applied sciences

MDPI

Article

JSUM: A Multitask Learning Speech Recognition Model for Jointly Supervised and Unsupervised Learning

Nurmemet Yolwas [1,2,*] and Weijing Meng [1,2]

[1] Xinjiang Multilingual Information Technology Laboratory, Urumqi 830017, China; mengweijing@stu.xju.edu.cn
[2] College of Information Science and Engineering, Xinjiang University, Urumqi 830017, China
* Correspondence: nurmemet@xju.edu.cn

Abstract: In recent years, the end-to-end speech recognition model has emerged as a popular alternative to the traditional Deep Neural Network—Hidden Markov Model (DNN-HMM). This approach maps acoustic features directly onto text sequences via a single network architecture, significantly streamlining the model construction process. However, the training of end-to-end speech recognition models typically necessitates a significant quantity of supervised data to achieve good performance, which poses a challenge in low-resource conditions. The use of unsupervised representation significantly reduces this necessity. Recent research has focused on end-to-end techniques employing joint Connectionist Temporal Classification (CTC) and attention mechanisms, with some also concentrating on unsupervised presentation learning. This paper proposes a joint supervised and unsupervised multi-task learning model (JSUM). Our approach leverages the unsupervised pre-trained wav2vec 2.0 model as a shared encoder that integrates the joint CTC-Attention network and the generative adversarial network into a unified end-to-end architecture. Our method provides a new low-resource language speech recognition solution that optimally utilizes supervised and unsupervised datasets by combining CTC, attention, and generative adversarial losses. Furthermore, our proposed approach is suitable for both monolingual and cross-lingual scenarios.

Keywords: end-to-end speech recognition; multitasking learning; supervised learning; unsupervised learning

Citation: Yolwas, N.; Meng, W. JSUM: A Multitask Learning Speech Recognition Model for Jointly Supervised and Unsupervised Learning. *Appl. Sci.* **2023**, *13*, 5239. https://doi.org/10.3390/app13095239

Academic Editor: Javier Hernando

Received: 5 April 2023
Revised: 14 April 2023
Accepted: 18 April 2023
Published: 22 April 2023

1. Introduction

Recent advances in deep learning architectures with a high degree of parameterization have led to the development of end-to-end approaches, which have demonstrated their effectiveness in achieving state-of-the-art performance in automatic speech recognition (ASR) tasks, as shown in several studies [1–3]. However, end-to-end speech recognition technology is fundamentally data-driven and necessitates a significant amount of supervised data to achieve satisfactory performance. Unfortunately, collecting such data is a time-consuming and expensive process, making it difficult to leverage this architecture for training speech recognition systems in low-resource languages.

Current work on end-to-end speech recognition focuses on two approaches: CTC [4–6] and attention-mechanism-based encoder–decoder [7–9]. CTC is advantageous as it permits duplicate and blank labels, does not necessitate exact alignment between input and output, and utilizes both forward and backward algorithms. However, CTC assumes independence between the outputs of different time steps, which may not be true for some sequential problems. On the other hand, attention-based encoder–decoders directly learn the mapping from acoustic frames to character sequences by utilizing encoders to convert input acoustic features into sequences of high-level hidden features. The attention module calculates attention weights between the previous decoder output and each frame of the encoder output, allowing the decoder to generate its output from previous output labels along

with context vectors. At each output time step, the model emits a character based on the input and target characters' historical context. Attention-based codecs do not rely on assumptions of conditional independence and generally outperform CTC without the need for language models. Nonetheless, the model's training is challenging due to misalignment issues in long input sequences, and the estimated alignment in the attention mechanism is susceptible to noise.

In the realm of computer vision and natural language processing, researchers have been investigating various hybrid architectures [10–12] that join CTC and attention mechanisms. These hybrid architectures have been shown to be effective in improving alignment errors and accelerating convergence in speech recognition tasks [13,14]. Several studies have successfully applied these hybrid models to such tasks. Compared to single-task models, multi-task learning methods can train multiple subtasks of a high-level task by sharing parameters between tasks to some degree, which enhances the generalization ability of the original task.

This paper proposes an end-to-end system (JSUM) based on multi-task learning using a joint loss approach inspired by previous research. In this approach, supervised learning and unsupervised adversarial learning are considered separate tasks, and their joint loss is used for multi-task learning. The supervised learning task uses a hybrid architecture of CTC and attention mechanisms, while the unsupervised learning task segments unlabeled audio using self-supervised phonetic representations and learns to map these representations to phonemes via adversarial training. The experimental results demonstrate that the proposed multi-task learning model can effectively utilize both supervised and unsupervised datasets within a unified end-to-end framework, which is beneficial for improving the use of limited data. Additionally, the results indicate that unsupervised cross-language representation learning in shared encoders can influence the multi-task learning model's performance and demonstrate its effectiveness in both single-language and cross-language scenarios.

2. Related Work

In this section, we will briefly review the work related to this paper in three sections: multilingual pre-training for speech recognition, joint CTC-Attention architecture, and unsupervised adversarial training.

2.1. Multilingual Pretraining of Speech Recognition

A common way to improve speech recognition in low-resource languages is to train multilingual speech recognition models. A. Stolcke et al. [15] trained features for phoneme classification in different languages and studied the portability of these features across domains and languages. L. Burget et al. [16] shared the parameters of a Gaussian mixture model for multilingual speech recognition, especially in low-resource conditions, which can be well achieved for cross-language sharing. Some work uses feedforward neural networks [17,18] or Long Short-Term Memory (LSTM) [19] to share the parameters of a neural network encoder to successfully apply multilingual acoustic models to low-resource speech recognition. Facebook proposed XLSR-53 [20], a large audio pre-training model pre-trained with unsupervised speech data in 53 languages, and demonstrated its effectiveness in speech recognition in low-resource languages. CLSRIL-23 [21] is an audio pre-training model based on self-supervised learning that learns cross-lingual speech representations from the original audio of 23 Indian languages to facilitate research on speech recognition in low-resource Indian languages.

2.2. Joint CTC-Attention Architecture

Due to CTC and attention having excelled in end-to-end speech recognition, many researchers have thought of combining the best of the two. Then the joint CTC-Attention architecture emerged in the work of Kim et al. [22], as shown in Figure 1. It alleviates alignment problems using the joint CTC-Attention model within a multitasking learning framework to improve robustness and fast convergence. This architecture outperforms the

CTC and attention model in speech recognition tasks under real noise and clean conditions. Deng K. et al. [14] propose a pre-trained Transformer (Preformer) S2S ASR architecture based on joint CTC/attention end-to-end models to fully utilize the pre-trained acoustic models (AMs) and language models (LMs). S. Liang et al. [23] used multi-language datasets based on the joint CTC-Attention architecture for end-to-end speech recognition.

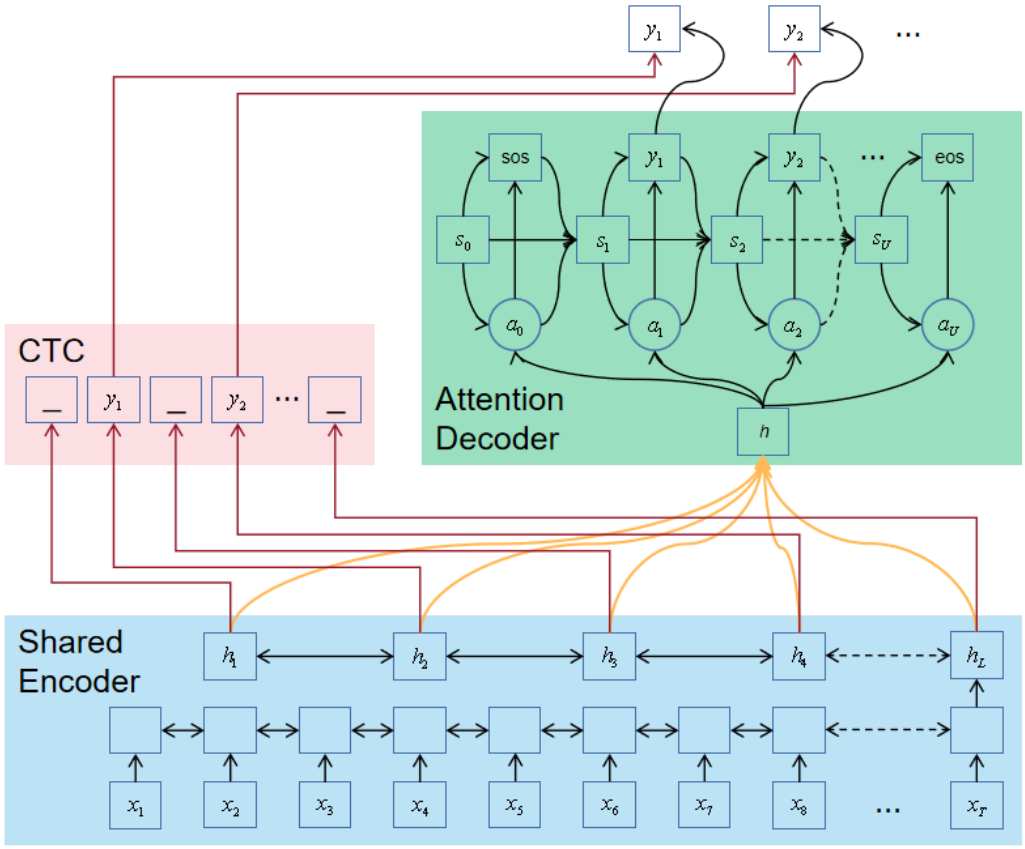

Figure 1. Joint CTC-Attention architecture.

2.3. Unsupervised Adversarial Training

DR Liu et al. [24] first implemented unsupervised phoneme recognition on TIMIT [25], a small, clean English data set for proof-of-concept experiments. Their system takes the Mel-scale Frequency Cepstral Coefficients (MFCC) representation of speech as input. It performs pre-processing steps that combine speech segmentation with word2vec [26] for (separately trained) segmented audio and feature discretization with k-mean clustering. The phoneme-based ASR system is trained by generating an adversarial network (GAN) [27] inspired by adversarial training using discrete features of segments. In the adversarial training framework, the ASR model acts as a generator that transcribes discrete indexed sequences into phoneme sequences with learnable embedded tables, which are then activated by softmax. A discriminator based on 2-layer convolution is trained to distinguish the output of a generator from the real phoneme sequence. The goal of a generator is to output a sequence of phonemes that is indistinguishable by a discriminator. By iteratively training these two modules, the generator can learn to map the pre-processed feature sequence to the phoneme sequence unsupervised.

3. Methods

3.1. Model

The multi-task learning model JSUM proposed in this paper is shown in Figure 2, which consists of a shared encoder and two decoders. The shared encoder is wav2vec2.0, pre-trained with unsupervised data from different languages. Each decoder tries to mini-

mize the loss of its own task. Given an unsupervised dataset and a finite supervised dataset, Decoder-1 and Decoder-2 can train both in our proposed model. The Decoder-1 uses supervised data for ASR training. The Decoder-2 uses unsupervised data for adversarial training to reconstruct the input features for better parameter learning in the shared encoder. This gives Decoder-2 the flexibility to leverage the large unsupervised datasets available for better representation learning in low-resource settings.

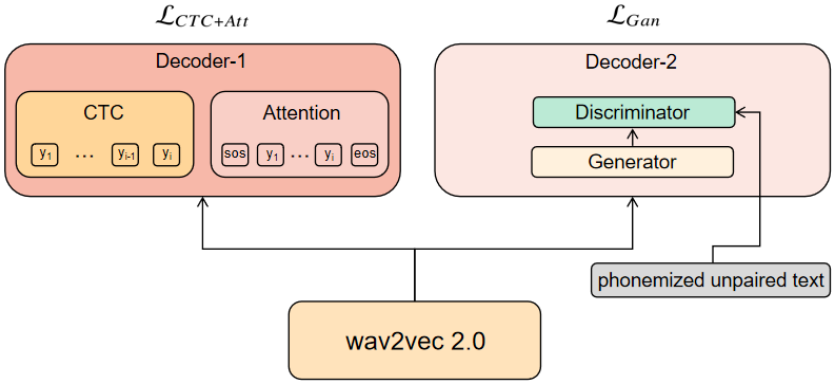

Figure 2. JSUM.

3.2. Loss Function

3.2.1. Joint CTC-Attention Loss

(1) $\mathcal{L}_{CTC}$

CTC is well suited for sequence modeling, and its training samples do not need to be aligned. By introducing a blank placeholder ε, the problem of the alignment of input and output labels in speech recognition is solved. The core idea of CTC is to allow repeated characters and blank characters. Given an input speech sequence $x = (x_1, \ldots, x_T)$ and the corresponding tag sequence $y = (y_1, \ldots, y_U)$, the tag sequence becomes $y' = (y_1, \ldots, y_U) \cup \{\varepsilon\}$. For example, if $y = (d, o, g)$, then $y' = (\varepsilon, d, \varepsilon, o, \varepsilon, g, \varepsilon)$. A CTC training model to maximize $P(y \mid x)$, namely in all possible tags sequence $\Phi(y')$ on the probability distribution of:

$$P(y|x) = \sum_{\pi \in \Phi(y')} P(\pi|x) \tag{1}$$

CTC is usually applied on top of the Recurrent Neural Network (RNN). Each RNN output cell is interpreted as the probability that the corresponding label will be observed at a particular time. Tags sequence $P(\pi \mid x)$ probability output is independent of the network and is modeled as a condition of the product of the:

$$P(\pi|x) \approx \prod_{t=1}^{T} P(\pi_t|x) = \prod_{t=1}^{T} q_t(\pi_t) \tag{2}$$

where $q_t(\pi_t)$ represents the softmax of π_t label in the output layer q of the RNN at time t.

The loss of CTC is minimized by the negative log-likelihood value of the true character sequence y^*:

$$\mathcal{L}_{\text{CTC}} \triangleq -\ln P(\boldsymbol{y}^* \mid \boldsymbol{x}) \tag{3}$$

The probability distribution $P(y \mid x)$ could be calculated before and after using the algorithm effectively:

$$P(y|x) = \sum_{u=1}^{|y'|} \frac{\alpha_t(u)\beta_t(u)}{q_t(y'_u)} \tag{4}$$

where $\alpha_t(u)$ is a forward variable, representing the total probability of all possible prefixes $(y'_{1:u})$ ending with the u-th label, and $\beta_t(u)$ is a backward variable of all possible suffixes

($y'_{u:U}$) beginning with the u-th label. The network is then trained with standard backpropagation by taking the derivative of the loss function with respect to $q_t(k)$ for any k labels, including whitespace characters.

(2) $\mathcal{L}_{Att}$

Unlike the CTC method, the attention model predicts each of the targets directly, without the need for intermediate representation or any hypothesis, and based on the probability chain rule according to the following recursion equation, directly estimates the a posteriori probability $P(y \mid x)$:

$$P(y|x) = \prod_u P(y_u|x, y_{1:u-1}) \tag{5}$$

$$h = Encoder(x) \tag{6}$$

$$y_u \sim \text{AttentionDecoder}(h, y_{1:u-1}) \tag{7}$$

The framework of the attention mechanism consists of two RNN layers: an encoder and an attention decoder, so it can learn sequences of two different lengths based on cross-entropy criteria. The encoder converts x to a high-level representation $h = (h_1, \ldots, h_L)$ in Equation (6). Then, the AttentionDecoder generates a probability distribution over the character y_u on h and all the characters seen previously, $y_{1:u-1}$ in Equation (7). L (L < T) is the number of frames output by the encoder. Here, a special start of the sentence (sos)/end of the sentence (eos) flag is added to the target set so that when (eos) is issued, the decoder completes the generation of the hypothesis. The loss function of the attention mechanism is computed from Equation (5) as follows:

$$\mathcal{L}_{\text{Att}} \triangleq -\ln P(\boldsymbol{y}^* \mid \boldsymbol{x}) = -\sum_u \ln P\left(y_u^* \mid \boldsymbol{x}, y_{1:u-1}^*\right) \tag{8}$$

where $y^*_{1:u-1}$ is the correct value of the previous character.

(3) $\mathcal{L}_{CTC+Att}$

The idea of the joint CTC-Attention architecture is to train the attention model encoder using the CTC objective function as an auxiliary task within the framework of multi-task learning (MTL). The overall architecture of the framework is shown in Figure 1, where the same Bidirectional-LSTM (Bi-LSTM) is shared by the CTC and the attention encoder network. The advantage of CTC over the attention model is that its forward-reverse algorithm can realize the monotonic alignment between the speech sequence and the label sequence and can make the learning of the model faster. Therefore, the objective function of the joint architecture is represented as follows by using both CTC in Equation (3) and the attention mechanism in Equation (8):

$$\mathcal{L}_{\text{MTL}} = \alpha_1 \mathcal{L}_{\text{CTC}} + \alpha_2 \mathcal{L}_{\text{Att}} \tag{9}$$

where $\alpha_1 + \alpha_2 = 1$, and α_1 and α_2 are adjustable hyperparameters.

For Decoder-1, we use Equation (9) to train on the available supervised dataset to obtain the joint CTC-Attention loss $\mathcal{L}_{CTC+Att}$.

3.2.2. Generative Adversarial Loss

In order to carry out unsupervised representation learning for Decoder-2, we use Equation (10) to train on unsupervised data to obtain the generative adversarial loss $\mathcal{L}_{Gan}$. In the setup of this article, the original GAN target with gradient penalty, fragment smoothing penalty, and phoneme diversity penalty is used:

$$\mathcal{L}_{Gan} = \min_{\mathcal{G}} \max_{\mathcal{C}} \mathbb{E}_{P_r \sim \mathcal{P}^r}[\log \mathcal{C}(P^r)] - \mathbb{E}_{S \sim \mathcal{S}}[\log(1 - \mathcal{C}(\mathcal{G}(S)))] - \lambda \mathcal{L}_{gp} + \gamma \mathcal{L}_{sp} + \eta \mathcal{L}_{pd} \tag{10}$$

where $P^r \sim \mathcal{P}^r$ is the phonemized unlabeled text, $\mathcal{G}(S)$ is the transcribed output of the generator representing S for the input segment of some unlabeled speech audio. The first one trains the discriminator to assign a high probability to the true transcript, and the second one encourages the discriminator to assign a low probability to the generator output, where $\mathcal{L}_{gp}$ is a gradient penalty, $\mathcal{L}_{sp}$ is a smoothing penalty, and $\mathcal{L}_{pd}$ is a phoneme diversity loss.

For gradient penalty $\mathcal{L}_{gp}$, the training is stabilized by punishing the discriminator with respect to the gradient norm of the input. Calculate the penalty for random samples $\widetilde{P} \sim \widetilde{\mathcal{P}}$, which are linear combinations of real and pseudo-sample pairs of activation:

$$\mathcal{L}_{gp} = \underset{\widetilde{P}\sim\widetilde{\mathcal{P}}}{\mathbb{E}}\left[\left(\parallel \nabla \mathcal{C}(\widetilde{P}) \parallel -1\right)^2\right] \tag{11}$$

For the segmented smoothness penalty $\mathcal{L}_{sp}$, k-mean segmentation of speech audio is more fine-grained than typical phonemized transcription, and adjacent representations are highly correlated:

$$\mathcal{L}_{sp} = \sum_{(p_t,p_{t+1})\in\mathcal{G}(S)} {p_t - p_{t+1}}^2 \tag{12}$$

For phoneme diversity loss $\mathcal{L}_{pd}$, it helps to punish the generation network's low usage of some phoneme words and maximizes the entropy of the generator's average softmax distribution $H_{\mathcal{G}}(\mathcal{G}(S))$ on the phoneme vocabulary of a batch of (B) utterances:

$$\mathcal{L}_{pd} = \frac{1}{|B|}\sum_{S\in B} -H_{\mathcal{G}}(\mathcal{G}(S)) \tag{13}$$

3.2.3. Multitasking Learning Loss

For the generative adversarial loss, we only consider the loss of an unsupervised data set, and combine the joint CTC and attention loss with the generative adversarial loss as the loss of multitasking learning $\mathcal{L}_{MTL}$:

$$\mathcal{L}_{MTL} = \alpha_1\mathcal{L}_{\text{CTC}} + \alpha_2\mathcal{L}_{\text{Att}} + \alpha_3\mathcal{L}_{\text{Gan}} \tag{14}$$

where $\alpha_1 + \alpha_2 + \alpha_3 = 1$, α_1, α_2, and α_3 are all adjustable hyperparameters.

4. Experimental Setup

4.1. Datasets

Several corpora and languages are considered to evaluate the validity of the proposed approach.

CommonVoice [28] is a multilingual reading speech corpus that currently has 17,127 h of speech data in 104 languages, including English, French, German, Dutch, and Chinese, and allows anyone to contribute their voice to it. The amount of data for each language ranged from 2 h for Kazakh ("low resource") to 3209 h for English ("high resource"). For Kazakh, the Kazakh data set in CommonVoice contains 2 h of 1400 pieces of speech data and their labels. Additionally, using the KSC dataset [29], which contains more than 330 h of Kazakh speech sounds and tags crowdsourced over the Internet, volunteers from different regions and age groups were asked to read sentences presented through a web browser. In total, there are 153,000 recordings from more than 1600 different devices. The recordings were collected by people from different regions and age groups who completed the recordings by reading sentences presented in web browsers, and all accepted recordings were manually checked by native Kazakh speakers. Additionally, all annotations are in the Cyrillic alphabet, and the voice is stored in WAV format. In this paper, 20 h speech data randomly selected from the training set is used as the voice data to generate adversarial training, and 10,000 unmatched texts from the training set are used as the unlabeled text data. An hour's worth of randomly selected data from KSC's test set was used for the evaluation.

For unsupervised pretraining of encoders, consider 960 h of untagged data using Librispeech [30] and 792 h of untagged data from CommonVoice in ten languages including Kazakh: Kazakh (kk), Turkish (tr), Kyrgyz (ky), Tatar (tt), Dutch (du), Chinese (zh), Russian (ru), Italian (it), Spanish (es), and French (fr). In keeping with the Kazakh language, The supervised data duration for each language trained by Decoder-1 is set to 2 h, and the audio and transcription of the same data are used as unlabeled data to train Decoder-2. The assessment duration is set according to the partitioning method proposed by Riviere et al. [31]. The data duration for each language is shown in Table 1.

Table 1. Data duration of 10 languages in CommonVoice.

Lang	kk	tr	ky	tt	du	zh	ru	it	es	fr
Unsup.pretrain (h)	2	11	17	17	29	50	55	90	168	353
Sup.train (h)	2	2	2	2	2	2	2	2	2	2
Evaluation (h)	1	1	1	1	1	1	1	1	1	1

For Decoder-1 and Decoder-2, TIMIT is first used to explore the most appropriate α setting for calculating joint loss $\mathcal{L}_{CTC+Att}$. Next, in order to explore the effectiveness of the JSUM in the use of supervised and unsupervised learning under low-resource conditions, experiments were conducted on several other corpora. LibriSpeech, KSC, TIMIT, and THUYG-20 data set specifications are shown in Table 2.

Table 2. Specifications of LibriSpeech, KSC, TIMIT, and THUYG-20 data sets.

Dataset	Duration			
	Train	Dev	Test	Total
Librispeech	960.9	10.7	10.5	982.1
KSC	318.4	7.1	7.1	332.6
TIMIT	2.9	0.3	0.2	3.4
THUYG-20	20.2	1.1	2.4	23.7

The TIMIT dataset contains about five hours of recordings and time-aligned phoneme labels, a total of 6300 sounds and corresponding labels, including SX (sentences with tight phonemes), SI (sentences with divergent phonemes), and SA (dialect sentences), with both "match" and "do not match" settings. The training set, development set, and test set in the "match" setting contain 3696, 400, and 192 voices and their corresponding labels, respectively. Only SX and SI are included in this setting. For the "mismatch" setting, there are 3000 speeches and 1000 labels separated from the training portion of the complete data set, but these unlabeled text labels do not contain transcripts of audio data. In this paper, the standard "matching" setting is used to explore different alphas to calculate the joint loss $\mathcal{L}_{CTC+Att}$.

For the Uygur language, use THUYG-20 [32], an open Uygur speech database jointly created by Tsinghua University and Xinjiang University. The database includes about 20 h of training data and 1 h of test data. The full audio training data is used as unsupervised data to generate adversarial learning to explore the effectiveness of the proposed multitasking learning model in a cross-language setting.

4.2. Model Configuration

The encoder–decoder module of the ESPNET toolkit [33] is used to develop the proposed JSUM that blends supervised and unsupervised adversarial learning, as shown in Figure 2. The shared encoder uses the unsupervised, pre-trained wav2vec 2.0 model. This pre-training process is implemented in fairseq [34]. In the wav2vec 2.0 model [35], there are two transformer architectures: (1) Base model, with 12 layers of transformer, model dimension is 768, internal dimension is 3072, and with 8 attention heads; (2) Large model,

with 24 layers of transformer, model dimension is 1024, internal dimension is 4096, and with 16 attention heads. This paper uses the Base model to pre-train the encoder. Decoder-1 consists of a single layer GRU containing 300 cells, as shown in Figure 3.

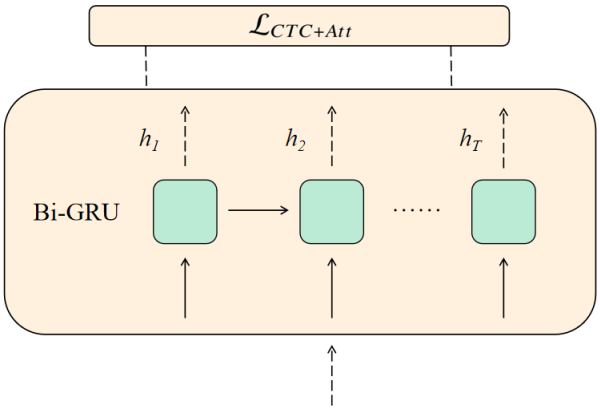

Figure 3. Schematic diagram of Decoder-1.

Decoder-2 consists of a generator and a discriminator, as shown in Figure 4. The generator is a single non-causal convolution with a kernel size of 4. The input is a segment of dimension 512 representing S, and the output is a $|O|$ dimensional vector, which represents the probability distribution on the phoneme vocabulary. The subsequent generator predictions are combined and then fed into the discriminator, and the softmax normalization is applied. The discriminator consists of three causal convolutional blocks with a kernel size of 6 and a hidden size of 384. The input to the discriminator is the output of the generator. The output is a single logit for each time step, indicating the likelihood that the sample came from the distribution of the data. The generator and discriminator were optimized by Adam using $\beta_1 = 0.5$ and $\beta_2 = 0.94$, respectively. The discriminator and generator were optimized alternately during this training. The discriminator's weight attenuation is 1×10^{-4}, while the generator does not use heavy attenuation. The discriminator's weight decay is set to 1×10^{-4}, while the generator does not use weight attenuation. The learning rates of the generator and discriminator are set to 1×10^{-5} and 1×10^{-4}, respectively, and remain constant throughout the training process.

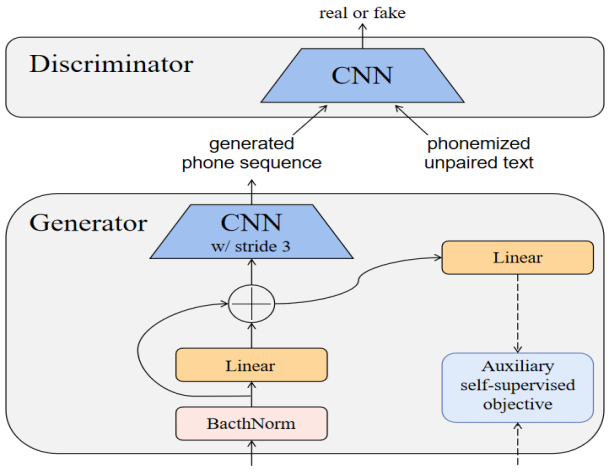

Figure 4. Schematic diagram of Decoder-2.

In each training batch, a joint CTC-Attention loss for supervised data and a generative adversarial loss for unsupervised data were calculated using mixed samples from both supervised and unsupervised data sets. A total of 150 k steps were trained. In each training step, Decoder-1 used 80 pairs of paired samples randomly selected from the supervised

data, and Decoder-2 used 80 samples randomly selected from the unlabeled audio data and 80 text samples randomly selected from the unlabeled text data. The entire training process was performed on a single V100 and took about 18 h.

4.3. Unsupervised Pre-Training

For the shared encoder wav2vec 2.0, an English model (LS_{960h}) is first pre-trained using Librispeech's 960 h of unlabelled audio. For comparison, a multilingual pre-training model uses unlabelled audio in 10 languages plus 167 h of English audio training ($ML_{960h\text{-}10}$).

4.4. Phonemized Unlabeled Text

TIMIT provides phonetic transcription aligned with time and annotates a set of 60 phonemes adapted from the ARPAbet system, which treats silence as a phoneme. In addition, it includes mappings from 60 phoneme tables to 48 and 39 phoneme tables. Phoneme error rates are usually calculated on 39 phoneme tables, and it is common practice to map phonemic tags to 39 phoneme tables for training.

For other languages, use the multilingual text-to-phoneme conversion tool Phonemizer [36], which supports phoneme conversion in a large number of different languages.

In order for shared encoder representations to jointly support supervised learning tasks and unsupervised adversarial learning tasks, Decoder-1 and Decoder-2 joint losses are trained to predict phonemes rather than characters or words. Therefore, phoneme error rate (PER) was used as an evaluation index in this study.

5. Results

5.1. Multitasking Learning Based on TIMIT

In the multi-task learning based on TIMIT, a shared encoder is selected to pre-train wav2vec2.0 with 960 h of Librispeech unsupervised audio. As shown in Table 3, compared with the situation where only CTC and attention joint loss $\mathcal{L}_{CTC+Att}$ were used as the total loss $\mathcal{L}_{MTL}$ for supervised learning without the addition of generating adversarial loss ($\alpha_1 = \alpha_2 = 0.5$, $\alpha_3 = 0$), adding generative adversarial losses (α_3 increased from 0 to 0.5) reduces the phoneme error rate.

Table 3. PER% observed for supervised training with TIMIT, with different α's considering generative adversarial for training Decoder-2.

α_1	α_2	α_3	PER
0.5	0.5	0	22.8
0.6	0.2	0.2	21.9
0.2	0.6	0.2	21.4
0.33	0.33	0.33	21.1
0.25	0.25	0.5	22.6

Through the experiments with different weights of CTC, attention, and generative adversarial loss, the phoneme error rate is 22.8% without taking generative adversarial loss into account. With the increasing generative adversarial loss weight, the phoneme error rate is reduced to different degrees. When the same weight $\alpha_1 = \alpha_2 = \alpha_3 = 0.33$, the phoneme error rate was 21.1%, which decreased by 1.7% compared with the best α combination with the same weight. Therefore, it can be inferred that unsupervised generative adversarial learning as an additional task can prevent the overfitting phenomenon of the model on a given small, supervised data set.

5.2. The Effect of Unsupervised Cross-Linguistic Representation Learning on JSUM in Shared Encoders

For the ten CommonVoice languages selected, the pure English pretraining model LS_{960h} and the multi-language pretraining model $ML_{960h\text{-}10}$ are used as shared encoders, respectively. For each language, Decoder-1 is trained with 2 h of the supervised data set,

and audio and text of the same data set are respectively used as unsupervised training data for Decoder-2 for adversarial learning.

When using the pure English pretraining model and the multilanguage pretraining model as shared encoders, experiments are carried out in 10 languages, respectively, by using the JSUM. In order to be consistent with the Kazakh language, all languages used 2 h of supervised data to simulate the setting of low resources. The experimental results are shown in Table 4. Compared with previous work, the multi-tasking learning method proposed in this paper has significantly reduced the phoneme error rate in all languages. Moreover, the selection of unsupervised cross-linguistic representation learning in shared encoders also has a significant impact on the recognition effect of the proposed multi-tasking learning model. It can be observed that the phoneme error rates of $ML_{960h\text{-}10}$ and LS_{960h} are 0.6% and 2.2% lower than those of low-resource languages such as Kazakh and Turkish, respectively.

Table 4. PER% of multi-tasking learning for ten languages with LS_{960h} and $ML_{960h\text{-}10}$ as shared encoders.

Model	Language									
	kk	**tr**	**ky**	**tt**	**du**	**zh**	**ru**	**it**	**es**	**fr**
Baselines from previous work (Unsupvised pretraining)										
m-CPC [28]	-	49.7	40.7	44.0	44.4	55.5	45.2	42.1	38.7	49.3
m-CPC [28]	-	47.3	41.2	42.0	42.5	55.0	43.7	40.5	38.0	47.1
Fer et al. [37]	-	43.4	38.7	42.5	47.9	54.3	45.2	39.0	36.6	48.3
Our monolingual and multilingual models										
Encoder										
LS_{960h}	22.3	16.3	13.8	12.3	20.6	29.9	19.7	20.2	13.7	22.4
$ML_{960h\text{-}10}$	21.7	14.1	11.2	10.4	18.9	27.6	15.8	16.5	10.8	18.3

5.3. Speech Recognition for Kazakh under Low Resource Conditions

This paper explores the effectiveness of the JSUM in using supervised learning and unsupervised learning under the condition of low resources and uses $ML_{960h\text{-}10}$ as a shared encoder. Using 2 h of Kazakh language data from CommonVoice (Kk-Sup2h) as supervised data, all experimental results were conducted on 1 h of randomly selected data from the KSC test set.

As shown in Table 5, when training the model using only supervised data, the phoneme error rate is 22.8%. When generative adversarial loss is added, the phonemic error rate decreases by 1.1%, which is consistent with the experimental results in Section 5.2. At this time, supervised data Kk-Sup2h is also used for generative adversarial learning, but text labels are scrambled and phonemized to be the text data required for unsupervised adversarial learning. In each training batch, a mixture of samples from both supervised and unsupervised data was used. For supervised data, only joint CTC and attention losses are calculated, and for unsupervised data, only generative adversarial losses are considered.

Table 5. PER% of multi-tasking learning for Kazakh under low resource conditions in single-language and cross-language settings.

α_1	α_2	α_3	**Supervied Train**	**Unsupervied Train**	**PER**
0.5	0.5	0	Kk-Sup2h	-	22.8
0.33	0.33	0.33	Kk-Sup2h	Kk-Sup2h	21.7
0.33	0.33	0.33	Kk-Sup2h	KSC-Unsup20h	21.2
0.33	0.33	0.33	Kk-Sup2h	THUYG-Unsup20h	21.4

In the monolingual environment, unsupervised data from other data sets are considered to improve the model's performance. Additionally, 20 h Kazakh voice frequency data (KSC-

unsup20h) is randomly selected from the training set of KSC as unsupervised data, and 10,000 unmatched training set texts are used as unlabeled text data. In this setting, the phoneme error rate was 21.2%, 0.5% lower than without any additional unsupervised data set.

In order to demonstrate that the JSUM can be adapted to cross-language settings, unsupervised data with phonemically similar features to Kazakh are used. Uygur and Kazakh belong to the Turkic language family of the Altaic family. To do this, consider training the model with THUYG-Unsup20h as unsupervised data and Kk-Sup2h as supervised data. It can be seen that, although the recognition effect is slightly worse than using additional same-language data, the phoneme error rate is reduced by 0.3% compared with using only the supervised data set, indicating that the JSUM is effective in cross-language settings under the condition of low resources.

6. Conclusions

In this paper, we propose JSUM, a multi-task learning ASR model that joins supervised learning and unsupervised learning. It can leverage data from supervised and unsupervised sources to maximize available resources. In the experiment of this paper, unsupervised pre-training and multi-task learning methods are preliminarily combined. As for the proposed joint learning method, it includes CTC, attention, and generative adversarial losses. First, we explore the most appropriate weight allocation between losses in JSUM. Then the shared encoder is pre-trained using single-language and multi-language settings, respectively. We also used ten languages in CommonVoice to simulate low-resource settings. Experiments have shown that JSUM can effectively reduce the phoneme error rate in these ten languages under low-resource conditions. Unsupervised cross-language representation learning in shared encoders also significantly impacts the multi-tasking learning model. The multi-language shared encoder can achieve a lower phoneme error rate than the pure English shared encoder. Finally, through the experiment with the Kazakh language under the condition of low resources, we find that using additional unsupervised datasets in JSUM can also improve the recognition effect. By using unsupervised data sets of languages with language similarity, we further demonstrate that using similar language data in a cross-language setting can enhance the performance of our model. In future work, we will investigate different architectures that are likely to improve performance.

Author Contributions: Writing—original draft, W.M.; writing—review and editing, N.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China—Research on Key Technologies of Speech Recognition of Chinese and Western Asian Languages under Resource Constraints (Grant No. 62066043), and the National Language Commission Key Project of China—Research on Speech Keyword Search Technology of Chinese and Western Asian Languages (Grant No. ZDI135-133).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this article are open source and available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Q.; Lu, H.; Sak, H.; Tripathi, A.; McDermott, E.; Koo, S.; Kumar, S. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 7829–7833.
2. Moritz, N.; Hori, T.; Le, J. Streaming automatic speech recognition with the transformer model. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6074–6078.

3. Kahn, J.; Lee, A.; Hannun, A. Self-training for end-to-end speech recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 7084–7088.
4. Graves, A.; Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; PMLR: London, UK, 2014; pp. 1764–1772.
5. Miao, Y.; Gowayyed, M.; Metze, F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 167–174.
6. Yao, Z.; Wu, D.; Wang, X.; Zhang, B.; Yu, F.; Yang, C.; Peng, Z.; Chen, X.; Xie, L.; Lei, X. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. *arXiv* **2021**, arXiv:2102.01547.
7. Pham, N.Q.; Nguyen, T.S.; Niehues, J.; Müller, M.; Waibel, A. Very deep self-attention networks for end-to-end speech recognition. *arXiv* **2019**, arXiv:1904.13377.
8. Rossenbach, N.; Zeyer, A.; Schlüter, R.; Ney, H. Generating synthetic audio data for attention-based speech recognition systems. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 7069–7073.
9. Yeh, C.F.; Wang, Y.; Shi, Y.; Wu, C.; Zhang, F.; Chan, J.; Seltzer, M.L. Streaming attention-based models with augmented memory for end-to-end speech recognition. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 8–14.
10. Xu, D.; Ouyang, W.; Wang, X.; Sebe, N. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 675–684.
11. Atapour-Abarghouei, A.; Breckon, T.P. To complete or to estimate, that is the question: A multi-task approach to depth completion and monocular depth estimation. In Proceedings of the 2019 International Conference on 3D Vision (3DV), Quebec, QC, Canada, 16–19 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 183–193.
12. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-task deep neural networks for natural language understanding. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6084–6088.
13. Miao, H.; Cheng, G.; Gao, C.; Zhang, P.; Yan, Y. Transformer-based online CTC/attention end-to-end speech recognition architecture. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6084–6088.
14. Deng, K.; Cao, S.; Zhang, Y.; Ma, L. Improving hybrid ctc/attention end-to-end speech recognition with pre-trained acoustic and language models. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 76–82.
15. Stolcke, A.; Grezl, F.; Hwang, M.-Y.; Lei, X.; Morgan, N.; Vergyri, D. Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech, and Signal Processing ICASSP, Toulouse, France, 14–19 May 2006.
16. Burget, L.; Schwarz, P.; Agarwal, M.; Akyazi, P.; Feng, K.; Ghoshal, A.; Glembek, O.; Goel, N.; Karafiat, M.; Povey, D.; et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 4334–4337.
17. Heigold, G.; Vanhoucke, V.; Senior, A.; Nguyen, P.; Ranzato, M.A.; Devin, M.; Dean, J. Multilingual acoustic models using distributed deep neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 8619–8623.
18. Huang, J.T.; Li, J.; Yu, D.; Deng, L.; Gong, Y. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 7304–7308.
19. Li, X.; Dalmia, S.; Black, A.W.; Metze, F. Multilingual speech recognition with corpus relatedness sampling. *arXiv* **2019**, arXiv:1908.01060.
20. Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A.; Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *arXiv* **2020**, arXiv:2006.13979.
21. Gupta, A.; Chadha, H.S.; Shah, P.; Chhimwal, N.; Dhuriya, A.; Gaur, R.; Raghavan, V. Clsril-23: Cross lingual speech representations for indic languages. *arXiv* **2021**, arXiv:2107.07402.
22. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4835–4839.
23. Liang, S.; Yan, W.Q. A hybrid CTC+ Attention model based on end-to-end framework for multilingual speech recognition. *Multimed. Tools Appl.* **2022**, *81*, 41295–41308. [CrossRef]
24. Liu, D.R.; Chen, K.Y.; Lee, H.; Lee, L.S. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings. *arXiv* **2018**, arXiv:1804.00316.

25. Garofolo, J.S.; Lamel, L.F.; Fisher, W.M.; Fiscus, J.G.; Pallett, D.S.; Dahlgren, N.L. *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM*; Linguistic Data Consortium: Philadelphia, PA, USA, 1993.
26. Wang, Y.H.; Lee, H.; Lee, L. Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 6269–6273.
27. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
28. Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F.M.; Weber, G. Common voice: A massively-multilingual speech corpus. *arXiv* **2019**, arXiv:1912.06670.
29. Khassanov, Y.; Mussakhojayeva, S.; Mirzakhmetov, A.; Adiyev, A.; Nurpeiissov, M.; Varol, H.A. A crowdsourced open-source Kazakh speech corpus and initial speech recognition baseline. *arXiv* **2020**, arXiv:2009.10334.
30. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 5206–5210.
31. Riviere, M.; Joulin, A.; Mazaré, P.E.; Dupoux, E. Unsupervised pretraining transfers well across languages. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 7414–7418.
32. Rouzi, A.; Shi, Y.; Zhiyong, Z.; Dong, W.; Hamdulla, A.; Fang, Z. THUYG-20: A free Uyghur speech database. *J. Tsinghua Univ. (Sci. Technol.)* **2017**, *57*, 182–187.
33. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Soplin, N.E.Y.; Heymann, J.; Wiesner, M.; Chen, N.; et al. Espnet: End-to-end speech processing toolkit. *arXiv* **2018**, arXiv:1804.00015.
34. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv* **2019**, arXiv:1904.01038.
35. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
36. Bernard, M.; Titeux, H. Phonemizer: Text to phones transcription for multiple languages in python. *J. Open Source Softw.* **2021**, *6*, 3958. [CrossRef]
37. Fer, R.; Matějka, P.; Grézl, F.; Plchot, O.; Veselý, K.; Černocký, J.H. Multilingually trained bottleneck features in spoken language recognition. *Comput. Speech Lang.* **2017**, *46*, 252–267. [CrossRef]