9, and update it via a feedback loop every $u = 32$ steps for stability, following the pseudocode given in Algorithm 4. The update is based on the exponential moving average $\bar{c}$ of the retention count c, with coefficient $\alpha_{ema} = 2/(1 + u/2)$ to approximate the average within

the update cycle.

Algorithm 4 Pseudo-code for the LTE training loop.

$\lambda_{:} \leftarrow 10^{-9}$ ▷ Initialize λ matrix (shaped [#layers, #heads])
for step $\leftarrow 0$ to T **do** ▷ T is total training steps
  $r, c, \ell \leftarrow$ model(fetch_inputs()) ▷ Retention score, retention count, and LM loss
  $\ell \leftarrow \ell + \lambda \cdot \mathrm{ReLU}(r)$ ▷ Augment loss with L1 penalty on r
  optimizer.step()
  $\bar{c} \leftarrow \alpha_{ema} \cdot \bar{c} + (1 - \alpha_{ema}) \cdot c$ ▷ Exponential moving average of retained token counts
  if step mod $u = 0$ **then** ▷ Every u steps, update λ
    Synchronize $\bar{c}$ across devices
    $s \leftarrow \mathbf{1}(\bar{c} > L) - \mathbf{1}(\bar{c} < 0.95 \cdot L)$ ▷ Direction to adjust λ
    $\lambda \leftarrow \lambda \cdot \alpha^{s}$
    $\lambda \leftarrow \min(\lambda, 1.0)$ ▷ Clamp λ to at most 1.0
    $\lambda[\lambda < 10^{-9}] \leftarrow 0$ ▷ Eliminate near-zero values
    $\lambda[\bar{c} > L \ \&\ \lambda = 0] \leftarrow 10^{-9}$ ▷ Re-enable penalty when too many retained
  end if
end for

During LTE decoding, we directly use the `flash_attention_kv_cache` API by Flash Attention, which supports different past KV cache length per sample in a batch by passing in `cache_seqlens`. Since we have different cache length per head, we merge the dimensions for KV head and batch size, leaving a KV cache with the number of query heads equal to the GQA group size. As for evaluation, we use LM-eval-harness for short-context and RULER evaluations, and leverage the reference implementation by Arora, Timalsina, et al. (2024) for EVAPORATE benchmarks. We follow their input truncation scheme, but only truncate the input length to 4K tokens when it exceeds 4K to better reflect long-context performance. Inference is carried out using greedy decoding in a single run in FP32 for stability, except that GDN and Flash Attention kernels support FP16 only, hence we perform all token-mixing ops in FP16 for parity. For compute-cost measurements, we report the total elapsed time between the CUDA event pairs at layer entry after a full warm-up phase; for prefilling we average the middle three of five runs. Samples from the training dataset are used as inputs.

D Details for Interview Analysis

D.1 Response Examples

Regarding the past behavior question:

> Can you tell me about a situation where one of your employees didn't complete a task on time? How did you react to this problem?

, an example of a pseudo-story response from our data is:

> *hmm for example a colleague* (PS) / *well I try if I'm able to save the situation in quotes* (PTA) / *well I give him my help* (PTA) / *and so that everyone gets out of it* (PTA) / *and but if I can't do anything* (PTA) / *well I couldn't do anything* (PTA) / *well then I wouldn't be the one to uh try to sink the person or whatever* (PTA) / *I'll always try to help him* (PTA)

While given the question:

> Can you tell me about a situation where you were responsible for carrying out several tasks in a short space of time? Tell me how you organized yourself to deal with this situation.

An example of a response loaded with other non-story utterances is:

> *okay uh well the* [sighs] *multitasking uh of course she uh doing several tasks uh at the same time is not my forte* (SD) / *but uh I think we w- we can all do several tasks uh at the same time* (VO) / *so uh me that's what- the style I prefer uh to do one at a time uh* (SD) / *but uh um there are times when we're obliged to do uh several uh*

things uh at the same time (VO) / *uh and uh the switch and manage the tasks at the same time it's a uh it's a bit tiring for the brain* (VO) / *but uh uh I can do it* (SD) / *so uh it's not it's what I prefer* (SD) / *uh but uh I can do it* (SD)

D.2 Dataset Preparation

There were 254 French-speaking undergraduates (59.8% women, $M_{age} = 21.60$, $SD = 3.07$) recruited through a pool of participants from a Swiss university. Fifty-two % of participants held a high school diploma, 48.8% had a university degree (bachelor: 39% and master: 9.8%) and 1.3% held other types of diplomas. Participants had already experienced 1.94 job interviews on average ($SD = 2.79$). Participants were offered monetary compensation consisting of a fixed sum and a performance-based bonus.

Participants engaged in a two-stage lab experiment. In the first stage, they started completing a consent form and a personality questionnaire. Then, they performed a work sample (i.e., two role plays), imitating a layoff scenario. After each role-play, participants self-rated their performance. About one week later, participants returned for a mock job interview for a managerial job position, randomly assigned to either FTF or AVI conditions. Once the job interview was completed, participants filled out a final short questionnaire. Throughout both the work sample and the job interview, participants were audio-recorded with a PG85 microphone connected to one Focusrite Scarlett 2i2 4th Generation, and video recorded using a front Logitech HD Pro C920 webcam and a side Sony CX625 camera.

During the interview, all participants were asked the same four questions. The first question was about self-presentation asking "Could you please introduce yourself and summarize your background in a few words?" The second and third questions were past-behavior questions targeting managerial competencies (e.g., multitasking: "Can you describe a situation where you were responsible for completing several tasks within a short period? How did you organize yourself to manage this situation?"). The fourth question was also a past-behavior question that targeted the role play of the previous week (i.e., "Can you tell us about a recent situation in which you had to give bad news to someone?"). In the FTF condition, participants were interviewed by a mock recruiter trained to follow a script. After the experimenter briefed them about the job interview and left the room, the recruiter greeted the participant, started with a brief overview of the interview, and continued asking the questions before concluding the interview. In the AVI condition, participants used the same AVI platform as in Roulin, Wong, et al. (2023)'s study to read and respond to on-screen questions. Following the experimenter's instructions, participants had up to 20 seconds to read each question and up to 5 minutes to answer. They had the flexibility to start and finish their responses before the respective timers ended, for all interview questions.

The per-utterance categories include: story (S) considered as a "set of events related to a unique past episode, characterized by a unity of time or action"; pseudo-story (P) defined as "a description of a generic situation or recurrent set of similar situations, without unity of

time or action", and decontextualized assertions about interviewees' opinions (value/opinion, VO), justification for their actions (justification, JS) or self-descriptions (self-description, SD) (Bangerter, Corvalan, and Cavin, 2014, p. 6). Any utterance not falling into these specific categories was annotated as "other," (OR). Categories were mutually exclusive. Narrative utterances (i.e., story and pseudo-story) were further annotated according to the STAR model (Kessler, 2006): Utterances providing information about the context were annotated for *situational* elements (SS for stories, PS for pseudo-stories), utterances providing information about roles and actions of protagonists were classified as *task*- and *action*-related elements (STA/PTA), and those referring to the results obtained were classified as *results*-related elements (SR/PR).

E Exploration in Dynamic Chunking and Compression of Long-form Speech

E.1 Introduction

Modern speech understanding systems largely rely on speech foundation models (SFMs) such as Whisper (Radford, Kim, et al., 2023) and HuBERT (Hsu, Bolte, et al., 2021). These models are pretrained on large-scale speech corpora and are used either directly or after fine-tuning for downstream tasks such as automatic speech recognition (ASR) and spoken language understanding (SLU), as well as the speech encoders in multimodal large language models (MLLMs). Despite their success, these Transformer-based models remain limited by several coupled factors: i) Computational cost: They exhibit the inherent quadratic time and linear memory costs due to the attention mechanism, which is prohibitive with long-form inputs; ii) Long-context generalization: They are typically pretrained on short audio clips (e.g. 30 seconds for Whisper) and struggle to generalize to longer audio, which further translates into the audio context window constraints in many MLLMs (Chu et al., 2024; Tang, Yu, et al., 2024; Ghosh, Kumar, et al., 2024); iii) Temporal redundancy: They operate at a temporal resolution much higher than the corresponding semantic representation (e.g. the 50Hz frame rate for HuBERT, compared to roughly 2-3 tokens per second for the transcribed text). This exacerbates problem i) and ii), and further leads to degraded MLLM performance due to the modality mismatch (Hsu, Zhang, et al., 2026).

However, speech understanding in the real world is inherently long-context: meetings and calls span minutes to hours, and their interpretation depends on discourse-level structure and long-range dependencies. A standard strategy for handling such inputs is to divide the audio into fixed-length segments and process them separately, as in Whisper and several MLLMs that support long-form audio (KimiTeam et al., 2025b; Ghosh, Kong, et al., 2025; Goel, Ghosh, et al., 2025; Liu, Ehrenberg, et al., 2025). These MLLMs often leverage an adaptor, such as a CNN stack, that downsamples the encoded speech to a lower frame rate, for example 12.5 Hz, to better match the text modality. Under this sliding window approach, the SFM itself has no access to out-of-window context, which can hinder long-form recognition performance (Koluguri et al., 2024; Chen, Kano, et al., 2024; Hori et al., 2021; Flynn and Ragni,

2024; Fox et al., 2024). Furthermore, fixed-rate compression in such adaptors ignores a basic property of speech: its information density is highly non-uniform over time. Long stretches of silence, background noise, or sustained vowels may span hundreds of milliseconds, whereas transient consonantal events can burst within only a few milliseconds. Aggressive fixed-rate compression may therefore fail to exploit highly compressible stationary chunks while discarding information from rapidly changing regions that are more sensitive to temporal compression. This motivates a natural but underexplored question: Can we dynamically split speech into variable-length, content-dependent chunks and compress each chunk into a single frame, thereby approaching the temporal resolution of text without substantially degrading fine-grained acoustic discrimination?

Very recently, H-Net (Hwang, Wang, and Gu, 2025) suggested that linear attention is well suited to modeling noisy and redundant sequences. By combining unidirectional linear attention, similarity-based boundary detection, and temporal smoothing, H-Net learns to hierarchically segment and compress raw UTF-8 text or DNA sequences into meaningful chunks, yielding more effective sequence modeling than standard Transformers. This capability is particularly desirable for speech, given its non-uniform information density, motivating us to explore dynamic chunking and compression for speech sequences. We therefore conduct a preliminary exploration in which lightweight dynamic chunking layers are inserted into pretrained Whisper models for ASR, adding less than 1% additional parameters. We then adapt these models to long-form ASR with inputs up to 30 minutes, using an aggressive 9× temporal compression ratio. The empirical results provide preliminary evidence that this technique shows promising performance and merits further investigation.

E.2 Methods

E.2.1 Model Backbone

We adopt pretrained Whisper large-v3 as the encoder-decoder ASR backbone and introduce several modifications to support long-form speech processing. The original Whisper uses absolute positional encodings: sinusoidal encodings in the encoder and learned embeddings in the decoder. These absolute encodings impose fixed maximum input and output lengths in the original implementation and become inconvenient for much longer contexts; for example, a 30-minute audio sequence would require 90,000 separate positional embeddings. In contrast, relative positional schemes such as RoPE do not rely on a fixed-sized embedding table and are often better suited to length extrapolation. We therefore drop the absolute positional embeddings, and fine-tune the model to use RoPE in their self-attention layers, similar to Gumma, Chitale, and Bali (2025). For compressed frames, we compute RoPE using the original start position of each chunk, rather than its index in the compressed sequence, which improves training stability.

In addition, Whisper pads input sequences to a fixed length of 30 seconds, and feeding

sequences with mismatched lengths can degrade performance. Because long-form audio samples vary substantially in length, fixed-length padding is highly inefficient. We therefore also adapt the model to accept variable-length inputs during this fine-tuning stage. Finally, to process long-form inputs efficiently without altering the existing encoder architecture, we replace full self-attention with sliding-window attention using a 60-second bidirectional window. The encoder henceforth reduces to full attention given <30-second inputs, but allows an efficient gathering and processing of local context given long-form inputs.

E.2.2 Chunking Layer

Following H-Net (Hwang, Wang, and Gu, 2025), we build the chunking layer from three components: a linear-attention encoder, a similarity-based routing module, and a smoothing module. Specifically, given a sequence of encoded frames $\boldsymbol{X} \in \mathbb{R}^{N\times d}$, we first apply a causal linear attention layer to obtain contextualized representations $\hat{\boldsymbol{X}}$. In our experiments, we instantiate this encoder with DeltaNet (Schlag, Irie, and Schmidhuber, 2021; Yang, Wang, Zhang, et al., 2024) rather than Mamba2 due to its strength and efficiency. We then identify chunk boundaries at points with notable representation shift, which signify transitions from one acoustic unit to another. To this end, a routing module measures the cosine distance between adjacent frames through projected queries and keys, with trainable $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$:

$$\boldsymbol{q}_t = \boldsymbol{W}_q \hat{\boldsymbol{x}}_t, \qquad \boldsymbol{k}_t = \boldsymbol{W}_k \hat{\boldsymbol{x}}_t, \qquad p_t = \frac{1}{2}\left(1 - \frac{\boldsymbol{q}_t^\top \boldsymbol{k}_{t-1}}{|\boldsymbol{q}_t||\boldsymbol{k}_{t-1}|}\right) \in [0,1], \tag{E.1}$$

with $p_1 = 1$ by definition. During training, we follow DLCM (Qu et al., 2026) to sample the hard boundary indicator b_t from a Bernoulli distribution by a sharpened version of p_t to encourage exploration. During inference, we deterministically set the hard boundary indicator as $b_t = \mathbf{1}[p_t \geq 0.5]$. To avoid over-compression, we limit the maximal chunk length to 1 second. Ideally, when adjacent frames $\hat{\boldsymbol{x}}_{t-1}$ and $\hat{\boldsymbol{x}}_t$ span a boundary between acoustic units, e.g. phonemes or syllable, the projection $\boldsymbol{q}_t$ will diverge from $\boldsymbol{k}_{t-1}$, yielding a larger p_t and eventually the start of a new chunk.

Given a list of start-of-chunk indices s_t where $b_{s_t} = 1$, the downsampler then compresses $\hat{\boldsymbol{X}}$ by taking the mean of each chunk, i.e. $\hat{\boldsymbol{z}}_t = \text{meanpool}([\hat{\boldsymbol{x}}_{s_t}, \hat{\boldsymbol{x}}_{s_t+1}, ..., \hat{\boldsymbol{x}}_{s_{t+1}-1}])$. Unlike the original H-Net, which applies smoothing in the dechunking layer, we apply smoothing immediately after downsampling, directly over the compressed chunk representations. Concretely, we use an exponential moving average over chunks,

$$\boldsymbol{z}_t = P_t \hat{\boldsymbol{z}}_t + (1 - P_t)\hat{\boldsymbol{z}}_{t-1}, \tag{E.2}$$

where $P_t = p_{s_t}$. This allows differentiable learning of boundary probabilities and potential error correction for boundaries with low p by sharing information from the preceding chunk.

We further adopt H-Net's auxiliary ratio loss to encourage the router to approach the desired

compression ratio. Given the original sequence length N and the prescribed compression ratio D, we compute the empirical boundary rate $F = \frac{1}{N}\sum_{t=1}^{N} b_t$, and the average boundary probability $G = \frac{1}{N}\sum_{t=1}^{N} p_t$. We then use

$$\mathscr{L}_{\text{ratio}} = \frac{D}{D-1}\left((D-1)FG + (1-F)(1-G)\right), \tag{E.3}$$

which provides continuous feedback through G while encouraging F to approach the desired value $1/D$. We add this term to the ASR loss with a coefficient of 0.05, so that the model is trained to optimize transcription accuracy while maintaining the prescribed compression ratio.

E.3 Empirical Studies

E.3.1 Experiment Settings

We compare the following systems in our experiment:

- **Whisper**: We directly use the pretrained Whisper (large-v3) model with its standard long-form transcription pipeline without fine-tuning. The underlying Whisper model operates on short audio windows. To handle long audio, Whisper uses a complicated sequential chunkwise transcription algorithm. It determines window shifts from emitted timestamps, conditions each 30-second chunk on previous transcriptions, actively detects potential errors using handcrafted heuristics, and retries decoding with alternative generation parameters when such errors are found.
- **Post-DC**: This model applies dynamic chunking twice after the transformer encoder, with each layer using a 3× compression ratio.
- **Pre+Post-DC**: This model applies dynamic chunking both before and after the Transformer encoder, with each layer using a 3× compression ratio. This variant is more efficient because the pre-encoder chunking layer reduces the sequence length processed by the Transformer encoder.
- **Post-CNN**: This model replaces dynamic chunking with fixed-rate CNN downsampling. It applies two stride-3 CNN layers after the Transformer encoder, yielding the same total 9× compression ratio as Post-DC.

The latter three models require fine-tuning but make it possible to process long-form inputs within our GPU memory constraint thanks to an aggressive total 9x compression.

We conduct the experiments using a long-form ASR dataset derived from the expanded TED-LIUMv3 corpus (Fox et al., 2024). The training split contains 484.5 hours of training data from 2.1K TED talks (documents) with durations ranging from 2 to 30 minutes. The dev set contains

Table E.1: Results (word error rate) of different models on long-form TED-LIUM data.

	utterance		document	
WER	dev	test	dev	test
Whisper	2.54	2.07	3.87	8.35
Pre+Post-DC	4.13	3.85	5.01	6.24
Post-DC	2.93	2.79	3.78	3.87
Post-CNN	4.26	3.16	42.49	61.10

8 talks totaling 1.7 hours, and the test set contains 11 talks totaling 3.0 hours. The documents are further segmented into 64.8K utterance-level samples, each less than 30 seconds long.

We fine-tune Whisper (large-v3) using the dataset in three stages: 1) We adapt the model to RoPE and variable-length inputs, implemented using FlashAttention's packed varlen mode, by fine-tuning it on the utterance-level data for 2K steps; 2) We insert the additional compression/dynamic chunking layers and fine-tune the model on the utterance-level data for 5K steps; 3) We fine-tune the model on a mixture of utterance- and document-level data for another 5K steps, yielding final models that can process long-form inputs directly. We use a learning rate=3e-4 and global dynamic batching with a maximal total audio duration of 128 minutes per batch. All the dynamic chunking models reach the desired compression ratio during training. We then evaluate models on both utterance and document levels using the checkpoint with the best performance on the document-level dev set. We use greedy decoding for evaluation and report word error rate (WER) after applying Whisper text normalization.

E.3.2 Results

Table E.1 reports the WER results of all evaluated models. All compressed models retain reasonable utterance-level performance, despite the aggressive 9× compression. Among them, Post-DC performs most consistently and achieves the lowest WER on the document-level test set across all models. Pre+Post-DC performs worse, suggesting that premature compression prior to bidirectional context aggregation from the Transformer encoder may hurt recognition accuracy. Post-CNN performs worst, especially at the document level, suggesting that aggressive fixed-rate compression is substantially less robust than adaptive dynamic chunking in this setting. Nevertheless, Post-DC still underperforms Whisper on utterance-level evaluation. In particular, qualitative inspection suggests that most document-level errors are not local phonetic confusions, but sequence-level failures such as skipping, repeating, and hallucinations. These errors suggest that a major challenge of long-form encoder-decoder ASR still lies in the maintenance of robust alignment between the acoustic contexts and generated tokens in a single but extensive decoding pass. Therefore, while dynamic compression can reduce the computational burden of long-form inputs, it may not by itself solve the alignment and decoding challenges of long-form ASR.

E.4 Related Work

Related work on speech foundation models and efficient attention has been discussed in Chapter 2. For the chunking and compression of speech frames, several recent approaches consider frame merging/pruning directly through frame similarity in MLLMs (Bhati, Thomas, et al., 2025; Wang, Li, Ma, et al., 2026; Lin, Fu, et al., 2025; Xiang et al., 2026). Related ideas also appear in neural codecs, known as variable frame rate codecs (Li, Qian, et al., 2025; Wang, Guo, et al., 2025; Zheng et al., 2025). These approaches typically rely on heuristic or post-hoc criteria rather than an explicit learned router for boundary prediction. More related to our scheme, continuous integrate-and-fire approaches dynamically chunk the input sequence by accumulating a segmentation score over time, and start a new chunk whenever the accumulated score exceeds a threshold (Dong and Xu, 2020; Fan et al., 2022; Dong, An, et al., 2023; Meng et al., 2023). Alternatively, Zuo, Zhang, et al. (2025) determine chunk boundaries using the predictive entropy of a separate frame-level language model. Recently, H-Net introduces the hierarchical dynamic chunking and compression approach and applies it to byte-level token-free language modeling and DNA modeling (Hwang, Wang, and Gu, 2025). Dynamic Large Concept Model (DLCM) (Qu et al., 2026) and ConceptMoE (Huang, Zhou, et al., 2026) further extend this idea to compress tokens into high-level semantic units in standard language modeling tasks. H-Net-style chunking has also been applied to video-to-audio generation (Simon et al., 2026). However, its application to long-form speech processing remains underexplored.

E.5 Conclusion

This appendix chapter presented a preliminary exploration of H-Net-style dynamic chunking for long-form ASR. The results suggest that post-encoder dynamic chunking can support aggressive temporal compression while maintaining good recognition accuracy, and that it is substantially more robust than a fixed-rate CNN baseline under the same nominal compression ratio. However, there is still a gap with the sequential chunk-by-chunk short-context transcription. These findings suggest that dynamic chunking is a promising compression mechanism, but that further work is needed before it can serve as a reliable approach for long-form speech processing.

Bibliography

Aguilar, Gustavo, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo (2020). "Knowledge Distillation from Internal Representations". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*. Vol. 34, pp. 7350–7357. DOI: 10.1609/aaai.v34i05.6229.

Ainslie, Joshua, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai (2023). "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 4895–4901.

Ainslie, Joshua, Tao Lei, et al. (2023). *CoLT5: Faster Long-Range Transformers with Conditional Computation*. arXiv preprint arXiv:2303.09752.

Ainslie, Joshua, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang (2020). "ETC: Encoding Long and Structured Inputs in Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Online: Association for Computational Linguistics, pp. 268–284.

Ali, Mohamed Nabih, Daniele Falavigna, and Alessio Brutti (2025). *MLMA: Towards Multilingual ASR With Mamba-based Architectures*. arXiv preprint arXiv:2510.18684.

Allgower, Eugene L and Kurt Georg (1990). *Numerical continuation methods: An introduction*. Vol. 13. Springer Science & Business Media.

Anagnostidis, Sotiris, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann (2023). "Dynamic Context Pruning for Efficient and Interpretable Autoregressive Transformers". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Ardila, Rosana, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber (2020). "Common Voice: A Massively-Multilingual Speech Corpus". In: *Proceedings of the Twelfth Language Resources and Evaluation Conference, LREC 2020*. Marseille, France: European Language Resources Association, pp. 4218–4222.

Arivazhagan, Naveen, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey (2019). *The Missing Ingredient in Zero-Shot Neural Machine Translation*. arXiv Preprint arXiv:1903.07091.

Arora, Siddhant, Siddharth Dalmia, Xuankai Chang, Brian Yan, Alan W. Black, and Shinji Watanabe (2022). "Two-Pass Low Latency End-to-End Spoken Language Understanding". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 3478–3482. DOI: 10.21437/Interspeech.2022-10890.

Arora, Siddhant, Siddharth Dalmia, Pavel Denisov, et al. (2022). "ESPnet-SLU: Advancing Spoken Language Understanding Through ESPnet". In: *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7167–7171. DOI: 10.1109/ICASSP43922.2022.9747674.

Arora, Simran, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Re (2024). "Zoology: Measuring and Improving Recall in Efficient Language Models". In: *12th International Conference on Learning Representations, ICLR 2024*.

Arora, Simran, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Re (2024). "Simple linear attention language models balance the recall-throughput tradeoff". In: *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*. PMLR, pp. 1763–1840.

Arora, Simran, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré (2024). *Just Read Twice: Closing the Recall Gap for Recurrent Language Models*. arXiv preprint arXiv:2407.05483.

Artetxe, Mikel and Holger Schwenk (2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610. DOI: 10.1162/tacl_a_00288.

Bae, Sangmin, Bilge Acun, Haroun Habeeb, Seungyeon Kim, Chien-Yu Lin, Liang Luo, Junjie Wang, and Carole-Jean Wu (2025). *Hybrid Architectures for Language Models: Systematic Analysis and Design Insights*. arXiv preprint arXiv:2510.04800.

Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NeurIPS 2020*. Vol. 33. Curran Associates, Inc., pp. 12449–12460.

Bain, Max, Jaesung Huh, Tengda Han, and Andrew Zisserman (2023). "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *Proceedings of the 24th Annual Conference of the International Speech Communication Association, Interspeech 2023*, pp. 4489–4493. DOI: 10.21437/Interspeech.2023-78.

Bang, Yejin et al. (2023). "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity". In: *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), IJCNLP-AACL 2023*. Nusa Dua, Bali: Association for Computational Linguistics, pp. 675–718. DOI: 10.18653/v1/2023.ijcnlp-main.45.

Bangerter, Adrian, Paloma Corvalan, and Charlotte Cavin (2014). "Storytelling in the selection interview? How applicants respond to past behavior questions". In: *Journal of Business and Psychology* 29.4, pp. 593–604. DOI: 10.1007/s10869-014-9350-0.

Bangerter, Adrian, Eric Mayor, Skanda Muralidhar, Emmanuelle P Kleinlogel, Daniel Gatica-Perez, and Marianne Schmid Mast (2023). "Automatic identification of storytelling responses to past-behavior interview questions via machine learning". In: *International Journal of Selection and Assessment* 31, pp. 376–387. DOI: 10.1111/ijsa.12428.

Bastianelli, Emanuele, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser (2020). "SLURP: A Spoken Language Understanding Resource Package". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Online: Association for Computational Linguistics, pp. 7252–7262. DOI: 10.18653/v1/2020.emnlp-main.588.

Behrouz, Ali, Peilin Zhong, and Vahab Mirrokni (2024). *Titans: Learning to Memorize at Test Time*. arXiv preprint arXiv:2501.00663.

Belinkov, Yonatan, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass (2017). "What Do Neural Machine Translation Models Learn about Morphology?" In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2017*. Association for Computational Linguistics, pp. 861–872. DOI: 10.18653/v1/P17-1080.

Belinkov, Yonatan, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass (2020). "On the Linguistic Representational Power of Neural Machine Translation Models". In: *Computational Linguistics* 46.1, pp. 1–52. DOI: 10.1162/coli_a_00367.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan (2020). *Longformer: The Long-Document Transformer*. arXiv preprint arXiv:2004.05150.

Bhati, Saurabhchand, Yuan Gong, Leonid Karlinsky, Hilde Kuehne, Rogerio Feris, and James Glass (2024a). "DASS: Distilled Audio State Space Models Are Stronger and More Duration-Scalable Learners". In: *arxiv preprint arxiv:2407.04082*.

Bhati, Saurabhchand, Yuan Gong, Leonid Karlinsky, Hilde Kuehne, Rogerio Feris, and James Glass (2024b). *State-Space Large Audio Language Models*. arXiv preprint arXiv:2411.15685.

Bhati, Saurabhchand, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass (2025). *Towards Audio Token Compression in Large Audio Language Models*. arXiv preprint arXiv:2511.20973.

Bick, Aviv, Kevin Li, Eric P. Xing, J. Zico Kolter, and Albert Gu (2024). "Transformers to SSMs: Distilling Quadratic Knowledge to Subquadratic Models". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Biderman, Stella et al. (2023). "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". In: *Proceedings of the 40th International Conference on Machine Learning, ICML 2023*. PMLR, pp. 2397–2430.

Booth, Brandon M, Louis Hickman, Shree Krishna Subburaj, Louis Tay, Sang Eun Woo, and Sidney K D'Mello (2021). "Bias and fairness in multimodal machine learning: A case study of automated video interviews". In: *Proceedings of the 2021 International Conference on Multimodal Interaction*, pp. 268–277. DOI: 10.1145/3462244.3479897.

Brosy, Julie, Adrian Bangerter, and Eric Mayor (2016). "Disfluent responses to job interview questions and what they entail". In: *Discourse Processes* 53.5-6, pp. 371–391. DOI: 10.1080/0163853X.2016.1150769.

Brown, Nathan, Ashton Williamson, Tahj Anderson, and Logan Lawrence (2023). "Efficient Transformer Knowledge Distillation: A Performance Review". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 54–65. DOI: 10.18653/v1/2023.emnlp-industry.6.

Brown, Tom, Benjamin Mann, et al. (2020). "Language Models Are Few-Shot Learners". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NeurIPS 2020*. Vol. 33. Curran Associates, Inc., pp. 1877–1901.

Bruner, Jerome (1991). "The narrative construction of reality". In: *Critical inquiry* 18.1, pp. 1–21. DOI: https://doi.org/10.1086/448619.

Cabannes, Loïc, Maximilian Beck, Gergely Szilvasy, Matthijs Douze, Maria Lomeli, Jade Copet, Pierre-Emmanuel Mazaré, Gabriel Synnaeve, and Hervé Jégou (2025). *Short window attention enables long-term memorization*. arXiv preprint arXiv:2509.24552.

Cai, Zefan et al. (2024). *PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling*. arXiv preprint arXiv:2406.02069.

Campion, Michael A, David K Palmer, and James E Campion (1997). "A review of structure in the selection interview". In: *Personnel Psychology* 50.3, pp. 655–702. DOI: 10.1111/j.1744-6570.1997.tb00709.x.

Cao, Yue, Xiaojun Wan, Jinge Yao, and Dian Yu (2020). "MultiSumm: Towards a Unified Model for Multi-Lingual Abstractive Summarization". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*. Vol. 34, pp. 11–18. DOI: 10.1609/aaai.v34i01.5328.

Chang, Ya-Hsin and Yun-Nung Chen (2022). "Contrastive Learning for Improving ASR Robustness in Spoken Language Understanding". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 3458–3462. DOI: 10.21437/Interspeech.2022-781.

Chang, Kai-Wei, Wei-Cheng Tseng, Shang-Wen Li, and Hung-yi Lee (2022). "An Exploration of Prompt Tuning on Generative Spoken Language Model for Speech Processing Tasks". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 5005–5009. DOI: 10.21437/Interspeech.2022-10610.

Chang, Kai-Wei, Yu-Kai Wang, Hua Shen, Iu-thing Kang, Wei-Cheng Tseng, Shang-Wen Li, and Hung-yi Lee (2023). *SpeechPrompt v2: Prompt Tuning for Speech Classification Tasks*. arXiv Preprint arXiv:2303.00733.

Chen, Lei, Ru Zhao, Chee Wee Leong, Blair Lehman, Gary Feng, and Mohammed Ehsan Hoque (2017). "Automated video interview judgment on a large-sized corpus collected online". In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, pp. 504–509. DOI: 10.1109/ACII.2017.8273646.

Chen, Mark, Jerry Tworek, et al. (2021). *Evaluating Large Language Models Trained on Code*. arXiv Preprint arXiv:2107.03374.

Chen, Renze, Zhuofeng Wang, Beiquan Cao, Tong Wu, Size Zheng, Xiuhong Li, Xuechao Wei, Shengen Yan, Meng Li, and Yun Liang (2024). "ArkVale: Efficient Generative LLM Inference with Recallable Key-Value Eviction". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Chen, Sanyuan, Chengyi Wang, Zhengyang Chen, et al. (2022). "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1505–1518. DOI: 10.1109/JSTSP.2022.3188113.

Chen, William, Takatomo Kano, Atsunori Ogawa, Marc Delcroix, and Shinji Watanabe (2024). "Train Long and Test Long: Leveraging Full Document Contexts in Speech Processing". In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13066–13070. DOI: 10.1109/ICASSP48485.2024.10446727.

Chen, Yilong, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu (2024). "NACL: A General and Effective KV Cache Eviction Framework for LLM at Inference Time". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp. 7913–7926.

Cheng, Xuxin, Bowen Cao, Qichen Ye, Zhihong Zhu, Hongxiang Li, and Yuexian Zou (2023). "ML-LMCL: Mutual Learning and Large-Margin Contrastive Learning for Improving ASR Robustness in Spoken Language Understanding". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 6492–6505. DOI: 10.18653/v1/2023.findings-acl.406.

Cheng, Xuxin, Zhihong Zhu, Ziyu Yao, Hongxiang Li, Yaowei Li, and Yuexian Zou (2023). "GhostT5: Generate More Features with Cheap Operations to Improve Textless Spoken Question Answering". In: pp. 1134–1138. DOI: 10.21437/Interspeech.2023-41.

Chi, Zewen, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei (2021). "mT6: Multilingual Pretrained Text-to-Text Transformer with Translation Pairs". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 1671–1683. DOI: 10.18653/v1/2021.emnlp-main.125.

Chi, Zewen, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang (2020). "Cross-Lingual Natural Language Generation via Pre-Training". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*. Vol. 34, pp. 7570–7577. DOI: 10.1609/aaai.v34i05.6256.

Choi, Sehyun (2024). *Cross-Architecture Transfer Learning for Linear-Cost Inference Transformers*. arXiv Preprint arXiv:2404.02684.

Chowdhery, Aakanksha et al. (2023). "PaLM: Scaling Language Modeling with Pathways". In: *Journal of Machine Learning Research* 24.240, pp. 1–113.

Chu, Yunfei et al. (2024). *Qwen2-Audio Technical Report*. arXiv preprint arXiv:2407.10759.

Chuang, Yung-Sung, Chi-Liang Liu, Hung-yi Lee, and Lin-shan Lee (2020). "SpeechBERT: An Audio-and-Text Jointly Learned Language Model for End-to-End Spoken Question Answering". In: *Proceedings of the 21st Annual Conference of the International Speech Communication Association, Interspeech 2020*, pp. 4168–4172. DOI: 10.21437/Interspeech.2020-1570.

Chung, Yu-An, Chenguang Zhu, and Michael Zeng (2021). "SPLAT: Speech-Language Joint Pre-Training for Spoken Language Understanding". In: *Proceedings of the 2021 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021. Online: Association for Computational Linguistics, pp. 1897–1907. DOI: 10.18653/v1/2021.naacl-main.152.

Cobbe, Karl et al. (2021). *Training Verifiers to Solve Math Word Problems*. arXiv Preprint arXiv:2110.14168.

Conneau, Alexis, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli (2021). "Unsupervised Cross-Lingual Representation Learning for Speech Recognition". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 2426–2430. DOI: 10.21437/Interspeech.2021-329.

Conneau, Alexis, Ankur Bapna, et al. (2022). "XTREME-S: Evaluating Cross-lingual Speech Representations". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 3248–3252. DOI: 10.21437/Interspeech.2022-10007.

Conneau, Alexis and Guillaume Lample (2019). "Cross-Lingual Language Model Pretraining". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems, NeurIPS 2019*. Vol. 32. Curran Associates, Inc.

Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang (2021). "Pre-Training With Whole Word Masking for Chinese BERT". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 3504–3514. DOI: 10.1109/TASLP.2021.3124365.

Dao, Tri (2024). "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning". In: *12th International Conference on Learning Representations, ICLR 2024*.

Dao, Tri, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Re (2022). "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems, NeurIPS 2022*.

Dao, Tri and Albert Gu (2024). "Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality". In: *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*.

De, Soham et al. (2024). *Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models*. arXiv preprint arXiv:2402.19427.

De Mori, Renato (2007). "Spoken Language Understanding: A Survey". In: *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 365–376. DOI: 10.1109/ASRU.2007.4430139.

DeepSeek-AI (2025). *DeepSeek-V3.2-Exp: Boosting Long-Context Efficiency with DeepSeek Sparse Attention*. Tech. rep.

DeepSeek-AI et al. (2024). *DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model*. arXiv preprint arXiv:2405.04434.

Défossez, Alexandre, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour (2024). *Moshi: a speech-text foundation model for real-time dialogue*. arXiv preprint arXiv:2410.00037.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Di Gangi, Mattia A., Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi (2019). "MuST-C: A Multilingual Speech Translation Corpus". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), NAACL-HLT 2019*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2012–2017. DOI: 10.18653/v1/N19-1202.

Dong, Juechu, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He (2024). *Flex Attention: A Programming Model for Generating Optimized Attention Kernels*. arXiv preprint arXiv:2412.05496.

Dong, Linhao, Zhecheng An, Peihao Wu, Jun Zhang, Lu Lu, and Ma Zejun (2023). "CIF-PT: Bridging Speech and Text Representations for Spoken Language Understanding via Continuous Integrate-and-Fire Pre-Training". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 8894–8907. DOI: 10.18653/v1/2023.findings-acl.566.

Dong, Linhao and Bo Xu (2020). "CIF: Continuous Integrate-And-Fire for End-To-End Speech Recognition". In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6079–6083. DOI: 10.1109/ICASSP40776.2020.9054250.

Dong, Xin, Yonggan Fu, et al. (2025). "Hymba: A Hybrid-head Architecture for Small Language Models". In: *13th International Conference on Learning Representations, ICLR 2025*.

Du, Jusen, Weigao Sun, Disen Lan, Jiaxi Hu, and Yu Cheng (2025). *MoM: Linear Sequence Modeling with Mixture-of-Memories*. arXiv preprint arXiv:2502.13685.

Eriguchi, Akiko, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey (2018). *Zero-Shot Cross-lingual Classification Using Multilingual Neural Machine Translation*. arXiv Preprint arXiv:1809.04686.

Fan, Zhiyun, Zhenlin Liang, Linhao Dong, Yi Liu, Shiyu Zhou, Meng Cai, Jun Zhang, Zejun Ma, and Bo Xu (2022). "Token-level Speaker Change Detection Using Speaker Difference and Speech Content via Continuous Integrate-and-fire". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 3749–3753. DOI: 10.21437/Interspeech.2022-914.

Fang, Qingkai, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng (2025). "LLaMA-Omni: Seamless Speech Interaction with Large Language Models". In: *The 13th International Conference on Learning Representations, ICLR 2025*.

Fang, Yunhao, Weihao Yu, Shu Zhong, Qinghao Ye, Xuehan Xiong, and Lai Wei (2025). *Artificial Hippocampus Networks for Efficient Long-Context Modeling*. arXiv preprint arXiv:2510.07318.

Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang (2022). "Language-Agnostic BERT Sentence Embedding". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, pp. 878–891. DOI: 10.18653/v1/2022.acl-long.62.

Feng, Yuan, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou (2025). *Ada-KV: Optimizing KV Cache Eviction by Adaptive Budget Allocation for Efficient LLM Inference*. arXiv preprint arXiv:2407.11550.

Ferreira, Emmanuel, Bassam Jabaian, and Fabrice Lefèvre (2015). "Zero-Shot Semantic Parser for Spoken Language Understanding". In: *Proceedings of the 16th Annual Conference of the International Speech Communication Association, Interspeech 2015*, pp. 1403–1407. DOI: 10.21437/Interspeech.2015-57.

FitzGerald, Jack et al. (2023). "MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 4277–4302. DOI: 10.18653/v1/2023.acl-long.235.

Flynn, Robert and Anton Ragni (2024). "How Much Context Does My Attention-Based ASR System Need?" In: *Proceedings of the 25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pp. 217–221. DOI: 10.21437/Interspeech.2024-870.

Fox, Jennifer Drexler, Desh Raj, Natalie Delworth, Quinn McNamara, Corey Miller, and Migüel Jetté (2024). "Updated Corpora and Benchmarks for Long-Form Speech Recognition". In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13246–13250. DOI: 10.1109/ICASSP48485.2024.10446286.

Fu, Yao, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot (2023). "Complexity-Based Prompting for Multi-step Reasoning". In: *11th International Conference on Learning Representations, ICLR 2023*.

Gao, Heting, Junrui Ni, Kaizhi Qian, Yang Zhang, Shiyu Chang, and Mark Hasegawa-Johnson (2022). "WavPrompt: Towards Few-Shot Spoken Language Understanding with Frozen Language Models". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 2738–2742. DOI: 10.21437/Interspeech.2022-11031.

Gao, Leo, Stella Biderman, et al. (2020). *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. arXiv Preprint arXiv:2101.00027.

Gao, Xiaoxue and Nancy F. Chen (2024). "Speech-Mamba: Long-Context Speech Recognition with Selective State Spaces Models". In: *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1–8. DOI: 10.1109/SLT61566.2024.10832137.

Gao, Yizhao, Zhichen Zeng, et al. (2025). *SeerAttention: Learning Intrinsic Sparse Attention in Your LLMs*. arXiv preprint arXiv:2410.13276.

Gebhard, Patrick, Tanja Schneeberger, Elisabeth André, Tobias Baur, Ionut Damian, Gregor Mehlmann, Cornelius König, and Markus Langer (2019). "Serious games for training social skills in job interviews". In: *IEEE Transactions on Games* 11.4, pp. 340–351. DOI: 10.1109/TG.2018.2808525.

Germanier, Elisabeth, Adrian Bangerter, Koralie Orji, Laetitia A. Renier, Marianne Schmid Mast, Mutian He, and Philip N. Garner (2025). "Responses to Past-Behavior Questions in Face-To-Face and Asynchronous Video Interviews: Storytelling, Interview Performance

and Criterion Validity". In: *Human Performance* 38.5, pp. 284–298. DOI: 10.1080/08959285.2025.2580653.

Germanier, Elisabeth, Adrian Bangerter, Laetitia Aurelie Renier, Emmanuelle P Kleinlogel, Marianne Schmid Mast, Dinesh Babu Jayagopi, Kumar Shubham, and Nicolas Roulin (2023). "Effects of Interview Medium and Culture on Applicant Storytelling, Disfluencies and Evaluations in Behavioral Interviews".

Germanier, Elisabeth, Mutian He, Amina Mardiyyah Rufai, Philip N. Garner, Adrian Bangerter, Laetitia A. Renier, Marianne Schmid Mast, and Koralie Orji (2025). "Identifying storytelling in job interviews using deep learning". In: *Computers in Human Behavior Reports* 19, p. 100688. DOI: 10.1016/j.chbr.2025.100688.

Gerz, Daniela, Pei-Hao Su, Razvan Kusztos, Avishek Mondal, Michał Lis, Eshan Singhal, Nikola Mrkšić, Tsung-Hsien Wen, and Ivan Vulić (2021). "Multilingual and Cross-Lingual Intent Detection from Spoken Data". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 7468–7475. DOI: 10.18653/v1/2021.emnlp-main.591.

Ghosh, Sreyan, Zhifeng Kong, Sonal Kumar, S. Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro (2025). "Audio Flamingo 2: An Audio-Language Model with Long-Audio Understanding and Expert Reasoning Abilities". In: *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*.

Ghosh, Sreyan, Sonal Kumar, Ashish Seth, Chandra Kiran Reddy Evuru, Utkarsh Tyagi, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha (2024). "GAMA: A Large Audio-Language Model with Advanced Audio Understanding and Complex Reasoning Abilities". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp. 6288–6313. DOI: 10.18653/v1/2024.emnlp-main.361.

Glorioso, Paolo, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge (2024). *Zamba: A Compact 7B SSM Hybrid Model*. arXiv preprint arXiv:2405.16712.

Goel, Arushi, Sreyan Ghosh, et al. (2025). *Audio Flamingo 3: Advancing Audio Intelligence with Fully Open Large Audio Language Models*. arXiv preprint arXiv:2507.08128.

Goel, Karan, Albert Gu, Chris Donahue, and Christopher Re (2022). "It's Raw! Audio Generation with State-Space Models". In: *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*. PMLR, pp. 7616–7633.

Grattafiori, Aaron et al. (2024). *The Llama 3 Herd of Models*. arXiv Preprint arXiv:2407.21783.

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006). "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". In: *Proceedings of the 23rd International Conference on Machine Learning, ICML 2006*. New York, NY, USA: Association for Computing Machinery, pp. 369–376. DOI: 10.1145/1143844.1143891.

Grazzi, Riccardo, Julien Siems, Jörg K. H. Franke, Arber Zela, Frank Hutter, and Massimiliano Pontil (2025). "Unlocking State-Tracking in Linear RNNs Through Negative Eigenvalues". In: *The 13th International Conference on Learning Representations, ICLR 2025*.

Green, Melanie C and Markus Appel (2024). "Narrative transportation: How stories shape how we see ourselves and the world". In: *Advances in Experimental Social Psychology* 70.1, pp. 1–82. DOI: 10.1016/bs.aesp.2024.03.002.

Gu, Albert and Tri Dao (2024). "Mamba: Linear-Time Sequence Modeling with Selective State Spaces". In: *1st Conference on Language Modeling, COLM 2024*.

Gu, Albert, Karan Goel, and Christopher Re (2022). "Efficiently Modeling Long Sequences with Structured State Spaces". In: *10th International Conference on Learning Representations, ICLR 2022*.

Gumma, Varun, Pranjal A Chitale, and Kalika Bali (2025). "Towards Inducing Long-Context Abilities in Multilingual Neural Machine Translation Models". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Albuquerque, New Mexico: Association for Computational Linguistics, pp. 7158–7170. DOI: 10.18653/v1/2025.naacl-long.366.

Haghani, Parisa, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters (2018). "From Audio to Semantics: Approaches to End-to-End Spoken Language Understanding". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 720–726. DOI: 10.1109/SLT.2018.8639043.

He, Mutian and Philip Garner (2023a). "The Interpreter Understands Your Meaning: End-to-end Spoken Language Understanding Aided by Speech Translation". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 4408–4423. DOI: 10.18653/v1/2023.findings-emnlp.291.

He, Mutian and Philip N. Garner (2023b). "Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding". In: *Proceedings of the 24th Annual Conference of the International Speech Communication Association, Interspeech 2023*, pp. 1109–1113. DOI: 10.21437/Interspeech.2023-1799.

He, Mutian and Philip N. Garner (2025). "Joint Fine-tuning and Conversion of Pretrained Speech and Language Models towards Linear Complexity". In: *The 13th International Conference on Learning Representations, ICLR 2025*.

Hemamou, Léo, Ghazi Felhi, Jean-Claude Martin, and Chloé Clavel (2019). "Slices of Attention in Asynchronous Video Job Interviews". In: *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 1–7. DOI: 10.1109/ACII.2019.8925439.

Hemamou, Léo, Ghazi Felhi, Vincent Vandenbussche, Jean-Claude Martin, and Chloé Clavel (2019). "HireNet: A Hierarchical Attention Model for the Automatic Analysis of Asynchronous Video Job Interviews". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, pp. 573–581. DOI: 10.1609/aaai.v33i01.3301573.

Hernandez, François, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève (2018). "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on

Speaker Adaptation". In: *Speech and Computer*. Cham: Springer International Publishing, pp. 198–208. DOI: 10.1007/978-3-319-99579-3_21.

Hickman, Louis, Nigel Bosch, Vincent Ng, Rachel Saef, Louis Tay, and Sang Eun Woo (2022). "Automated video interview personality assessments: Reliability, validity, and generalizability investigations". In: *Journal of Applied Psychology* 107.8, pp. 1323–1351. DOI: /10.1037/apl0000695.

Hickman, Louis, Markus Langer, Rachel M Saef, and Louis Tay (2024). "Automated Speech Recognition Bias in Personnel Selection: The Case of Automatically Scored Job Interviews". In: *Journal of Applied Psychology*. DOI: 10.1037/apl0001247.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). *Distilling the Knowledge in a Neural Network*. arXiv Preprint arXiv:1503.02531.

Hoffmann, Jordan et al. (2022). *Training Compute-Optimal Large Language Models*. arXiv preprint arXiv:2203.15556.

Holtrop, Djurre, Janneke K Oostrom, Ward R J van Breda, Antonis Koutsoumpis, and Reinout E de Vries (2022). "Exploring the application of a text-to-personality technique in job interviews". In: *European Journal of Work and Organizational Psychology* 31.6, pp. 799–816. DOI: 10.1080/1359432X.2022.2051484.

Hoque, Mohammed, Matthieu Courgeon, Jean-Claude Martin, Bilge Mutlu, and Rosalind W Picard (2013). "Mach: My automated conversation coach". In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 697–706. DOI: 10.1145/2493432.2493502.

Hori, Takaaki, Niko Moritz, Chiori Hori, and Jonathan Le Roux (2021). "Advanced Long-Context End-to-End Speech Recognition Using Context-Expanded Transformers". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 2097–2101. DOI: 10.21437/Interspeech.2021-1643.

Hou, Haowen, Zhiyi Huang, Kaifeng Tan, Rongchang Lu, and Fei Richard Yu (2025). *RWKV-X: A Linear Complexity Hybrid Language Model*. arXiv preprint arXiv:2504.21463.

Hou, Yutai, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu (2020). "Few-Shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Online: Association for Computational Linguistics, pp. 1381–1393. DOI: 10.18653/v1/2020.acl-main.128.

Hsieh, Cheng-Ping, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg (2024). "RULER: What's the Real Context Size of Your Long-Context Language Models?" In: *1st Conference on Language Modeling, COLM 2024*.

Hsu, Chan-Jan, Ho-Lam Chung, Hung-Yi Lee, and Yu Tsao (2023). "T5lephone: Bridging Speech and Text Self-Supervised Models for Spoken Language Understanding Via Phoneme Level T5". In: *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096189.

Hsu, Ming-Hao, Xueyao Zhang, Xiaohai Tian, Jun Zhang, and Zhizheng Wu (2026). *Anatomy of the Modality Gap: Dissecting the Internal States of End-to-End Speech LLMs*. arXiv preprint arXiv:2603.01502.

Hsu, Wei-Ning, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed (2021). "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 3451–3460. DOI: 10.1109/TASLP.2021.3122291.

Hu, Yuxuan, Jianchao Tan, Jiaqi Zhang, Wen Zan, Pingwei Sun, Yifan Lu, Yerui Sun, Yuchen Xie, Xunliang Cai, and Jing Zhang (2025). *Optimizing Native Sparse Attention with Latent Attention and Local Global Alternating Strategies*. arXiv preprint arXiv:2511.00819.

Huang, Zhiqi, Milind Rao, Anirudh Raju, Zhe Zhang, Bach Bui, and Chul Lee (2022). "MTL-SLT: Multi-Task Learning for Spoken Language Tasks". In: *Proceedings of the 4th Workshop on NLP for Conversational AI, NLP4ConvAI 2022*. Dublin, Ireland: Association for Computational Linguistics, pp. 120–130. DOI: 10.18653/v1/2022.nlp4convai-1.11.

Huang, Zihao, Jundong Zhou, Xingwei Qu, Qiyang Min, and Ge Zhang (2026). *ConceptMoE: Adaptive Token-to-Concept Compression for Implicit Compute Allocation*. arXiv preprint arXiv:2601.21420.

Huffcutt, Allen I, James M Conway, Philip L Roth, and Nancy J Stone (2001). "Identification and meta-analytic assessment of psychological constructs measured in employment interviews". In: *Journal of Applied Psychology* 86.5, pp. 897–913. DOI: 10.1037/0021-9010.86.5.897.

Huffcutt, Allen I and Sara A Murphy (2023). "Structured interviews: Moving beyond mean validity…" In: *Industrial and Organizational Psychology* 16.3, pp. 344–348. DOI: 10.1017/iop.2023.42.

Hwang, Sukjun, Brandon Wang, and Albert Gu (2025). "Dynamic Chunking for End-to-End Hierarchical Sequence Modeling". In.

Irie, Kazuki, Morris Yau, and Samuel J. Gershman (2025). *Blending Complementary Memory Systems in Hybrid Quadratic-Linear Transformers*. arXiv preprint arXiv:2506.00744.

Iyer, Shankar, Nikhil Dandekar, and Kornel Csernai (2017). *First Quora Dataset Release: Question Pairs*.

Jafari, Aref, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi (2021). "Annealing Knowledge Distillation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*. Online: Association for Computational Linguistics, pp. 2493–2504. DOI: 10.18653/v1/2021.eacl-main.212.

Jamba Team et al. (2024). *Jamba-1.5: Hybrid Transformer-Mamba Models at Scale*. arXiv preprint arXiv:2408.12570.

Janz, Tom (1989). "The patterned behavior description interview: The best prophet of the future is the past". In: *The employment interview: Theory, research, and practice*. Sage Publications, pp. 158–168.

Jelassi, Samy, David Brandfonbrener, Sham M. Kakade, and Eran Malach (2024). "Repeat After Me: Transformers are Better than State Space Models at Copying". In: *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*. PMLR, pp. 21502–21521.

Jiang, Huiqiang, Yucheng Li, Chengruidong Zhang, et al. (2024). "MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Jiang, Xilin, Yinghao Aaron Li, Adrian Nicolas Florea, Cong Han, and Nima Mesgarani (2024). *Speech Slytherin: Examining the Performance and Efficiency of Mamba for Speech Separation, Recognition, and Synthesis*. arXiv preprint arXiv:2407.09732.

Jo, Hyun-rae and Dongkun Shin (2024). *A2SF: Accumulative Attention Scoring with Forgetting Factor for Token Pruning in Transformer Decoder*. arXiv preprint arXiv:2407.20485.

Johnson, Melvin et al. (2017). "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation". In: *Transactions of the Association for Computational Linguistics* 5, pp. 339–351. DOI: 10.1162/tacl_a_00065.

Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy (2020). "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8, pp. 64–77. DOI: 10.1162/tacl_a_00300.

Kale, Mihir and Scott Roy (2020). "Machine Translation Pre-training for Data-to-Text Generation - A Case Study in Czech". In: *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*. Dublin, Ireland: Association for Computational Linguistics, pp. 91–96. DOI: 10.18653/v1/2020.inlg-1.13.

Kale, Mihir, Aditya Siddhant, Rami Al-Rfou, Linting Xue, Noah Constant, and Melvin Johnson (2021). "nmT5 - Is Parallel Data Still Relevant for Pre-Training Massively Multilingual Language Models?" In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 683–691. DOI: 10.18653/v1/2021.acl-short.87.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). *Scaling Laws for Neural Language Models*. arXiv Preprint arXiv:2001.08361.

Kasai, Jungo, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith (2021). "Finetuning Pretrained Transformers into RNNs". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10630–10643. DOI: 10.18653/v1/2021.emnlp-main.830.

Katharopoulos, Angelos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret (2020). "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*. PMLR, pp. 5156–5165.

Kessler, Robin (2006). *Competency-based interviews: Master the tough new interview style and give them the answers that will win you the job*. Frankilin Lakes : Career Press.

Keyu An, Shiliang Zhang (2023). *Exploring RWKV for Memory Efficient and Low Latency Streaming ASR*. arXiv Preprint arXiv:2309.14758.

Kim, Minsoo, Kyuhong Shim, Jungwook Choi, and Simyung Chang (2024). "InfiniPot: Infinite Context Processing on Memory-Constrained LLMs". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp. 16046–16060.

Kim, Seongbin, Gyuwan Kim, Seongjin Shin, and Sangmin Lee (2021). "Two-Stage Textual Knowledge Distillation for End-to-End Spoken Language Understanding". In: *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7463–7467. DOI: 10.1109/ICASSP39728.2021.9414619.

Kimi Team et al. (2025a). *Kimi Linear: An Expressive, Efficient Attention Architecture*. arXiv preprint arXiv:2510.26692.

KimiTeam et al. (2025b). *Kimi-Audio Technical Report*. arXiv preprint arXiv:2504.18425.

Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv Preprint arXiv:1412.6980.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *the National Academy of Sciences* 114.13, pp. 3521–3526.

Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya (2020). "Reformer: The Efficient Transformer". In: *The 8th International Conference on Learning Representations, ICLR 2020*.

Koluguri, Nithin Rao, Samuel Kriman, Georgy Zelenfroind, Somshubra Majumdar, Dima Rekesh, Vahid Noroozi, Jagadeesh Balam, and Boris Ginsburg (2024). "Investigating End-to-End ASR Architectures for Long Form Audio Transcription". In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13366–13370. DOI: 10.1109/ICASSP48485.2024.10448309.

Koutsoumpis, Antonis, Sina Ghassemi, Janneke K Oostrom, Djurre Holtrop, Ward van Breda, Tianyi Zhang, and Reinout E de Vries (2024). "Beyond traditional interviews: Psychometric analysis of asynchronous video interviews for personality and interview performance Evaluation using machine learning". In: *Computers in Human Behavior* 154, p. 108128. DOI: 10.1016/j.chb.2023.108128.

Kung, T.H. et al. (2023). "Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models". In: *PLOS Digital Health* 2.2, e0000198.

Labov, William and Joshua Waletzky (1997). "Narrative analysis: Oral versions of personal experience." In: *Journal of Narrative and Life History* 7, pp. 3–38.

Lan, Disen, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng (2025). "Liger: Linearizing Large Language Models to Gated Recurrent Structures". In: *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*. PMLR, pp. 32452–32466.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *8th International Conference on Learning Representations, ICLR 2020*.

Łańcucki, Adrian, Konrad Staniszewski, Piotr Nawrot, and Edoardo M. Ponti (2025). *Inference-Time Hyper-Scaling with KV Cache Compression*. arXiv preprint arXiv:2506.05345.

Langer, Markus, Cornelius J König, Patrick Gebhard, and Elisabeth André (2016). "Dear computer, teach me manners: Testing virtual employment interview training". In: *International Journal of Selection and Assessment* 24.4, pp. 312–323. DOI: 10.1111/ijsa.12150.

Lee, Chia-Hsuan, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee (2018). "Spoken SQuAD: A Study of Mitigating the Impact of Speech Recognition Errors on Listening Comprehension". In: *Proceedings of the 19th Annual Conference of the International Speech Communication Association, Interspeech 2018*. ISCA, pp. 3459–3463. DOI: 10.21437/Interspeech.2018-1714.

Lee, Heejun, Geon Park, Youngwan Lee, Jaduk Suh, Jina Kim, Wonyong Jeong, Bumsik Kim, Hyemin Lee, Myeongjae Jeon, and Sung Ju Hwang (2025). "A Training-Free Sub-quadratic Cost Transformer Model Serving Framework with Hierarchically Pruned Attention". In: *13th International Conference on Learning Representations, ICLR 2025*.

Lenz, Barak et al. (2025). "Jamba: Hybrid Transformer-Mamba Language Models". In: *13th International Conference on Learning Representations, ICLR 2025*.

Lester, Brian, Rami Al-Rfou, and Noah Constant (2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Online: Association for Computational Linguistics, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703.

Li, Jiaqi, Yao Qian, Yuxuan Hu, Leying Zhang, Xiaofei Wang, Heng Lu, Manthan Thakker, Jinyu Li, Sheng Zhao, and Zhizheng Wu (2025). *FlexiCodec: A Dynamic Neural Audio Codec for Low Frame Rates*. arXiv preprint arXiv:2510.00981.

Li, Kai, Guo Chen, Runxuan Yang, and Xiaolin Hu (2024). *SPMamba: State-space Model Is All You Need in Speech Separation*. arXiv Preprint arXiv:2404.02063.

Li, Xian, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli (2021). *Multilingual Speech Translation with Efficient Finetuning of Pretrained Models*. arXiv Preprint arXiv:2010.12829.

Li, Xingjian, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan (2019). "DELTA: Deep Learning Transfer Using Feature Map with Attention for Convolutional Networks". In: *The 7th International Conference on Learning Representations, ICLR 2019*.

Li, Xuhong, Yves Grandvalet, and Franck Davoine (2018). "Explicit Inductive Bias for Transfer Learning with Convolutional Networks". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. PMLR, pp. 2825–2834.

Li, Yixiao, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao (2023). "LoSparse: Structured Compression of Large Language Models Based on Low-Rank and Sparse Approximation". In: *Proceedings of the 40th International Conference on Machine Learning, ICML 2023*. PMLR, pp. 20336–20350.

Li, Yuhong, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen (2024). "SnapKV: LLM Knows What You are Looking

for Before Generation". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Liang, Wanchao et al. (2024). "TorchTitan: One-stop PyTorch native solution for production ready LLM pretraining". In.

Liao, Hank (2013). "Speaker Adaptation of Context Dependent Deep Neural Networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7947–7951. DOI: 10.1109/ICASSP.2013.6639212.

Liem, Cynthia CS, Markus Langer, Andrew Demetriou, Annemarie MF Hiemstra, Achmadnoer Sukma Wicaksana, Marise Ph Born, and Cornelius J König (2018). "Psychology meets machine learning: Interdisciplinary perspectives on algorithmic job candidate screening". In: *Explainable and interpretable models in computer vision and machine learning*. Springer, pp. 197–253. DOI: 10.1007/978-3-319-98131-4_9.

Liff, Josh, Nathan Mondragon, Cari Gardner, Christopher J Hartwell, and Adam Bradshaw (2024). "Psychometric properties of automated video interview competency assessments". In: *Journal of Applied Psychology*. DOI: 10.1037/apl0001173.

Lin, Chin-Yew (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: ACL, pp. 74–81.

Lin, Guan-Ting, Yung-Sung Chuang, Ho-Lam Chung, Shu-wen Yang, Hsuan-Jui Chen, Shuyan Annie Dong, Shang-Wen Li, Abdelrahman Mohamed, Hung-yi Lee, and Lin-shan Lee (2022). "DUAL: Discrete Spoken Unit Adaptive Learning for Textless Spoken Question Answering". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 5165–5169. DOI: 10.21437/Interspeech.2022-612.

Lin, Tzu-Quan, Heng-Cheng Kuo, Tzu-Chieh Wei, Hsi-Chun Cheng, Chun-Wei Chen, Hsien-Fu Hsiao, Yu Tsao, and Hung-yi Lee (2025). *An Exploration of Mamba for Speech Self-Supervised Models*. arXiv preprint arXiv:2506.12606.

Lin, Xi, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang (2023). "Continuation Path Learning for Homotopy Optimization". In: *Proceedings of the 40th International Conference on Machine Learning, ICML 2023*. PMLR, pp. 21288–21311.

Lin, Yueqian, Yuzhe Fu, Jingyang Zhang, Yudong Liu, Jianyi Zhang, Jingwei Sun, Hai Helen Li, and Yiran Chen (2025). "SpeechPrune: Context-Aware Token Pruning for Speech Information Retrieval". In: *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. DOI: 10.1109/ICME59968.2025.11209113.

Ling Team et al. (2025). *Every Attention Matters: An Efficient Hybrid Architecture for Long-Context Reasoning*. arXiv preprint arXiv:2510.19338.

Liu, Alexander H., Andy Ehrenberg, et al. (2025). *Voxtral*. arXiv preprint arXiv:2507.13264.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2023). "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9, 195:1–195:35. DOI: 10.1145/3560815.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer (2020). "Multilingual Denoising Pre-training for Neural Ma-

chine Translation". In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742. DOI: 10.1162/tacl_a_00343.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv Preprint arXiv:1907.11692.

Liu, Zichang, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava (2023). "Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Lozhkov, Anton (2022). *Hugging Face: anton-l/xtreme_s_xlsr_300m_minds14*.

Lu, Enzhe, Zhejun Jiang, et al. (2025). *MoBA: Mixture of Block Attention for Long-Context LLMs*. arXiv preprint arXiv:2502.13189.

Lu, Yi, Xin Zhou, Wei He, Jun Zhao, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang (2024). "LongHeads: Multi-Head Attention is Secretly a Long Context Processor". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp. 7136–7148.

Lu, Yichao, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun (2018). "A Neural Interlingua for Multilingual Machine Translation". In: *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018*. Brussels, Belgium: Association for Computational Linguistics, pp. 84–92. DOI: 10.18653/v1/W18-6309.

Lufkin, Leon, Tomás Figliolia, Beren Millidge, and Kamesh Krishnamurthy (2026). *Hybrid Associative Memories*. arXiv preprint arXiv:2603.22325.

Lukacik, Eden-Raye, Joshua S Bourdage, and Nicolas Roulin (2022). "Into the void: A conceptual model and research agenda for the design and use of asynchronous video interviews". In: *Human Resource Management Review* 32.1, pp. 1–15. DOI: 10.1016/j.hrmr.2020.100789.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts (2011). "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL-HLT 2011*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 142–150.

MacKay, David J. C. (1992). "A Practical Bayesian Framework for Backpropagation Networks". In: *Neural Computation* 4.3, pp. 448–472. DOI: 10.1162/neco.1992.4.3.448.

Maddox, Wesley J, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson (2019). "A Simple Baseline for Bayesian Uncertainty in Deep Learning". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems, NeurIPS 2019*. Vol. 32. Curran Associates, Inc.

Mallinson, Jonathan, Rico Sennrich, and Mirella Lapata (2020). "Zero-Shot Crosslingual Sentence Simplification". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Online: Association for Computational Linguistics, pp. 5109–5126. DOI: 10.18653/v1/2020.emnlp-main.415.

Mao, Huanru Henry (2022). "Fine-Tuning Pre-trained Transformers into Decaying Fast Weights". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 10236–10242. DOI: 10.18653/v1/2022.emnlp-main.697.

McAuliffe, Michael, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger (2017). "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi". In: pp. 498–502. DOI: 10.21437/Interspeech.2017-1386.

McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher (2017). "Learned in Translation: Contextualized Word Vectors". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017*. Vol. 30. Curran Associates, Inc.

McDermott, Luke, Robert W. Heath Jr, and Rahul Parhi (2025). *LoLA: Low-Rank Linear Attention With Sparse Caching*. arXiv preprint arXiv:2505.23666.

McKenzie, Ian et al. (2022). *The Inverse Scaling Prize*.

Meng, Yen, Hsuan-Jui Chen, Jiatong Shi, Shinji Watanabe, Paola Garcia, Hung-yi Lee, and Hao Tang (2023). "On Compressing Sequences for Self-Supervised Speech Models". In: *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1128–1135. DOI: 10.1109/SLT54892.2023.10022991.

Mercat, Jean, Igor Vasiljevic, Sedrick Scott Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and Thomas Kollar (2024). "Linearizing Large Language Models". In: *1st Conference on Language Modeling, COLM 2024*.

Miceli Barone, Antonio Valerio, Barry Haddow, Ulrich Germann, and Rico Sennrich (2017). "Regularization Techniques for Fine-Tuning in Neural Machine Translation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1489–1494. DOI: 10.18653/v1/D17-1156.

MiniCPM Team et al. (2026). *MiniCPM-SALA: Hybridizing Sparse and Linear Attention for Efficient Long-Context Modeling*. arXiv preprint arXiv:2602.11761.

MiniMax et al. (2025). *MiniMax-01: Scaling Foundation Models with Lightning Attention*. arXiv preprint arXiv:2501.08313.

Miyazaki, Koichi, Yoshiki Masuyama, and Masato Murata (2024). "Exploring the Capability of Mamba in Speech Applications". In: *Proceedings of the 25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pp. 237–241. DOI: 10.21437/Interspeech.2024-994.

Miyazaki, Koichi, Masato Murata, and Tomoki Koriyama (2023). "Structured State Space Decoder for Speech Recognition and Synthesis". In: *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096135.

Mohtashami, Amirkeivan and Martin Jaggi (2023). "Random-Access Infinite Context Length for Transformers". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Nagrani, Arsha, Joon Son Chung, Weidi Xie, and Andrew Zisserman (2020). "Voxceleb: Large-scale Speaker Verification in the Wild". In: *Computer Speech & Language* 60, p. 101027. DOI: 10.1016/j.csl.2019.101027.

Naim, Iftekhar, M Iftekhar Tanveer, Daniel Gildea, and Mohammed Ehsan Hoque (2015). "Automated prediction and analysis of job interview performance: The role of what you say and how you say it". In: *11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. Vol. 1, pp. 1–6. DOI: 10.1109/FG.2015.7163127.

Nakano, Reiichiro et al. (2022). *WebGPT: Browser-assisted Question-Answering with Human Feedback*. arXiv Preprint arXiv:2112.09332.

Nguyen, Chien Van, Huy Huu Nguyen, et al. (2024). *Taipan: Efficient and Expressive State Space Language Models with Selective Attention*. arXiv preprint arXiv:2410.18572.

Nguyen, Laurent Son, Denise Frauendorfer, Marianne Schmid Mast, and Daniel Gatica-Perez (2014). "Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior". In: *IEEE Transactions on Multimedia* 16.4, pp. 1018–1031. DOI: 10.1109/TMM.2014.2307169.

Nunez, Elvis, Luca Zancato, Benjamin Bowman, Aditya Golatkar, Wei Xia, and Stefano Soatto (2025). "Expansion Span: Combining Fading Memory and Retrieval in Hybrid State Space Models". In: *Proceedings of the International Conference on Neuro-symbolic Systems*. PMLR, pp. 570–596.

Oren, Matanel, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz (2024). "Transformers are Multi-State RNNs". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp. 18724–18741.

Orji, K., A. Bangerter., E. Germanier, L. A. Renier, Schmid Mast, M. M. He, and P. N. Garner (2024). "Extended Responses in Asynchronous Video Interviews: Investigating Frequency, Content, and Interview Outcomes [Manuscript in preparation]".

Ouyang, Long et al. (2022). *Training Language Models to Follow Instructions with Human Feedback*. arXiv Preprint arXiv:2203.02155.

Pagliardini, Matteo, Daniele Paliotta, Martin Jaggi, and François Fleuret (2023). "Fast Attention Over Long Sequences With Dynamic Sparse Flash Attention". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Parcollet, Titouan, Rogier van Dalen, Shucong Zhang, and Sourav Bhattacharya (2024). "SummaryMixing: A Linear-Complexity Alternative to Self-Attention for Speech Recognition and Understanding". In: *Proceedings of the 25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pp. 3460–3464. DOI: 10.21437/Interspeech.2024-40.

Park, Se Jin, Julian Salazar, Aren Jansen, Keisuke Kinoshita, Yong Man Ro, and R. J. Skerry-Ryan (2024). *Long-Form Speech Generation with Spoken Language Models*. arXiv preprint arXiv:2412.18603.

Pascanu, Razvan and Yoshua Bengio (2014). *Revisiting Natural Gradient for Deep Networks*. arXiv Preprint arXiv:1301.3584.

Penedo, Guilherme, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf (2024). "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*.

Peng, Baolin, Chenguang Zhu, Michael Zeng, and Jianfeng Gao (2021). "Data Augmentation for Spoken Language Understanding via Pretrained Language Models". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 1219–1223. DOI: 10.21437/Interspeech.2021-117.

Peng, Bo, Eric Alcaide, et al. (2023). "RWKV: Reinventing RNNs for the Transformer Era". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 14048–14077.

Pennebaker, James W, Ryan L Boyd, Kayla Jordan, and Kate Blackburn (2015). *The development and psychometric properties of LIWC2015*. Manual. Austin, TX: The University of Texas at Austin.

Pepino, Leonardo, Pablo Riera, and Luciana Ferrer (2021). "Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 3400–3404. DOI: 10.21437/Interspeech.2021-703.

Piękos, Piotr, Róbert Csordás, and Jürgen Schmidhuber (2025). *Mixture of Sparse Attention: Content-Based Learnable Sparse Attention via Expert-Choice Routing*. arXiv preprint arXiv:2505.00315.

Poliak, Adam, Yonatan Belinkov, James Glass, and Benjamin Van Durme (2018). "On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), NAACL-HLT 2018*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 513–523. DOI: 10.18653/v1/N18-2082.

Pratap, Vineel et al. (2024). "Scaling Speech Technology to 1,000+ Languages". In: *Journal of Machine Learning Research* 25.97, pp. 1–52.

Qin, Chengwei, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang (2023). "Is ChatGPT a General-Purpose Natural Language Processing Task Solver?" In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*. Singapore: Association for Computational Linguistics, pp. 1339–1384. DOI: 10.18653/v1/2023.emnlp-main.85.

Qin, Libo, Tianbao Xie, Wanxiang Che, and Ting Liu (2021). "A Survey on Spoken Language Understanding: Recent Advances and New Frontiers". In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021*. Vol. 5, pp. 4577–4584. DOI: 10.24963/ijcai.2021/622.

Qin, Ziran, Yuchen Cao, Mingbao Lin, Wen Hu, Shixuan Fan, Ke Cheng, Weiyao Lin, and Jianguo Li (2024). "CAKE: Cascading and Adaptive KV Cache Eviction with Layer Preferences". In: *13th International Conference on Learning Representations, ICLR 2025*.

Qu, Xingwei et al. (2026). *Dynamic Large Concept Models: Latent Reasoning in an Adaptive Semantic Space*. arXiv preprint arXiv:2512.24617.

Qwen Team (2025). *Qwen3-Next: Towards Ultimate Training & Inference Efficiency*.

Qwen Team (2026). *Qwen3.5: Towards Native Multimodal Agents*.

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever (2023). "Robust Speech Recognition via Large-Scale Weak Supervision". In: *Proceedings of the 40th International Conference on Machine Learning, ICML 2023*. PMLR, pp. 28492–28518.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). *Language Models are Unsupervised Multitask Learners*. RT RN. Open AI.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv Preprint arXiv:1910.10683.

Raganato, Alessandro, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann (2019). "An Evaluation of Language-Agnostic Inner-Attention-Based Representations in Machine Translation". In: *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP 2019*. Florence, Italy: Association for Computational Linguistics, pp. 27–32. DOI: 10.18653/v1/W19-4304.

Rahman, Wasifur, Sazan Mahbub, Asif Salekin, Md Kamrul Hasan, and Ehsan Hoque (2021). "HirePreter: A Framework for Providing Fine-grained Interpretation for Automated Job Interview Analysis". In: *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 1–5. DOI: 10.1109/ACIIW52867.2021.9666201.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*. Austin, Texas: Association for Computational Linguistics, pp. 2383–2392. DOI: 10.18653/v1/D16-1264.

Raju, Anirudh, Milind Rao, Gautam Tiwari, Pranav Dheram, Bryan Anderson, Zhe Zhang, Chul Lee, Bach Bui, and Ariya Rastrow (2022). "On Joint Training with Interfaces for Spoken Language Understanding". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 1253–1257. DOI: 10.21437/Interspeech.2022-11067.

Ralston, Steven M, William G Kirkwood, and Patricia A Burant (2003). "Helping interviewees tell their stories". In: *Business Communication Quarterly* 66.3, pp. 8–22. DOI: 10.1177/108056990306600303.

Rasmussen, Keith G (1984). "Nonverbal behavior, verbal behavior, resumé credentials, and selection interview outcomes". In: *Journal of Applied Psychology* 69.4, pp. 551–556. DOI: 10.1037/0021-9010.69.4.551.

Reid, Machel and Mikel Artetxe (2022). "PARADISE: Exploiting Parallel Data for Multilingual Sequence-to-Sequence Pretraining". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2022. Seattle, United States: Association for Computational Linguistics, pp. 800–810. DOI: 10.18653/v1/2022.naacl-main.58.

Rekesh, Dima et al. (2023). "Fast Conformer With Linearly Scalable Attention For Efficient Speech Recognition". In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. DOI: 10.1109/ASRU57964.2023.10389701.

Ren, Liliang, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen (2024). "Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling". In: *The 13th International Conference on Learning Representations, ICLR 2025*.

Ren, Liliang, Yang Liu, Shuohang Wang, Yichong Xu, Chenguang Zhu, and ChengXiang Zhai (2023). "Sparse Modular Activation for Efficient Sequence Modeling". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Richter, Manuela, Cornelius J König, Christopher Koppermann, and Michael Schilling (2016). "Displaying fairness while delivering bad news: Testing the effectiveness of organizational bad news training in the layoff context". In: *Journal of Applied Psychology* 101.6, pp. 779–792. DOI: 10.1037/apl0000087.

Riggio, Ronald E and Barbara Throckmorton (1988). "The relative effects of verbal and nonverbal behavior, appearance, and social skills on evaluations made in hiring interviews 1". In: *Journal of Applied Social Psychology* 18.4, pp. 331–348. DOI: 0.1111/j.1559-1816.1988.tb00020.x.

Roulin, Nicolas, Le Koi Anh Pham, and Joshua S Bourdage (2023). "Ready? Camera rolling... Action! Examining interviewee training and practice opportunities in asynchronous video interviews". In: *Journal of Vocational Behavior* 145, p. 103912. DOI: 10.1016/j.jvb.2023.103912.

Roulin, Nicolas, Odelia Wong, Markus Langer, and Joshua S Bourdage (2023). "Is more always better? How preparation time and re-recording opportunities impact fairness, anxiety, impression management, and performance in asynchronous video interviews". In: *European Journal of Work and Organizational Psychology* 32.3, pp. 333–345. DOI: 10.1080/1359432X.2022.2156862.

Roy, Aurko, Mohammad Saffar, Ashish Vaswani, and David Grangier (2021). "Efficient Content-Based Sparse Attention with Routing Transformers". In: *Transactions of the Association for Computational Linguistics* 9, pp. 53–68.

Rupasinghe, Atigallage Tharuka, Nadeesha Lakmini Gunawardena, S Shujan, and Das Atukorale (2016). "Scaling personality traits of interviewees in an online job interview by vocal spectrum and facial cue analysis". In: *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, pp. 288–295. DOI: 10.1109/ICTER.2016.7829933.

Rush, Alexander M., Sumit Chopra, and Jason Weston (2015). "A Neural Attention Model for Abstractive Sentence Summarization". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Lisbon, Portugal: Association for Computational Linguistics, pp. 379–389. DOI: 10.18653/v1/D15-1044.

Salesky, Elizabeth, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post (2021). "The Multilingual TEDx Corpus for Speech Recognition and Translation". In: *Proceedings of the 22nd Annual Conference of the In-*

ternational Speech Communication Association, Interspeech 2021, pp. 3655–3659. DOI: 10.21437/Interspeech.2021-11.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2020). *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. arXiv Preprint arXiv:1910.01108.

Saxon, Michael, Samridhi Choudhary, Joseph P. McKenna, and Athanasios Mouchtaris (2021). "End-to-End Spoken Language Understanding for Generalized Voice Assistants". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 4738–4742. DOI: 10.21437/Interspeech.2021-1826.

Schick, Timo, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom (2023). "Toolformer: Language Models Can Teach Themselves to Use Tools". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Schlag, Imanol, Kazuki Irie, and Jürgen Schmidhuber (2021). "Linear Transformers Are Secretly Fast Weight Programmers". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*. PMLR, pp. 9355–9366.

Schuster, Sebastian, Sonal Gupta, Rushin Shah, and Mike Lewis (2019). "Cross-Lingual Transfer Learning for Multilingual Task Oriented Dialog". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), NAACL-HLT 2019*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3795–3805. DOI: 10.18653/v1/N19-1380.

Schwenk, Holger and Matthijs Douze (2017). "Learning Joint Multilingual Sentence Representations with Neural Machine Translation". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP, RepL4NLP 2017*. Vancouver, Canada: Association for Computational Linguistics, pp. 157–167. DOI: 10.18653/v1/W17-2619.

Senellart, Jean, Dakun Zhang, Bo Wang, Guillaume Klein, Jean-Pierre Ramatchandirin, Josep Crego, and Alexander Rush (2018). "OpenNMT System Description for WNMT 2018: 800 Words/Sec on a Single-Core CPU". In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NGT 2018*. Melbourne, Australia: Association for Computational Linguistics, pp. 122–128. DOI: 10.18653/v1/W18-2715.

Seo, Seunghyun, Donghyun Kwak, and Bowon Lee (2022). "Integration of Pre-Trained Networks with Continuous Token Interface for End-to-End Spoken Language Understanding". In: *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7152–7156. DOI: 10.1109/ICASSP43922.2022.9747047.

Serdyuk, Dmitriy, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio (2018). "Towards End-to-end Spoken Language Understanding". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5754–5758. DOI: 10.1109/ICASSP.2018.8461785.

Shakhadri, Syed Abdul Gaffar, Kruthika KR, and Kartik Basavaraj Angadi (2025). *Samba-ASR: State-Of-The-Art Speech Recognition Leveraging Structured State-Space Models*. arXiv preprint arXiv:2501.02832.

Shi, Jingze, Yifan Wu, Bingheng Wu, Yiran Peng, Liangdong Wang, Guang Liu, and Yuyu Luo (2025). *Trainable Dynamic Mask Sparse Attention*. arXiv preprint arXiv:2508.02124.

Shi, Xing, Inkit Padhi, and Kevin Knight (2016). "Does String-Based Neural MT Learn Source Syntax?" In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*. Austin, Texas: Association for Computational Linguistics, pp. 1526–1534. DOI: 10.18653/v1/D16-1159.

Shwartz-Ziv, Ravid, Micah Goldblum, Hossein Souri, Sanyam Kapoor, Chen Zhu, Yann LeCun, and Andrew G. Wilson (2022). "Pre-Train Your Loss: Easy Bayesian Transfer Learning with Informative Priors". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems, NeurIPS 2022*. Vol. 35, pp. 27706–27715.

Si, Chenglei, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang (2023). "Prompting GPT-3 To Be Reliable". In: *11th International Conference on Learning Representations, ICLR 2023*.

Siddhant, Aditya, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman (2020). "Evaluating the Cross-Lingual Effectiveness of Massively Multilingual Neural Machine Translation". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*. Vol. 34, pp. 8854–8861. DOI: 10.1609/aaai.v34i05.6414.

Siems, Julien, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi (2025). "DeltaProduct: Improving State-Tracking in Linear RNNs via Householder Products". In: *Proceedings of the 39th International Conference on Neural Information Processing Systems, NeurIPS 2025*.

Silvester, Joanne (1997). "Spoken attributions and candidate success in graduate recruitment interviews". In: *Journal of Occupational and Organizational Psychology* 70.1, pp. 61–73. DOI: 10.1111/j.2044-8325.1997.tb00631.x.

Simon, Christian et al. (2026). *Echoes Over Time: Unlocking Length Generalization in Video-to-Audio Generation Models*. arXiv preprint arXiv:2602.20981.

Singhania, Prajwal, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele (2024). "Loki: Low-rank Keys for Efficient Sparse Attention". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642.

Song, Feifan, Lianzhe Huang, and Houfeng Wang (2024). "A Unified Framework for Multi-Intent Spoken Language Understanding with Prompting". In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9966–9970. DOI: 10.1109/ICASSP48485.2024.10447804.

Srivastava, Aarohi et al. (2023). "Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models". In: *Transactions on Machine Learning Research*.

Stevens, Cynthia Kay and Amy L Kristof (1995). "Making the right impression: A field study of applicant impression management during job interviews". In: *Journal of Applied Psychology* 80.5, pp. 587–606. DOI: 10.1037/0021-9010.80.5.587.

Su, Jianlin, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu (2024). "RoFormer: Enhanced transformer with Rotary Position Embedding". In: *Neurocomputing* 568, p. 127063.

Suen, Hung-Yue, Kuo-En Hung, and Chien-Liang Lin (2019). "TensorFlow-Based Automatic Personality Recognition Used in Asynchronous Video Interviews". In: *IEEE Access* 7, pp. 61018–61023. DOI: 10.1109/ACCESS.2019.2902863.

Sun, Siqi, Yu Cheng, Zhe Gan, and Jingjing Liu (2019). "Patient Knowledge Distillation for BERT Model Compression". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), EMNLP-IJCNLP 2019*. Hong Kong, China: Association for Computational Linguistics, pp. 4323–4332. DOI: 10.18653/v1/D19-1441.

Sun, Yutao, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei (2023). *Retentive Network: A Successor to Transformer for Large Language Models*. arXiv preprint arXiv:2307.08621.

Takase, Sho and Naoaki Okazaki (2022). "Multi-Task Learning for Cross-Lingual Abstractive Summarization". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022*. Marseille, France: European Language Resources Association, pp. 3008–3016.

Tan, Xin, Yuetao Chen, Yimin Jiang, Xing Chen, Kun Yan, Nan Duan, Yibo Zhu, Daxin Jiang, and Hong Xu (2025). *DSV: Exploiting Dynamic Sparsity to Accelerate Large-Scale Video DiT Training*. arXiv preprint arXiv:2502.07590.

Tang, Changli, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang (2024). "SALMONN: Towards Generic Hearing Abilities for Large Language Models". In: *The 12th International Conference on Learning Representations, ICLR 2024*.

Tang, Jiaming, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han (2024). "QUEST: Query-Aware Sparsity for Efficient Long-Context LLM Inference". In: *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*. PMLR, pp. 47901–47911.

Tang, Raphael, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin (2019). *Distilling Task-Specific Knowledge from BERT into Simple Neural Networks*. arXiv Preprint arXiv:1903.12136.

Tay, Yi, Mostafa Dehghani, Dara Bahri, and Donald Metzler (2022). "Efficient Transformers: A Survey". In: *ACM Comput. Surv.* 55.6, 109:1–109:28. DOI: 10.1145/3530811.

Tencent Hunyuan Team et al. (2025). *Hunyuan-TurboS: Advancing Large Language Models through Mamba-Transformer Synergy and Adaptive Chain-of-Thought*. arXiv preprint arXiv:2505.15431.

van der Goot, Rob, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank (2021). "From Masked Language Modeling to Translation: Non-English Auxiliary Tasks Improve Zero-

shot Spoken Language Understanding". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*. Online: Association for Computational Linguistics, pp. 2479–2497. DOI: 10.18653/v1/2021.naacl-main.197.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017*. Vol. 30. Curran Associates, Inc.

Vázquez, Raúl, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz (2019). "Multilingual NMT with a Language-Independent Attention Bridge". In: *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP 2019*. Florence, Italy: Association for Computational Linguistics, pp. 33–39. DOI: 10.18653/v1/W19-4305.

Villatoro-Tello, Esaú, Srikanth Madikeri, Juan Zuluaga-Gomez, Bidisha Sharma, Seyyed Saeed Sarfjoo, Iuliia Nigmatulina, Petr Motlicek, Alexei V. Ivanov, and Aravind Ganapathiraju (2023). "Effectiveness of Text, Acoustic, and Lattice-Based Representations in Spoken Language Understanding Tasks". In: *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095168.

Waleffe, Roger et al. (2024). *An Empirical Study of Mamba-based Language Models*. arXiv preprint arXiv:2406.07887.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, EMNLP 2018*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355. DOI: 10.18653/v1/W18-5446.

Wang, Changhan, Anne Wu, Jiatao Gu, and Juan Pino (2021). "CoVoST 2 and Massively Multilingual Speech Translation". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 2247–2251. DOI: 10.21437/Interspeech.2021-2027.

Wang, Dustin, Rui-Jie Zhu, Steven Abreu, et al. (2025). *A Systematic Analysis of Hybrid Linear Attention*. arXiv preprint arXiv:2507.06457.

Wang, Hankun, Yiwei Guo, Chongtian Shao, Bohan Li, Xie Chen, and Kai Yu (2025). *CodecSlime: Temporal Redundancy Compression of Neural Speech Codec via Dynamic Frame Rate*. arXiv preprint arXiv:2506.21074.

Wang, Hanrui, Zhekai Zhang, and Song Han (2021). "SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning". In: *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 97–110.

Wang, Hualei, Yiming Li, Shuo Ma, Hong Liu, and Xiangdong Wang (2026). "Listening Between the Frames: Bridging Temporal Gaps in Large Audio-Language Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 40. 31, pp. 26233–26241. DOI: 10.1609/aaai.v40i31.39827.

Wang, Junxiong, Daniele Paliotta, Avner May, Alexander M Rush, and Tri Dao (2024). "The Mamba in the Llama: Distilling and Accelerating Hybrid Models". In: *Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*.

Wang, Sinong, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma (2020). *Linformer: Self-Attention with Linear Complexity*. arXiv Preprint arXiv:2006.04768.

Wang, Xiong, Sining Sun, Lei Xie, and Long Ma (2021). "Efficient Conformer with Prob-Sparse Attention Mechanism for End-to-End Speech Recognition". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 4578–4582. DOI: 10.21437/Interspeech.2021-415.

Wang, Yingzhi, Abdelmoumene Boumadane, and Abdelwahab Heba (2022). *A Fine-tuned Wav2vec 2.0/HuBERT Benchmark For Speech Emotion Recognition, Speaker Verification and Spoken Language Understanding*. arXiv Preprint arXiv:2111.02735.

Wang, Ziteng, Jun Zhu, and Jianfei Chen (2024). "ReMoE: Fully Differentiable Mixture-of-Experts with ReLU Routing". In: *13th International Conference on Learning Representations, ICLR 2025*.

Wei, Jason, Yi Tay, et al. (2022). "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research*.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems, NeurIPS 2022*.

Wen, Kaiyue, Xingyu Dang, and Kaifeng Lyu (2025). "RNNs are not Transformers (Yet): The Key Bottleneck on In-Context Retrieval". In: *13th International Conference on Learning Representations, ICLR 2025*.

Wu, Ting-Wei, Ruolin Su, and Biing Juang (2021). "A Label-Aware BERT Attention Network for Zero-Shot Multi-Intent Detection in Spoken Language Understanding". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4884–4896. DOI: 10.18653/v1/2021.emnlp-main.399.

Wu, Wenhao, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu (2024). "Retrieval Head Mechanistically Explains Long-Context Factuality". In: *The Thirteenth International Conference on Learning Representations*.

Wu, Yangjun, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang (2022). "Incorporating Instructional Prompts into a Unified Generative Framework for Joint Multiple Intent Detection and Slot Filling". In: *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 7203–7208.

Xiang, Bajian, Tingwei Guo, Xuan Chen, and Yang Han (2026). *Do We Need Distinct Representations for Every Speech Token? Unveiling and Exploiting Redundancy in Large Speech Language Models*. DOI: 10.48550/arXiv.2604.06871.

Xiao, Chaojun, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun (2024). "InfLLM: Training-Free Long-Context Extrapolation for

LLMs with an Efficient Context Memory". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Xiao, Guangxuan (2025). *Why Stacking Sliding Windows Can't See Very Far*.

Xiao, Guangxuan, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis (2023). "Efficient Streaming Language Models with Attention Sinks". In: *The 12th International Conference on Learning Representations, ICLR 2024*.

Xu, Ruochen, Chenguang Zhu, Yu Shi, Michael Zeng, and Xuedong Huang (2020). "Mixed-Lingual Pre-training for Cross-lingual Summarization". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL 2020*. Suzhou, China: Association for Computational Linguistics, pp. 536–541. DOI: 10.18653/v1/2020.aacl-main.53.

Yadav, Sarthak and Zheng-Hua Tan (2024). "Audio Mamba: Selective State Spaces for Self-Supervised Audio Representations". In: *Proceedings of the 25th Annual Conference of the International Speech Communication Association, Interspeech 2024*, pp. 552–556. DOI: 10.21437/Interspeech.2024-1274.

Yang, Dongjie, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao (2024). "PyramidInfer: Pyramid KV Cache Compression for High-throughput LLM Inference". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp. 3258–3270.

Yang, Mingyu, Mehdi Rezagholizadeh, Guihong Li, Vikram Appia, and Emad Barsoum (2025). *Zebra-Llama: Towards Extremely Efficient Hybrid Models*. arXiv preprint arXiv:2505.17272.

Yang, Shu-wen, Po-Han Chi, et al. (2021). "SUPERB: Speech Processing Universal PERformance Benchmark". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, Interspeech 2021*, pp. 1194–1198. DOI: 10.21437/Interspeech.2021-1775.

Yang, Songlin, Jan Kautz, and Ali Hatamizadeh (2024). "Gated Delta Networks: Improving Mamba2 with Delta Rule". In: *The 13th International Conference on Learning Representations, ICLR 2025*.

Yang, Songlin, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim (2024). "Gated Linear Attention Transformers with Hardware-Efficient Training". In: *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*. PMLR, pp. 56501–56523.

Yang, Songlin, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim (2024). "Parallelizing Linear Transformers with the Delta Rule over Sequence Length". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems, NeurIPS 2024*.

Yang, Songlin and Yu Zhang (2024). *FLA: A Triton-Based Library for Hardware-Efficient Implementations of Linear Attention Mechanism*.

Yazdani, Majid and James Henderson (2015). "A Model of Zero-Shot Learning of Spoken Language Understanding". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Lisbon, Portugal: Association for Computational Linguistics, pp. 244–249. DOI: 10.18653/v1/D15-1027.

Yu, Wenhao, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang (2023). "Generate Rather than Retrieve: Large Language Models Are Strong Context Generators". In: *11th International Conference on Learning Representations, ICLR 2023*.

Yuan, Jingyang et al. (2025). "Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*. Vienna, Austria: Association for Computational Linguistics, pp. 23078–23097.

Zaheer, Manzil et al. (2020). "Big Bird: Transformers for Longer Sequences". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NeurIPS 2020*. Vol. 33. Curran Associates, Inc., pp. 17283–17297.

Zeng, Zihao, Bokai Lin, Tianqi Hou, Hao Zhang, and Zhijie Deng (2024). *In-context KV-Cache Eviction for LLMs via Attention-Gate*. arXiv preprint arXiv:2410.12876.

Zhan, Zhihao, Jianan Zhao, Zhaocheng Zhu, and Jian Tang (2025). "Overcoming Long Context Limitations of State Space Models via Context Dependent Sparse Attention". In: *Proceedings of the 39th International Conference on Neural Information Processing Systems, NeurIPS 2025*.

Zhang, Biao, Ivan Titov, Barry Haddow, and Rico Sennrich (2021). "Beyond Sentence-Level End-to-End Speech Translation: Context Helps". In: *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), ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 2566–2578. DOI: 10.18653/v1/2021.acl-long.200.

Zhang, Jintao, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen (2025). "SpargeAttention: Accurate and Training-free Sparse Attention Accelerating Any Model Inference". In: *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*.

Zhang, Michael, Kush Bhatia, Hermann Kumbong, and Christopher Re (2024). "The Hedgehog & the Porcupine: Expressive Linear Attentions with Softmax Mimicry". In: *The 12th International Conference on Learning Representations, ICLR 2024*.

Zhang, Susan, Stephen Roller, et al. (2022). *OPT: Open Pre-trained Transformer Language Models*. arXiv Preprint arXiv:2205.01068.

Zhang, Xiangyu, Jianbo Ma, Mostafa Shahin, Beena Ahmed, and Julien Epps (2025). "Rethinking Mamba in Speech Processing by Self-Supervised Models". In: *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10889111.

Zhang, Xiangyu, Qiquan Zhang, Hexin Liu, Tianyi Xiao, Xinyuan Qian, Beena Ahmed, Eliathamby Ambikairajah, Haizhou Li, and Julien Epps (2024). *Mamba in Speech: Towards an Alternative to Self-Attention*. arXiv Preprint arXiv:2405.12609.

Zhang, Yu and Songlin Yang (2025). *Flame: Flash Language Modeling Made Easy*.

Zhang, Zhenyu, Ying Sheng, et al. (2023). "H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems, NeurIPS 2023*.

Zhao, Weilin, Zihan Zhou, et al. (2025). *InfLLM-V2: Dense-Sparse Switchable Attention for Seamless Short-to-Long Adaptation*. arXiv preprint arXiv:2509.24663.

Zhao, Youpeng, Di Wu, and Jun Wang (2024). "ALISA: Accelerating Large Language Model Inference via Sparsity-Aware KV Caching". In: *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 1005–1017.

Zheng, Rui-Chen, Wenrui Liu, Hui-Peng Du, Qinglin Zhang, Chong Deng, Qian Chen, Wen Wang, Yang Ai, and Zhen-Hua Ling (2025). *Say More with Less: Variable-Frame-Rate Speech Tokenization via Adaptive Clustering and Implicit Duration Coding*. arXiv preprint arXiv:2509.04685.

Zhou, Xiabin, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding (2025). *DynamicKV: Task-Aware Adaptive KV Cache Compression for Long Context LLMs*. arXiv preprint arXiv:2412.14838.

Zhu, Han, Li Wang, Gaofeng Cheng, Jindong Wang, Pengyuan Zhang, and Yonghong Yan (2022). "Wav2vec-S: Semi-Supervised Pre-Training for Low-Resource ASR". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*, pp. 4870–4874. DOI: 10.21437/Interspeech.2022-909.

Zhu, Junnan, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong (2019). "NCLS: Neural Cross-Lingual Summarization". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), EMNLP-IJCNLP 2019*. Hong Kong, China: Association for Computational Linguistics, pp. 3054–3064. DOI: 10.18653/v1/D19-1302.

Zuo, Jialong, Guangyan Zhang, Minghui Fang, Shengpeng Ji, Xiaoqi Jiao, Jingyu Li, Yiwen Guo, and Zhou Zhao (2025). *Entropy-based Coarse and Compressed Semantic Speech Representation Learning*. arXiv preprint arXiv:2509.00503.

Zuo, Simiao, Xiaodong Liu, Jian Jiao, Denis X. Charles, Eren Manavoglu, Tuo Zhao, and Jianfeng Gao (2024). "Efficient Hybrid Long Sequence Modeling with State Space Augmented Transformers". In: *1st Conference on Language Modeling, COLM 2024*.

MUTIAN HE

hemutiann@gmail.com | Homepage: mutiann.github.io | Google Scholar | GitHub

Research Profile: I specialize in hardware-aware efficient architectures (linear/sparse/hybrid attention) to enable long-form language and speech modeling with practical speedups.

EDUCATION

Idiap Research Institute, École Polytechnique Fédérale de Lausanne (EPFL) | PhD
Martigny, Switzerland | Advised by Philip Garner | Jun 2022 –

The Hong Kong University of Science and Technology | MPhil
Hong Kong SAR | Advised by Yangqiu Song | Sep 2019 – Jun 2022 | GPA 4.03/4.3

Beihang University (BUAA) | B.Eng. in Computer Science
Beijing | Sep 2015 – Jun 2019 | GPA 3.8/4.0 (top 5%)

University of Toronto | Exchange Student
Toronto | 2017 | GPA 4.0/4.0

RESEARCH EXPERIENCE

Idiap Research Institute, EPFL | PhD Student & Research Assistant | Jun 2022 –
Explored speech and language processing with foundation models & LLMs, with a focus on **hardware-aware efficient (linear/sparse/hybrid) attention**, **LLM architectures**, and **long-form modeling**.

• Led the development and writing of the successful SwissAI project grant proposal *Long-form Speech Foundation Models with Hierarchical Dynamic Chunking* based on linear and hybrid attention, currently working on the project.

• Pretrained LLMs using a hybrid of linear and sparse attention based on contextualized learnable token eviction to improve long-context retrieval while preserving the time and space efficiency thanks to hardware-aware design and Triton implementation.

• Enabled conversion of pretrained speech and language transformers to linear attention through layerwise distillation using only downstream task data, recovering transformer performance without re-pretraining.

• Improved semantic understanding and few-shot cross-lingual transfer for spoken language understanding models using speech translation for auxiliary mid-training, combined with Bayesian continual learning regularization.

• Benchmarked large language models for spoken language understanding in in-context learning settings and revealed their failure modes.

• Contributed to projects on automatic analysis of job interview recordings, including storytelling detection and interview response analysis.

Speech Output Scientists, Microsoft | Intern | Jul 2018 – Aug 2019; Aug 2020 – Oct 2021
Conducted research on end-to-end **neural text-to-speech** systems, with emphasis on **robustness** and **scalability to new languages**.

• Developed robust end-to-end TTS methods that reduced output failures and mis-pronunciations on out-of-domain inputs using stepwise monotonic attention and ASR-TTS dual learning.

• Architected a multilingual TTS generalist model covering 40+ languages, enabling adaptation to a new language with less than one minute of data and no language-specific G2P pipeline, and explored its internal mechanism.

• Built a fully end-to-end (G2P-free) TTS model that reads textual lexicon entries of characters to improve pronunciation control and polyphone disambiguation in non-phonemic scripts such as Chinese.

HKUST KnowComp Group | MPhil Student | Sept 2019 – Jun 2022

Investigated the role of **conceptualization** in **commonsense knowledge graphs**.

- Crowdsourced a large-scale dataset for concept abstraction and instantiation under language context, and built a machine conceptual induction framework upon natural language, resulting in improved commonsense reasoning capabilities.
- Contributed to projects on subgraph isomorphism counting and time-evolving text classification.

SELECTED PUBLICATIONS & PREPRINTS

Mutian He, Philip N. Garner. Alleviating Forgetfulness of Linear Attention by Hybrid Sparse Attention and Contextualized Learnable Token Eviction. *arXiv:2510.20787. Presented on ICLR Workshop MemAgents.* [Code]

Mutian He, Philip N. Garner. Joint Fine-tuning and Conversion of Pretrained Speech and Language Models towards Linear Complexity. *ICLR 2025.* [Code]

Elisabeth Germanier, **Mutian He**, Amina Mardiyyah Rufai, Philip N. Garner, Adrian Bangerter, Laetitia A Renier, Marianne Schmid Mast, Koralie Orji. Identifying Storytelling in Job Interviews using Deep Learning. *Computers in Human Behavior Reports (Journal, 2025).*

Mutian He, Tianqing Fang, Weiqi Wang, Yangqiu Song. Acquiring and Modelling Abstract Commonsense Knowledge via Conceptualization. *Artificial Intelligence (Journal, 2024).* [Code & Data]

Mutian He, Philip N. Garner. The Interpreter Understands Your Meaning: End-to-end Spoken Language Understanding Aided by Speech Translation. *Findings of EMNLP 2023.* [Code & Data]

Mutian He, Philip N. Garner. Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding. *Interspeech 2023 (Oral).*

Mutian He, Jingzhou Yang, Lei He, Frank K. Soong. Neural Lexicon Reader: Reduce Pronunciation Errors in End-to-end TTS by Leveraging External Textual Knowledge. *Interspeech 2022.* [Code]

Mutian He, Jingzhou Yang, Lei He, Frank K. Soong. Multilingual Byte2Speech Models for Scalable Low-resource Speech Synthesis. *arXiv:2103.03541.* [Code]

Mutian He, Yangqiu Song, Kun Xu, Dong Yu. On the Role of Conceptualization in Commonsense Knowledge Graph Construction. *arXiv:2003.03239.* [Code]

Xin Liu, Haojie Pan, **Mutian He**, Yangqiu Song, Xin Jiang. Neural Subgraph Isomorphism Counting. *KDD 2020.*

Mutian He, Yan Deng, Lei He. Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS. *Interspeech 2019 (Oral).* [Demo]

Yu He, Jianxin Li, Yangqiu Song, **Mutian He**, Hao Peng. Time-evolving Text Data Classification with Deep Neural Networks. *IJCAI/ECAI 2018.*

SKILLS

Research Focus: efficient attention (linear, sparse, and hybrid attention), LLM architecture, long-context modeling, speech processing, spoken language understanding

Systems & Engineering: Python, PyTorch, TensorFlow, Hugging Face, Triton

Machine Learning: optimization, generalization, representation learning, continual learning, data collection, model training, kernel development, model evaluation

Languages: Mandarin Chinese (native), English (advanced)

TEACHING & SERVICE

TA of COMP4221 Intro to Natural Language Processing, HKUST, Spring 2020
TA of M06 Intro to Speech Processing, Idiap, Fall 2022 & 2023
Reviewer: ACL (ARR), IEEE SPL, ICASSP, ICLR, NeurIPS, COLM