ENHANCING AUTOMATIC SPEECH RECOGNITION FOR DIVERSE LANGUAGES THROUGH JOINT PRE-TRAINING AND FINE-TUNING WITH WAV2VEC2 MODELS

Yui Edward Chen [1] *A F M Saif* [2]

Chun Hui Senior High School [1]
Rensselaer Polytechnic Institute [2]

ABSTRACT

Joint self-supervised and supervised training has shown promising results in Automatic Speech Recognition (ASR) compared to the traditional two-stage training scheme. This joint training approach optimizes both self-supervised and supervised objectives simultaneously, seeking to unify the learning process for improved performance across tasks. In this paper, we perform an in-depth study of one such joint training algorithm, **BL-JUST**, as proposed in [17], using a pre-trained Wav2Vec2 model and an LSTM model as encoders. While the Wav2Vec2 model is pre-trained on the LibriSpeech dataset, the LSTM model is trained from scratch. We jointly pre-train and fine-tune both models on the LibriSpeech dataset and multiple languages from the CovoSt V2 dataset. Our results demonstrate that joint training significantly enhances the performance of foundation models like Wav2Vec2. Specifically, joint training reduces the Word Error Rate (WER) in Arabic, Dutch, and Mongolian by 1.9%, 1.3%, and 1.1% compared to baseline methods, respectively.

***Index Terms*—** automatic speech recognition, deep neural networks, self-supervised training, supervised training, Wav2Vec2

1. INTRODUCTION

With the rapid advancement of artificial intelligence (AI) and the growing complexity of input requirements in deep learning systems, Automatic Speech Recognition (ASR) has emerged as a widely used tool in numerous applications. ASR, an efficient input method, can save an average of 20 minutes in our daily life compared to conventional methods such as typing [8]. Despite the notable successes of AI in ASR tasks, these AI systems continue to face several challenges. Key challenges include robustness to noise, handling various accents and dialects, limited data resources, and difficulties adapting to new domains and speakers [6].

A promising approach to overcoming the challenges in ASR tasks is pre-training models on large, unlabeled datasets, enabling them to capture the underlying structure of languages. Self-supervised learning (SSL) has gained significant traction as a pre-training technique, allowing models to extract rich acoustic and linguistic features from speech data without requiring labeled inputs. One notable SSL technique, Wav2Vec2, has significantly improved ASR performance in multilingual settings [1]. By leveraging vast amounts of unlabeled speech data, Wav2Vec2 learns powerful representations that are highly effective for ASR tasks. Conventional supervised ASR training often relies on large labeled datasets from major languages like English [8] and faces challenges with low-resource languages, such as Ika [15], Arabic, Mongolian, or Dutch. Wav2Vec2 helps mitigate some of these issues through its robust architecture and ability to learn from unlabeled data.

Despite its success, Wav2Vec2's performance in low-resource languages remains subpar compared to rich-resource languages [7], which raises the question of whether the model's latent feature representations can be further aligned with specific downstream tasks, particularly in low-resource settings. We find that alignment can be enhanced using the BL-JUST algorithm [17]. BL-JUST integrates joint self-supervised training and supervised fine-tuning, refining the latent feature representation of Wav2Vec2 using an iterative process. This method allows the model to learn general acoustic features and also adapt these features for improved performance on downstream ASR tasks, especially for low-resource languages.

The BL-JUST algorithm can be valuable in low-resource ASR tasks, where labeled data is often limited. By cycling between SSL and SL, BL-JUST enhances the model's ability to generalize and adapt across different linguistic settings, making it well-suited for low-resource ASR challenges. This cyclic feedback mechanism allows Wav2Vec2 to incrementally improve its representations, making it more robust to dialectal variations, noise, and domain-specific requirements. BL-JUST's impact is especially significant for under-represented languages, where traditional models often struggle. The combination of Wav2Vec2's strong pre-training and BL-JUST's task-specific fine-tuning enables the model to better handle low-resource languages like Dutch, Mongolian, and Arabic, as found in the CovoSt V2 dataset [19]. This integration demonstrates Wav2Vec2's adaptability and performance improvements in low-resource, multilingual ASR tasks, addressing key challenges in the field [10, 7].

The contributions of our method are twofold:

- We integrate BL-JUST with the Wav2Vec2 model to improve the alignment of Wav2Vec2's latent feature representations with downstream tasks. BL-JUST leverages recent advances in penalty-based optimization to address ASR challenges, while maintaining computational efficiency and ensuring convergence [17].
- We demonstrate the model's ability to generalize across multiple languages using the CovoSt V2 dataset, highlighting its performance in diverse multilingual settings [4].

By incorporating joint SSL and SL training steps, BL-JUST refines the learned feature representations of the Wav2Vec2 model beyond traditional fine-tuning, leading to superior performance in ASR tasks, particularly across a range of low-resource languages, compared to conventional approaches [20].

2. PROBLEM FORMULATION

In this section, we provide preliminaries on bilevel optimization, and formulate Wav2Vec2 model training as BL-JUST training.

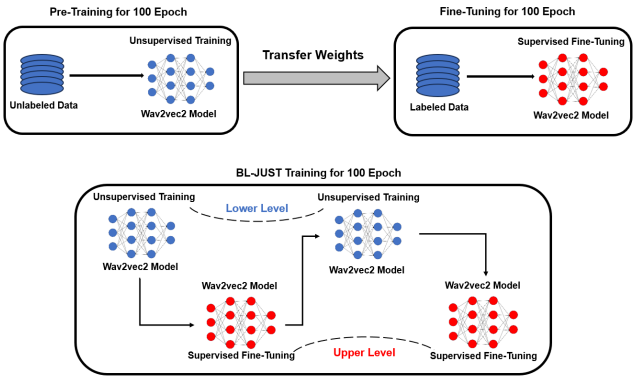

Fig. 1: Comparison between the proposed Wav2vec2-enbaled BL-JUST training method (bottom) with the PT+FT method (upper).

2.1. Bilevel optimization preliminaries

Bilevel optimization is a hierarchical problem consisting of two interdependent levels. The upper-level problem aims to minimize an objective function, subject to constraints imposed by the solution of a lower-level problem. Let the upper-level objective be represented as $F : \mathbb{R}^r \times \mathbb{R}^s \mapsto \mathbb{R}$ and the lower-level objective as $G : \mathbb{R}^r \times \mathbb{R}^s \mapsto \mathbb{R}$. The bilevel optimization problem can then be formulated as:

$$\min_{\phi \in \mathbb{R}^r, \theta \in \mathbb{R}^s} F(\phi, \theta) \quad \text{s.t.} \quad \theta \in \mathcal{T}(\phi) := \arg\min_{\theta \in \mathbb{R}^s} G(\theta, \phi), \tag{1}$$

where $\mathcal{T}(\phi)$ denotes the non-empty and closed set of optimal solutions for the lower-level problem given $\phi \in \mathbb{R}^r$. Although bilevel optimization has numerous applications, it is challenging to solve due to its inherent non-convexity and non-differentiability [2]. Recently, methods based on implicit gradients and unrolled differentiation have been proposed to address bilevel problems [16, 3, 13]. Nevertheless, these approaches are computationally intensive and struggle to scale to large models typical in ASR tasks.

2.2. Bilevel optimization for Wav2Vec2 model training

To reformulate the acoustic Wav2Vec2 model training as a bilevel optimization problem, we first introduce the self-supervised and supervised objective functions we will use in this work.

Self-supervised loss of Wav2Vec2. For self-supervised training, we use the same self-supervised losses used in Wav2Vec2 pretraining [1] to learn a good representation of the input speech from unlabeled data. In the original Wav2Vec2 paper, contrastive loss, $\mathcal{L}_c$, and diversity loss, $\mathcal{L}_d$ are used as a self-supervised loss. Then the total self-supervised loss is

$$\mathcal{L}_s = \mathcal{L}_c + \alpha \mathcal{L}_d, \tag{2}$$

where α is a tunable parameter.

Constrastive loss. Let, the output of the transformer is $\mathcal{T} \colon \mathcal{Z} \to \mathcal{C}$, where $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ is the output of the convolutional feature encoder for T time step, and $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_T\}$ is the output of the transformer model. The output of the feature encoder when it is discretized is $\mathbf{q}_t$, then the contrastive loss function, defined as

$$\mathcal{L}_c(\theta) = -\log \frac{\exp\left(\text{sim}(\mathbf{c}_t(\theta), \mathbf{q}_t)/\kappa\right)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp\left(\text{sim}(\mathbf{c}_t(\theta), \tilde{\mathbf{q}}/\kappa)\right)}. \tag{3}$$

Table 1: WERs under different learning rate settings of BL-JUST using CNN-LSTM acoustic models on Librispeech. There are 100 hours of data in the lower-level self-supervised training and 100 hours of data in the upper-level supervised training.

	α 5×	β 5×	Number of CNN	Number of LSTM	WER
LSTM	10^{-2}	10^{-2}	3	5	14.3%
	10^{-3}	10^{-3}	3	5	12.1%
	10^{-4}	10^{-3}	3	5	11.2%
	10^{-3}	10^{-4}	3	5	**10.0%**

Here, κ is a non-negative temperature and $\text{sim}(\cdot)$ is the cosine similarity between context representations and quantized latent representations calculated using,

$$\text{sim}(\mathbf{c}_t(\theta), \mathbf{q}_t) = \frac{\mathbf{c}_t(\theta)^\top \mathbf{q}_t}{\|\mathbf{c}_t(\theta)\| \|\mathbf{q}_t\|}. \tag{4}$$

Diversity loss. We use diversity loss similar to the Wav2Vec2 pretraining. The contrastive task relies on the codebook to encode both positive and negative samples. To promote a more effective utilization of the quantized codebook representations, the diversity loss $\mathcal{L}_d(\theta)$ is introduced. Specifically, it encourages the equal utilization of the V entries in each of the G codebooks by maximizing the entropy of the averaged softmax distribution $\bar{p}_g(\theta)$ over the codebook entries, which depends on the model parameters θ, across a batch of utterances. The softmax distribution does not include Gumbel noise or a temperature factor. The diversity loss is defined as:

$$\mathcal{L}_d(\theta) = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g(\theta)) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v}(\theta) \log \bar{p}_{g,v}(\theta) \tag{5}$$

CTC Loss. The Connectionist Temporal Classification (CTC) loss [5] is applied for supervised learning. Given an input sequence x_n and its corresponding label sequence y_n, the CTC loss minimizes the negative log-likelihood of y_n being predicted from the model's output, expressed as:

$$\mathcal{L}_{\text{CTC}}(\phi, \theta) = \frac{1}{N} \sum_{n=1}^{N} -\log P(y_n | m(x_n; \phi, \theta)) \tag{6}$$

Here, $m(x_n; \phi, \theta)$ is the model's output, where ϕ refers to the parameters of the classification layer, and θ refers to the parameters of the remaining network (referred to as the "backbone").

In the BL-JUST approach, we integrate the CTC loss within a bilevel optimization framework, where the upper-level objective is to minimize the supervised CTC loss, and the lower-level objective minimizes a self-supervised loss:

$$\begin{aligned} \min_{\phi, \theta} \quad & \mathcal{L}_{\text{CTC}}(\phi, \theta) \\ \text{s.t.} \quad & \theta \in \mathcal{S} := \arg\min_{\theta} \mathcal{L}_s(\theta). \end{aligned} \tag{7}$$

In this formulation, the lower-level problem constrains the optimization of the backbone parameters θ to those that minimize the self-supervised loss $\mathcal{L}_s(\theta)$, while the upper-level objective optimizes the CTC loss $\mathcal{L}_{\text{CTC}}(\phi, \theta)$ using the backbone parameters from the lower-level solution.

Table 2: WERs under various number of CNN and LSTM layers settings of BL-JUST using CNN-LSTM acoustic models on Librispeech. There are 100 hours of data in the lower-level self-supervised training and 100 hours of data in the upper-level supervised training.

Num of param	Num of CNN	Num of LSTM	WER (PT+FT)	WER (BL-JUST)
22415487	2	4	16.2%	13.1%
22530847	3	4	14.4%	12.9%
22646207	4	4	14.2%	12.3%
27142271	2	5	12.1%	10.2%
27257631	3	5	11.6%	10.0%
27372991	4	5	11.4%	10.0%
31869055	2	6	10.9%	9.7%
31984415	3	6	11.0%	9.6%
32099775	4	6	10.5%	9.5%

3. EXPERIMENTS

In this section, we describe the experimental details and compare traditional Wav2Vec2 fine-tuning with Wav2Vec2-JUST training.

3.1. Dataset

We evaluate the CNN-LSTM and Wav2Vec2 models on the LibriSpeech and CovoSt V2 datasets, respectively. The LibriSpeech dataset contains 960 hours of speech data. It has a sampling rate of 16KHz. For the CNN-LSTM model, we use 100 hours (train-clean-100) of unsupervised training data and 100 hours (train-clean-100) of supervised fine-tuning data, evaluating the model on the test-clean subset. For the Wav2Vec2 model, we selected three languages from CovoSt V2: Dutch (Nl), Mongolian (Mn), and Arabic (Ar). The Dutch dataset includes 2 hours of training and 2 hours of testing data, while the Mongolian dataset has 3 hours of training and 3 hours of testing data. The Arabic dataset consists of 2 hours of training and 2 hours of testing data. CovoSt V2 has a sampling rate of 48 KHz.

3.2. Training Strategy.

We follow the same strategy as in the original BL-JUST paper [17]. As for the evaluation for the Wav2Vec2 model, we test the performance of the Wav2Vec2-JUST and Wav2Vec2-FT respectively in the three CovoSt V2 datasets in Table 3. In Table: 2, we demonstrate the results of CNN-LSTM model using different number of CNN and LSTM layers.

3.3. Model

Wav2Vec2. The Wav2Vec2 model consists of a multi-layer convolutional feature encoder that takes raw audio input and outputs latent speech representations across multiple time steps [1]. We trained the model on three datasets: Dutch, Mongolian, and Arabic. The feature encoder includes blocks with a temporal convolution, layer normalization, and GELU activation, producing standardized latent representations labeled $\mathbf{z}_1, \ldots, \mathbf{z}_T$ for T time steps. These representations are fed into a Transformer network that captures context across the entire sequence, using relative positional embeddings [1]. To enhance pre-training, a portion of time steps in the latent space are masked and replaced with a shared feature vector [11]. The encoder output is then quantized to discrete representations via product quantization [9], which enables effective self-supervised learning.

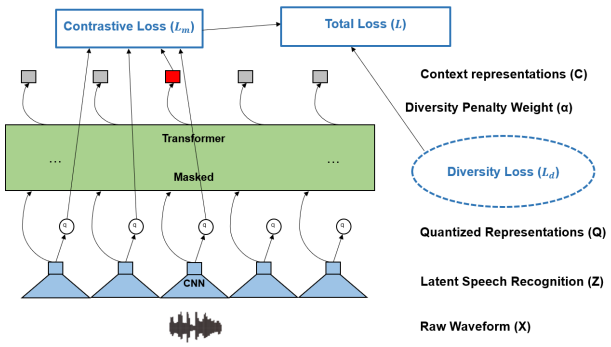

Fig. 2: The model architecture of Wav2vec2

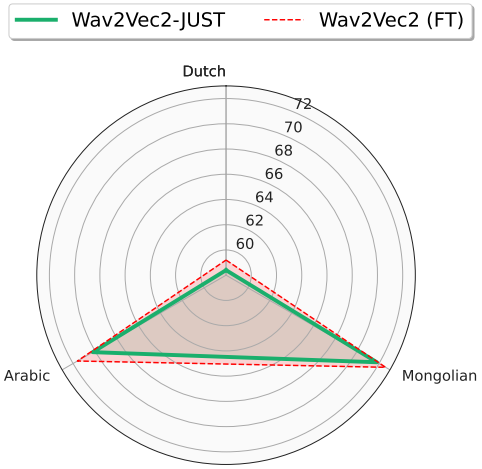

Fig. 3: Radar plots of ASR (WER) performance for Wav2Vec2 model. **Closer proximity to the origin** indicates better ASR performance.

CNN-LSTM. CNN-LSTM model consists of an input layer, different number of 1D convolutional layers, LSTM layers, and a fully connected layer [14]. CNN, known for its efficiency in reducing parameters in time-series data [12], uses convolutional and pooling layers to extract features. Each convolutional layer contains 32 feature maps with a 3x3 kernel and stride of 1. The pooling layer reduces the feature dimension by selecting the maximum value from each kernel-sized partition [14]. LSTM, designed to mitigate gradient issues in RNNs, has 256x2 hidden units per layer and uses forget, input, and output gates to process sequential data [18]. In this model, the CNN performs feature extraction, while the LSTM captures temporal dependencies, resulting in a robust model for ASR tasks [14].

3.4. ASR Performance

We compare Wav2Vec2-JUST method with traditional fine-tuning approaches. Additionally, we compare the performance of traditional pre-training plus fine-tuning (PT+FT) LSTM training with LSTM-JUST training.

Wav2Vec2. In Table 3, we present a comparison between the traditional Wav2Vec2 model fine-tuning and the BL-JUST training method. For this comparison, we utilize three languages from the CovoSt V2 dataset. BL-JUST outperforms traditional fine-tuning by 1.3%, 1.1%, and 1.9% in Dutch, Mongolian, and Arabic, respectively. Figure 3 provides a radar plot of the WERs to facilitate a better

Table 3: WERs of Wav2Vec2-JUST and Wav2Vec2 (FT) on Dutch, Mongolian, and Arabic languages, evaluated on the test dataset.

Language	Model	Method	Test WER (%)
Dutch	Wav2Vec2	JUST	58.4
		FT	59.2
Mongolian	Wav2Vec2	JUST	71.8
		FT	72.6
Arabic	Wav2Vec2	JUST	70.2
		FT	71.6

understanding of the results. The incorporation of the BL-JUST method with the pre-trained Wav2Vec2 model significantly enhances the quality of latent features, resulting in improved WER values compared to the traditional fine-tuning method.

CNN-LSTM Model. We compare the performance of the CNN-LSTM model trained with the BL-JUST method against the traditional PT+FT training approach, as shown in Table 2. Across all configurations, the BL-JUST method consistently outperforms PT+FT, with relative improvements ranging from 9.5% to 19.1%. Notably, BL-JUST demonstrates significant advantages, particularly in smaller models. For instance, with a model size of approximately 22.4M parameters, BL-JUST achieves a relative improvement of 19.1% over PT+FT. These results highlight the effectiveness of BL-JUST, especially in scenarios where model size is a limiting factor.

3.5. Ablation study.

In Table 1, we conduct a grid search to identify the optimal learning rate combination for self-supervised (lower-level) and supervised (upper-level) training in the BL-JUST framework using a CNN-LSTM acoustic model with 3 convolutional layers and 5 LSTM layers. The goal is to optimize performance on Librispeech data, utilizing 100 hours for each training phase. The table presents WERs under different learning rate settings for α (self-supervised) and β (supervised). The first configuration, with $\alpha = 10^{-2}$ and $\beta = 10^{-2}$, results in a high WER of 14.3%, indicating instability due to larger learning rates. Reducing both rates to $\alpha = 10^{-3}$ and $\beta = 10^{-3}$ improves WER to 12.1%, but further refinement is necessary. The third configuration, $\alpha = 10^{-4}$ and $\beta = 10^{-3}$, reduces WER to 11.2%, highlighting the benefits of cautious tuning in the self-supervised phase. The best configuration, $\alpha = 10^{-3}$ and $\beta = 10^{-4}$, yields the lowest WER of 10.0%. This balance enhances representation learning in the self-supervised phase while preventing overfitting during supervised training.

In Table 2, we demonstrate the effect of varying the number of LSTM and CNN layers on ASR performance. Increasing the number of layers generally improves the WER, with deeper models showing consistent performance gains. For example, models with 6 LSTM layers achieve the best WER across both PT+FT and BL-JUST methods, with BL-JUST reaching a WER as low as 9.5%. However, this improvement comes at the cost of increased computational complexity, as larger models with more layers require more memory and longer training times. Due to computational constraints, we limited our experiments to a maximum of 6 LSTM layers and 4 CNN layers, balancing model performance with practical feasibility. Further increases in model depth may yield additional gains but would require significantly more resources.

4. OPEN-SOURCE GITHUB

Our code is available on GitHub. It includes scripts for data preprocessing, model training, and evaluation, along with detailed setup instructions and links to the pre-trained models used in our experiments. We also include a speech-to-text transcription demonstration in the GitHub repository.

5. CONCLUSIONS

In this paper, we explore the impact of the recently proposed BL-JUST method [17] on fine-tuning the Wav2Vec2 model. Our results show that integrating BL-JUST with the pre-trained Wav2Vec2 model yields significantly better WER performance compared to traditional fine-tuning techniques. Additionally, we conducted extensive experiments with the LSTM model, where BL-JUST outperformed the conventional pre-training and fine-tuning approach. These findings emphasize the effectiveness of the BL-JUST method in enhancing automatic speech recognition training.

References

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

[2] Luce Brotcorne, Bernard Fortz, and Martine Labbé. Special issue on bilevel optimization. *EURO Journal on Computational Optimization*, 8:1–2, 2020.

[3] Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488, 2022.

[4] Yao-Fei Cheng, Hayato Futami, Yosuke Kashiwagi, Emiru Tsunoo, Wen Shen Teo, Siddhant Arora, and Shinji Watanabe. Task arithmetic for language expansion in speech translation. *arXiv preprint arXiv:2409.11274*, 2024.

[5] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of International Conference on Machine Learning*, pages 369–376, 2006.

[6] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[7] Rishabh Jain, Andrei Barcovschi, Mariam Yahayah Yiwere, Dan Bigioi, Peter Corcoran, and Horia Cucu. A wav2vec2-based experimental study on self-supervised learning methods to improve child speech recognition. *IEEE Access*, 11:46938–46948, 2023.

[8] Jesin James, Deepa P Gopinath, et al. Advocating character error rate for multilingual asr evaluation. *arXiv preprint arXiv:2410.07400*, 2024.

[9] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[10] Yacouba Kaloga, Shakeel A Sheikh, and Ina Kodrasi. Multiview canonical correlation analysis for automatic pathological speech detection. *arXiv preprint arXiv:2409.17276*, 2024.

[11] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[12] Byeong Soo Kim and Tag-Gon Kim. Cooperation of simulation and data model for performance analysis of complex systems. *International Journal of Simulation Modelling*, 18(4):608–619, 2019.

[13] Songtao Lu. Bilevel optimization with coupled decision-dependent distributions. In *International Conference on Machine Learning*, pages 22758–22789, 2023.

[14] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. A cnn-lstm-based model to forecast stock prices. *Complexity*, 2020(1):6622927, 2020.

[15] Uchenna Nzenwata and Daniel Ogbuigwe. Automatic speech recognition for the ika language. *arXiv preprint arXiv:2410.00940*, 2024.

[16] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, pages 737–746, 2016.

[17] AFM Saif, Xiaodong Cui, Han Shen, Songtao Lu, Brian Kingsbury, and Tianyi Chen. Joint unsupervised and supervised training for automatic speech recognition via bilevel optimization. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10931–10935. IEEE, 2024.

[18] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[19] Sheng Shi, Xuyang Cao, Jun Zhao, and Guoxin Wang. Joyhallo: Digital human model for mandarin. *arXiv preprint arXiv:2409.13268*, 2024.

[20] Ioannis Tsiamas, Gerard I Gállego, José AR Fonollosa, and Marta R Costa-jussà. Pushing the limits of zero-shot end-to-end speech translation. *arXiv preprint arXiv:2402.10422*, 2024.