Context-Driven Dynamic Pruning for Large Speech Foundation Models

Masao Someki[1], *Shikhar Bharadwaj*[1], *Atharva Anand Joshi*[1], *Chyi-Jiunn Lin*[1], *Jinchuan Tian*[1], *Jee-weon Jung*[†1], *Markus Müller*[2], *Nathan Susanj*[2], *Jing Liu*[2], *Shinji Watanabe*[1]

[1]Language Technologies Institute, Carnegie Mellon University, USA
[2]Neural Efficiency Science, Amazon Artificial General Intelligence, USA

msomeki@andrew.cmu.edu

Abstract

Speech foundation models achieve strong generalization across languages and acoustic conditions, but require significant computational resources for inference. In the context of speech foundation models, pruning techniques have been studied that dynamically optimize model structures based on the target audio leveraging external context. In this work, we extend this line of research and propose context-driven dynamic pruning, a technique that optimizes the model computation depending on the context between different input frames and additional context during inference. We employ the Open Whisper-style Speech Model (OWSM) and incorporate speaker embeddings, acoustic event embeddings, and language information as additional context. By incorporating the speaker embedding, our method achieves a reduction of 56.7 GFLOPs while improving BLEU scores by a relative 25.7% compared to the fully fine-tuned OWSM model.

Index Terms: Pruning, Dynamic Pruning, Speech Foundation Model, Speech to Text, Speech Recognition

1. Introduction

Training large neural networks on large-scale speech datasets has achieved significant success in various speech-related tasks [1–4]. These speech foundation models demonstrate strong generalization ability and robustness across languages [3], speakers, and acoustic conditions [1]. While these large models achieve high performance, the use of large-scale models with billions of parameters requires substantial computational resources. Several approaches have been proposed to mitigate these challenges, including knowledge distillation [5–7], quantization [8–10], pruning [11–15], or combination of these techniques [10, 16].

Among these lines of work, recent research has focused on pruning techniques that dynamically adjust model computation during inference, including frame skipping or temporal pruning [17, 18], early exit [19–21], and structured dynamic pruning for large speech models [15, 22]. Specifically, Someki et al [15] proposed a dynamic pruning method for speech foundation models that leverages speech features, task characteristics, and language to adapt the model architecture during inference. They introduced a Global Gate Predictor (globalGP) that generates pruning masks from the input speech and applied dynamic pruning during inference. However, since this method determines pruning masks for all modules based on input, it cannot account for layer-wise output variations across utterances.

Based on these trends, this study proposes a pruning mechanism named the Local Gate Predictor (localGP), which applies pruning decisions at each layer independently, in contrast to globalGP, which applies a uniform pruning mask across all layers. We designed localGP to adaptively generate pruning masks based on context that varies across inferences. We enhance encoder pruning by conditioning on external contexts including speaker embeddings (ECAPA_TDNN [23]), acoustic event feature (BEATs [24]), and language information (URIEL lang2vec [25]), We further propose training the localGP at the token level, allowing different modules to be used depending on token history, and analyze the improvements in performance and pruning characteristics. Since localGP utilizes external context at each layer, it is more effective at incorporating contextual information compared to globalGP, which applies pruning decisions uniformly without considering the internal computations of the model. Our contributions can be summarized as follows:

- We propose localGP, a model designed to utilize various external contexts that dynamically change during inference.
- We enhance encoder pruning by integrating speaker embeddings and acoustic events, reducing 56.7 GFLOPs while boosting BLEU by 25.7% and preserving ASR performance comparable to the fully fine-tuned OWSM model.
- Analyzing pruning patterns in Section 4.4 and Section 4.5, we found that encoder-side pruning behaves similarly to a voice activity detection (VAD) system, while decoder-side pruning shows distinct patterns for specific token types.

2. Preliminaries and Related Works

2.1. Related Works

Pruning techniques for speech processing models have been extensively studied [11, 14, 26, 27]. However, in these approaches, the pruning mask is determined either during or after training and remains fixed during inference. As a result, the sparse pattern is *static* regardless of the input data during inference. Therefore, while such methods improve computational efficiency, they do not account for contextual information, which may prevent the model from achieving optimal performance. To mitigate this limitation, recent research has explored approaches that *dynamically* adjust the model structure during inference [15, 19, 21, 22, 28, 29], enabling more adaptive and context-aware pruning. Following this trend, our study further advances pruning methods that dynamically modify the model during inference and explores their applicability. Furthermore, we propose localGP that leverages pretrained models to extract context information, enabling dynamic pruning conditioned on a more diverse set of contextual factors during inference.

Previous research has explored pruning input data for models [30]. This approach has been adopted in many speech processing models to shorten input length. In recent years, dynamic pruning techniques have also been developed in this do-

[†]Currently at Apple.

Algorithm 1 Local Gate Predictor for layer i

Require: $C_{\text{key}} \in \mathbb{R}^{T \times D^C \times D}$, $C_{\text{value}} \in \mathbb{R}^{T \times D^C \times D}$
Require: $x^i \in \mathbb{R}^{T \times 1 \times D}$ ▷ x_i is used as a query.
  gates $\leftarrow$ []
  $a^i \leftarrow \text{Softmax}(x^i \times C_{\text{key}}^\top, \text{dim} = -1)$ ▷ $\mathbb{R}^{T \times 1 \times D^C}$
  $a^i \leftarrow a^i \times C_{\text{value}} + x^i$ ▷ $\mathbb{R}^{T \times 1 \times D}$
  for $n = 1$ **to** N **do**
    $p_n \leftarrow t(\text{Linear}_n(a^i), \text{dim} = -1)$ ▷ $\mathbb{R}^{T \times 1 \times 2}$
    gates.append($p_n[0]$)
  end for
  $C_{\text{key}} \leftarrow \texttt{concat}((C_{\text{key}}, x^i), \text{dim} = 1)$
  $C_{\text{value}} \leftarrow \texttt{concat}((C_{\text{value}}, \text{Linear}(x^i)), \text{dim} = 1)$
  return gates, C_{key}, C_{value}

main [18, 31]. However, these methods either provide frames reduced by a fixed ratio regardless of speech information or perform frame pruning using only speech input. In this study, we leverage external context captured by localGP to apply optimal input data pruning to each module within the model (e.g., self-attention). We refer to this approach as *temporal pruning*, as it dynamically subsamples frames based on contextual information rather than pruning entire module computations for all frames in an utterance. Additionally, we compare its performance with *utterance-wise pruning*, which removes computations at the utterance level, and discuss the results in Section 3.2.

2.2. Dynamic Pruning

Let z be the pruning mask, θ the model parameters, $f(\cdot)$ the neural network subject to pruning, and x the input to the model. The output $\tilde{y}$ of the pruned model is computed as $\tilde{y} = f(x; \theta \odot z), \quad z \in \{0,1\}^{|\theta|}$, where $|\theta|$ represents the number of parameters, and $\odot$ denotes element-wise multiplication. Typically, z is generated as a binary mask by applying a threshold to probabilities computed by another neural network g. For context-aware dynamic pruning, function g takes the context C as its input. However, performing such discretization during training results in non-differentiability, causing the gradient to become zero and making parameter updates difficult.

To address this issue, Someki et al [15] proposed using the Straight-through Gumbel-softmax Estimator (SGSE) [32], which enables the pruning mask to remain binary during training, thereby ensuring consistency between training and inference computations. Specifically, letting $t(\cdot)$ denote the SGSE function, the output $\tilde{y}$ is computed as $\tilde{y} = f(x; \theta \odot t(g(\cdot)))$.

Since $t(\cdot)$ is based on the softmax function, it cannot compute a probability for a single value like the sigmoid function. To overcome this, the output of $g(\cdot)$ is formulated as a two-class classification problem, and the probability for one class is used as the pruning mask. During inference, a similar computation is performed using the softmax function, where a threshold is applied to the probability of the class representing the pruning mask to determine the final binary pruning mask. This approach ensures that the pruning decision follows the same computational flow in both training and inference, maintaining consistency and stability in the pruning process.

2.3. Global Gate Predictor.

In globalGP [15], the parameter set θ encompasses the modules within both the encoder and decoder, with the pruning probabilities for all modules computed simultaneously. Specifically, the set of encoder modules is defined as $\mathcal{M}_{\text{enc}} = \{\text{self-attn}^i, \text{cgMLP}^i, ...\}$, comprising the feed forward network (FFN), self-attention (self-attn) and MLP with convolutional gating (cgMLP) [33] in the i-th E-Branchformer [34] layers. Here, the superscript represents the layer index. Similarly, the decoder module set is defined as $\mathcal{M}_{\text{dec}} = \{\text{self-attn}^i, \text{src-attn}^i, ...\}$, which includes FFN, self-attn, and source attention (src-attn) mechanisms. Then the outputs x_{enc}^i and x_{dec}^i from one of the modules in i-th layer $\mathtt{m}_{\text{enc}}^i \in \mathcal{M}_{\text{enc}}$ and $\mathtt{m}_{\text{dec}}^i \in \mathcal{M}_{\text{dec}}$ of encoder and decoder networks, respectively, are computed as follows:

$$z_{\text{enc}}^i = t(g(x, C)), \qquad z_{\text{dec}}^i = t(g(x, C)), \tag{1}$$

$$x_{\text{enc}}^i = \mathtt{m}_{\text{enc}}^i(x_{\text{enc}}^{i-1}, z_{\text{enc}}^i), \qquad x_{\text{dec}}^i = \mathtt{m}_{\text{dec}}^i(x_{\text{dec}}^{i-1}, z_{\text{dec}}^i). \tag{2}$$

Here, z_{enc}^i and z_{dec}^i serve as pruning masks for the modules in the i-th layer of the encoder and decoder, respectively. The function $g(x)$ outputs the pruning probabilities for all modules simultaneously.

3. Proposed Method

3.1. Local Gate Predictor

In localGP, the pruning mask z is computed separately for each layer. Consequently, Equation 1 is modified as follows:

$$z_{\text{enc}}^i = t(g(x_{\text{enc}}^{i-1}, \hat{C})), \qquad z_{\text{dec}}^i = t(g(x_{\text{dec}}^{i-1}, \hat{C})). \tag{3}$$

In globalGP, the context C represents discrete identifiers such as language ID or task ID, often implemented as special tokens. However, in this work, we introduce $\hat{C}$ to represent a richer set of contextual information, such as speaker embeddings and acoustic event information. LocalGP generates a separate pruning mask for each layer, enabling adaptive pruning customized to the characteristics of individual layers. To enhance contextual awareness, we modify the input by incorporating x^i, enabling each layer to access and leverage information accumulated from previous layers.

Algorithm 1 details the process, where D is the dimensionality of the context, T is the sequence length, D^C is the number of contexts used, and N is the number of modules inside the i-th layer. We use C as C_{key} and apply a linear transformation to C_{value}, which forms key-value pairs. The input feature x^i to the i-th layer is used as the query. Regarding D^C, for example, when using speaker embeddings and acoustic event embeddings, we set $D^C = 2$. Cross-attention is computed with a residual connection, and z is derived from t. A linear layer is applied to compute the logits. Finally, the results are added to C_{key} and C_{value}.

3.2. Temporal pruning vs. utterance-wise pruning

Let $\mathtt{m}_n \in \mathcal{M}_{\text{enc}} \cup \mathcal{M}_{\text{dec}}$ be a module, $\tilde{y}_n$, x_n, and z_n be the input, output, and the pruning mask of the module $\mathtt{m}_n$, respectively. Then, $\tilde{y}_n$ in temporal pruning and utterance-wise pruning becomes:

$$\tilde{y}_n = \begin{cases} \texttt{pad}(\mathtt{m}(\mathtt{h}(x_n, z_n)), z_n), & \text{if pruning by frame,} \\ x_n * 0, & \text{if pruning by utt, } z_n = 0, \\ \mathtt{m}(x_n), & \text{if pruning by utt, } z_n = 1. \end{cases} \tag{4}$$

The $\mathtt{h}$ function selects non-skipped frames, while the $\texttt{pad}$ function zero-pads the output tensor based on the pruning mask. If the number of indices selected by $\mathtt{h}$ is smaller than the convolution kernel size in cgMLP, computation becomes infeasible. To address this, the cgMLP module is always computed for all frames in our work.

Table 1: *WER and BLEU scores for ASR and ST in German (de), French (fr), and Italian (it) with utterance-wise pruning (globalGP) and temporal pruning (localGP) strategies on OWSM-v3.1. The* `Context` *column denotes the additional context used for pruning.* `Enc+Dec` *indicates that pruning is applied to both encoder and decoder modules, while* `front` *refers to subsampled speech features from the frontend.* `spk` *and* `event` *indicate that pruning is guided by speaker embeddings and acoustic event information, respectively.*

No.	Pruned Module	Is localGP?	Context	ASR-WER (↓)				ST (↑)							GFLOPs (Enc)
				de	fr	it	Average	de-fr	de-it	fr-de	fr-it	it-de	it-fr	Average	
1	-		full fine-tuning (baseline)	13.6	11.0	13.2	12.6	8.4	6.4	11.2	13.0	11.8	13.5	10.7	568.5
Utterance-wise pruning (GlobalGP)															
2	Encoder		front (baseline)	15.0	11.9	15.2	14.0	5.8	6.0	8.6	12.3	6.5	12.6	8.6	452.5
3	Decoder		front (baseline)	15.2	11.9	12.9	13.3	9.8	8.4	12.2	13.1	11.3	13.4	11.4	-
4	Enc+Dec		front (baseline)	15.0	12.6	14.6	14.1	7.3	4.7	9.9	9.5	8.7	11.5	8.6	-
Temporal pruning (LocalGP)															
5	Encoder	✓	front	14.6	11.4	13.0	13.0	10.4	7.8	13.0	14.6	11.2	15.3	12.0	541.6
6	Encoder	✓	+ spk	14.5	**11.1**	**12.7**	12.8	**12.0**	**9.2**	**14.4**	**15.6**	**12.7**	**16.9**	**13.5**	511.8
7	Encoder	✓	+ event	**14.1**	**11.1**	13.0	**12.7**	11.4	8.2	13.5	15.1	12.0	16.3	12.8	510.1
8	Encoder	✓	+ spk + event	15.2	12.4	14.5	14.0	10.6	8.1	13.6	15.0	12.0	16.3	12.6	497.0
9	Decoder	✓	lang2vec	15.0	11.2	**12.7**	13.0	11.2	8.4	13.7	15.2	12.3	16.4	12.9	-
10	Enc+Dec	✓	front + event + lang2vec	14.5	11.0	13.1	12.9	10.5	8.0	13.1	14.9	11.9	16.0	12.4	-

4. Experiments

4.1. Models.

In this study, we employed version 3.1 of OWSM model [4], an open-source alternative to OpenAI's Whisper [1]. We chose OWSM-v3.1 for its openness and reproducibility; unlike Whisper, it is trained entirely on publicly available data. This allows us to ensure that our evaluation set was not included in pretraining, enabling a fair assessment of pruning effectiveness. To the best of our knowledge, no other public encoder-decoder model with attention-CTC is available for such comparison.

4.2. Experimental Setups.

We used version 1.1 of the Europarl-ST [35] corpus to evaluate our method and the baseline. We selected French, German, and Italian among the 9 languages in this corpus to ensure a language-balanced training data. Each language contains approximately 20 hours of speech data. With this dataset, we fine-tuned all models with a pruning objective across ASR and speech translation (ST) tasks. We used a batch size of 4, Adam optimizer, Warmup LR Scheduler with learning rate of $1e^{-5}$ and 6000 warmup steps. In all experiments, we followed [15] and used OWSM-v3.1 as the backbone model with a target sparsity of 30%, meaning that 70% of the model's modules or frames remained active during training and inference.

To better capture audio information for pruning on the encoder, and language information for decoder, we utilized the SOTA models for each context: ECAPA_TDNN [23] for speaker embedding; BEATs [24] for acoustic events information; and URIEL lang2vec [25] for language information. We specifically use BEATs' second layer because we want to keep the context extraction module small. When the dimensionalities of speaker embeddings and acoustic event embeddings differ from D^C, we use a linear layer to align them. Similarly, if the sequence lengths vary, we either duplicate the final frame or trim the sequence to match the length of the audio.

We evaluate ASR performance with Word Error Rate (WER) and ST using BLEU scores, while measuring encoder GFLOPs to assess the impact of pruning and additional context. Since OWSM uses autoregressive decoding with fixed 30-second input speech, external factors like hypothesis count and end-detection mechanisms are controlled. To isolate the effect of pruning strategies and additional context, we specifically measure the encoder GFLOPs rather than the entire model. This allows us to clearly visualize the computational differences introduced by temporal pruning, utterance-wise pruning, and contextual inputs. A beam size of 5 was used for evaluation.

4.3. Results

In Table 1, we compare row 2 and 5 to evaluate the effect of temporal pruning. Temporal pruning with LocalGP leads to a significant performance improvement in both ASR and ST, achieving an average relative improvement of 39.6% in BLEU score. Additionally, ASR exhibits an average performance gain of 7.3% in WER. In terms of GFLOPs reduction, temporal pruning does not achieve the same level of reduction as globalGP by comparing the row 2 and 5 to 8. This is likely because, in utterance-wise pruning, the computations for pruned modules are entirely skipped, whereas in temporal pruning, all modules are still computed. Nevertheless, simply applying our proposed method effectively reduces 26.9 GFLOPs from the topline OWSM-v3.1 model while achieving a 25.7% relative improvement in BLEU, demonstrating its efficiency in pruning. We also measured the averaged encoder-side wall-clock time: the model with speaker embedding (row 6) took 0.124s, slightly higher than globalGP's 0.111s (row 2).

Next, we examine the characteristics of different acoustic features by comparing rows 5 through 8. Overall, the addition of speaker embeddings in row 6 led to a significant performance improvement. Compared to the baseline in row 2, ASR achieved an average relative WER reduction of 9.0%, while translation tasks exhibited an average relative BLEU score improvement of 56.0%. A similar trend was observed with acoustic event information (row 7), where ST achieved an average relative BLEU score gain of 47.7%. These results suggest that rather than directly using subsampled speech features, leveraging a pre-trained model to extract rich contextual information enables more effective selection of frames for pruning. For ST, incorporating speaker embeddings significantly outperformed the fully fine-tuned OWSM-v3.1 model, achieving a relative BLEU score improvement of 28.6%. Speaker embeddings and acoustic event features also showed a similar trend in GFLOPs reduction, achieving reductions of 56.7 GFLOPs and 58.4 GFLOPs, respectively. These findings demonstrate that our proposed method improves both inference flops and the performance of the OWSM-v3.1 encoder.

Comparing decoder results in Table 1, rows 4 and 10, we find that temporal pruning also improves performance. How-

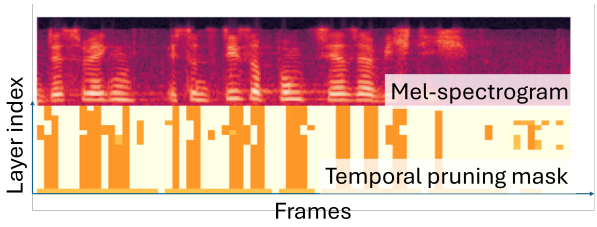


Figure 1: *Log-Mel spectrogram (top) and temporal pruning mask for self-attention modules (bottom). The y-axis of the lower plot represents the layers, with the initial layer at the bottom. The x-axis of both plots represents the time scale. The orange regions in the lower plot indicate activated self-attention modules, while the white regions represent pruned modules.*

Table 2: *Statistical test results on source-attention usage*

Test	Test statistic	p-value
Mann-Whitney U test	5.38×10^8	< 0.0001
Welch's t-test	70.9	< 0.0001

ever, due to batch-wise beam search, different pruning masks were applied to different beams. As a result, most decoder modules were still computed, negating expected inference speed gains. A tailored implementation could theoretically accelerate decoder-side pruning as well.

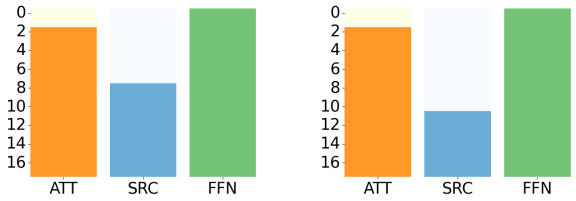
4.4. Temporal Pruning Pattern Analysis for Encoder

We visualized how temporal pruning is performed when using the speaker embedding model as a context. Figure 1 illustrates an example from the test set, showing the log-mel spectrogram and the pruning mask of the self-attention module. The figure reveals that frames are actively utilized during speech segments, while relatively fewer computations are allocated to silence regions. Notably, the first layer attends to almost all frames, whereas the deeper layers exhibit more pronounced pruning patterns, suggesting that the initial layer has acquired a VAD-like function. However, unlike conventional VAD systems, certain layers, particularly in the middle to later stages, exhibit non-negligible attention to silence frames, suggesting that these layers encode silence-related features. This indicates that the model selectively activates frames in layers responsible for capturing silence-related information, achieving optimal temporal pruning at both the layer and module levels.

As discussed in Section 4.3, temporal pruning on the encoder side appears to have a similar effect to VAD system. This suggests that adding speaker embeddings may have allowed the model to respond more sensitively to the speaker's voice activity. However, when combining these two feature types in row 9, no significant performance improvement was observed. Although GFLOPs were lower than those of the two individual models, considering the additional computation required for two pre-trained models, this approach cannot be regarded as an effective reduction in computational cost. Furthermore, performance was comparable to or even worse than that of row 8. These findings indicate that multiple audio context features does not necessarily lead to performance improvement, and it is sufficient to select a single model that with the best results.

4.5. Analysis on Decoder result

We also focused on the decoder side to examine how pruning is performed for each token. As shown in Figure 2, we found that



(a) *Token:* `[Space]wollen` (b) *Token:* `struktur`

Figure 2: *Pruning pattern for tokens* `[Space]wollen` *and* `struktur`*. The* `ATT`*,* `SRC`*, and* `FFN` *represent self-attention, source-attention, and the feed-forward network inside the decoder, respectively. The y-axis indicates the layers, with the top representing the first layer. Colored modules indicate activated modules. The number of* `SRC` *modules is higher when a space precedes a token.*

the usage rate of source-attention differs depending on whether a token begins with a space, meaning that the token is the start of a new word. To verify this, we analyzed the relationship between the sparsity ratio of source-attention and the presence of spaces in the tokens generated in the German ASR test set. Among all output tokens, 27,032 contained space, while 30,667 did not. We conducted statistical tests to assess this difference, as summarized in Table 2. The results indicate a significant difference in sparsity ratio between tokens with and without spaces. Specifically, tokens starting a new word require more attention to the encoder output, indicating that the model relies more on audio information for these tokens. This suggests that the model dynamically adjusts its reliance on specific modules based on token traits, which could have implications for optimizing decoding efficiency. Furthermore, the tendency to suppress source-attention of early decoder layers is consistent with (Someki et al., 2025). We hypothesize that localGP amplifies this effect by utilizing the required amount of source attention for each tokens, reducing encoder-induced noise during decoding and acting as a regularizer, particularly improving ST.

5. Conclusion

In this study, we proposed localGP, a context-driven dynamic inference optimization method that integrates external context, including speaker embeddings, acoustic events, and linguistic information. Our experiments demonstrated that combining localGP with temporal pruning and speaker embeddings as additional context reduced computation by 56.7 GFLOPs compared to the original OWSM-v3.1. Additionally, we achieved ST performance exceeded that of the fine-tuned baseline OWSM-v3.1, with BLEU score relatively improvements of 25.6% on average. Furthermore, our analysis of the pruning masks showed that the first layer of the OWSM encoder acquired a VAD-like function, confirming that our method dynamically optimizes model computation based on input speech characteristics.

6. Acknowledgement

Experiments of this work used the Bridges2 system at PSC and Delta system at NCSA through allocations CIS210014 and IRI120008P from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, supported by National Science Foundation grants #2138259,#:2138286, #:2138307, #:2137603, and #:2138296.

7. References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023.

[2] K. C. Puvvada, P. Żelasko, H. Huang, O. Hrinchuk, N. R. Koluguri, K. Dhawan, S. Majumdar, E. Rastorgueva, Z. Chen, V. Lavrukhin, J. Balam, and B. Ginsburg, "Less is more: Accurate speech recognition & translation without web-scale data," in *Interspeech 2024*, 2024.

[3] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling speech technology to 1,000+ languages," *arXiv preprint*, vol. 2305.13516, 2023.

[4] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J. weon Jung, and S. Watanabe, "Owsm v3.1: Better and faster open whisper-style speech models based on e-branchformer," in *Interspeech 2024*, 2024.

[5] S. Gandhi, P. von Platen, and A. M. Rush, "Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling," 2023.

[6] H.-J. Chang, S.-w. Yang, and H.-y. Lee, "Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[7] H.-J. Chang, N. Dong, R. Mavlyutov, S. Popuri, and Y.-A. Chung, "Colld: Contrastive layer-to-layer distillation for compressing multilingual pre-trained speech encoders," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[8] A. Fasoli, C.-Y. Chen, M. Serrano, X. Sun, N. Wang, S. Venkataramani, G. Saon, X. Cui, B. Kingsbury, W. Zhang, Z. Tüske, and K. Gopalakrishnan, "4-bit quantization of lstm-based speech recognition models," in *Interspeech 2021*, 2021.

[9] O. Rybakov, P. Meadowlark, S. Ding, D. Qiu, J. Li, D. Rim, and Y. He, "2-bit conformer quantization for automatic speech recognition," in *Interspeech 2023*, 2023.

[10] S. Ding, D. Qiu, D. Rim, Y. He, O. Rybakov, B. Li, R. Prabhavalkar, W. Wang, T. N. Sainath, Z. Han *et al.*, "Usm-lite: Quantization and sparsity aware fine-tuning for speech recognition with universal speech models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[11] Z. Aojun, M. Yukun, Z. Junnan, L. Jianbo, Z. Zhijie, Y. Kun, S. Wenxiu, and L. Hongsheng, "Learning N: M fine-grained structured sparse neural networks from scratch," *International Conference on Learning Representations (ICLR)*, 2021.

[12] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l_0 regularization," in *The International Conference on Learning Representations (ICLR)*, 2018.

[13] C.-I. J. Lai, Y. Zhang, A. H. Liu, S. Chang, Y.-L. Liao, Y.-S. Chuang, K. Qian, S. Khurana, D. Cox, and J. Glass, "Parp: Prune, adjust and re-prune for self-supervised speech recognition," *Advances in Neural Information Processing Systems*, 2021.

[14] Y. Peng, K. Kim, F. Wu, P. Sridhar, and S. Watanabe, "Structured pruning of self-supervised pre-trained models for speech recognition and understanding," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[15] M. Someki, Y. Peng, S. Arora, M. Müller, A. Mouchtaris, G. Strimel, J. Liu, and S. Watanabe, "Context-aware dynamic pruning for speech foundation models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[16] Y. Peng, Y. Sudo, M. Shakeel, and S. Watanabe, "Dphubert: Joint distillation and pruning of self-supervised speech models," in *24th Annual Conference of the International Speech Communication Association, Interspeech 2023, Dublin, Ireland, August 20-24, 2023*, 2023.

[17] Q. Tang, B. Zhang, J. Liu, F. Liu, and Y. Liu, "Dynamic token pruning in plain vision transformers for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[18] S. Zhang, E. Loweimi, Y. Xu, P. Bell, and S. Renals, "Trainable dynamic subsampling for end-to-end speech recognition," in *Interspeech 2019*, 2019.

[19] A. Graves, "Adaptive computation time for recurrent neural networks," *arXiv preprint*, vol. 1603.08983, 2016.

[20] D. Berrebbi, B. Yan, and S. Watanabe, "Avoid overthinking in self-supervised models for speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*, 2023.

[21] H. Xu, F. Jia, S. Majumdar, H. Huang, S. Watanabe, and B. Ginsburg, "Efficient sequence transduction by jointly predicting tokens and durations," in *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[22] Y. Peng, J. Lee, and S. Watanabe, "I3d: Transformer architectures with input-dependent dynamic depth for speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[23] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech 2020*, 2020.

[24] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "Beats: audio pre-training with acoustic tokenizers," in *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023.

[25] P. Littell, D. R. Mortensen, K. Lin, K. Kairis, C. Turner, and L. Levin, "Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.

[26] D. Gao, X. He, Z. Zhou, Y. Tong, K. Xu, and L. Thiele, "Rethinking pruning for accelerating deep inference at the edge," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[27] S. Ding, T. Chen, and Z. Wang, "Audio lottery: Speech recognition made ultra-lightweight, noise-robust, and transferable," in *International Conference on Learning Representations*, 2021.

[28] Y. Xie, J. J. Macoskey, M. Radfar, F.-J. Chang, B. King, A. Rastrow, A. Mouchtaris, and G. Strimel, "Compute cost amortized transformer for streaming asr," in *Interspeech 2022*, 2022.

[29] J. Macoskey, G. P. Strimel, J. Su, and A. Rastrow, "Amortized neural networks for low-latency speech recognition," in *Interspeech 2021*, 2021.

[30] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Interspeech 2015*, 2015.

[31] B. Zhang, I. Titov, B. Haddow, and R. Sennrich, "Adaptive feature selection for end-to-end speech translation," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[32] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations, ICLR*, 2017.

[33] J. Sakuma, T. Komatsu, and R. Scheibler, "MLP-based architecture with variable length input for automatic speech recognition," 2022. [Online]. Available: https://openreview.net/forum?id=RA-zVvZLYIy

[34] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 84–91.

[35] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, "Europarl-st: A multilingual corpus for speech translation of parliamentary debates," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.