

GUIDED CONTRASTIVE SELF-SUPERVISED PRE-TRAINING FOR AUTOMATIC SPEECH RECOGNITION

Aparna Khare¹, Minhua Wu¹, Saurabhchand Bhati^{2*}, Jasha Droppo¹, Roland Maas¹

¹Amazon Alexa, USA

²Johns Hopkins University, USA

ABSTRACT

Contrastive Predictive Coding (CPC) is a representation learning method that maximizes the mutual information between intermediate latent representations and the output of a given model. It can be used to effectively initialize the encoder of an Automatic Speech Recognition (ASR) model. We present a novel modification of CPC called Guided Contrastive Predictive Coding (GCPC). Our proposed method maximizes the mutual information between representations from a prior-knowledge model and the output of the model being pre-trained, allowing prior knowledge injection during pre-training. We validate our method on 3 ASR tasks: German, French and English. Our method outperforms CPC pre-training on all three datasets, reducing the Word Error Rate (WER) by 4.44%, 6.55% and 15.43% relative on the German, French and English (Librispeech) tasks respectively, compared to training from scratch, while CPC pre-training only brings 2.96%, 1.01% and 14.39% relative WER reduction respectively.

Index Terms— Self-supervised learning, RNN-T, ASR

1. INTRODUCTION

Self-supervised Learning (SSL) has drawn a lot of recent attention in the machine learning community. After its successful applications in the natural language processing domain [1, 2, 3], it has also become an active research area for speech processing.

One of the main categories of SSL methods learns representations by reconstructing the signal such as full reconstruction with autoencoders [4, 5], future reconstruction with Autoregressive Predictive Coding (APC) [6] and masked reconstructions [7, 8, 9]. Instead of reconstructing the exact signal, HuBERT [10] learns representations by utilizing an offline clustering step to provide aligned target labels for a masked prediction loss. Another category of SSL technology in literature learns representations through a contrastive loss by distinguishing a true future audio sample from a set of negative examples, such as the Contrastive Predictive Coding (CPC) model [11] and wav2vec [12]. Vq-wav2vec [13] uses a vector quantization module in addition to contrastive loss

to learn discrete representations and wav2vec 2.0 [14] minimizes the contrastive loss defined over contextual representations in the masked region. In addition, w2v-BERT [15] combines the two categories by optimizing two self-supervised losses simultaneously (the contrastive loss and masked language modeling loss).

All of these methods learn representations from the acoustic data distribution only, which may not be optimal for the downstream ASR task. More recently, Wang et al. propose two supervision-guided codebook generation approaches to get better pre-trained embeddings for the downstream ASR task in [16]. On top of HuBERT pre-training, it uses the phoneme alignments as training targets. It also tries to perform K-means clustering on the supervised speech features extracted from an end-to-end CTC model [17]. However, this work focuses on the masked prediction self-supervised learning and all the ASR experiments are conducted with the Librispeech dataset with just a few hundred hours of labeled data. In our work, we focus on exploring the contrastive loss based SSL method instead and experiment with large-scale datasets. We propose to introduce weak guidance to improve alignment between the learned representations and the downstream task. The weak guidance is provided in the form of posteriors from a prior-knowledge model learned from a small labeled dataset, which will be discussed in detail in Section 2.2.

To combine the self-supervised and supervised training to improve performance of the final ASR task, most existing methods in the literature adopt a 2-stage scheme, where only the self-supervised loss is optimized at the first pre-training stage, and the supervised loss is optimized at the second stage. Wav2vec [12] and vq-wav2vec [13] build the wav2letter [18] acoustic model by using the pre-trained embeddings as input features instead of log-mel filterbanks. Wav2vec 2.0 [14] and HuBERT [10] pre-train the transformer based encoder using the self-supervised loss, add a randomly initialized output layer on top and fine-tune with the CTC loss [17]. More recent research has shown that joint training with both supervised and unsupervised losses during the pre-training/fine-tuning stage or as a single training process helps improve the ASR performance. The initial UniSpeech work [19] demonstrates that representations learned during pre-training can be improved if the self supervised contrastive loss is combined with phonetic CTC loss, and the following Unispeech at scale work [20] demonstrates better representations from the pre-

*The author contributed to this work during an internship at Amazon

training stage for the downstream ASR task when combining the contrastive loss and the transducer loss. [21] alternatively minimizes an unsupervised masked CPC loss and a supervised CTC loss. This single-stage method is shown to match the performance of the two-stage wav2vec 2.0 on the Librispeech 100-hours dataset. [22] uses multitask learning comprising of supervised CTC, attention and self-supervised reconstruction losses to directly train acoustic models under low-resource settings. [23] explores the benefit of combining the supervised RNN-T loss [24], the self-supervised contrastive loss and masked language modeling (MLM) losses during different training stages. In this paper, we demonstrate benefits of our proposed method mainly on the conventional 2-stage training scheme. We additionally try the joint training scheme on one ASR task during the ablation study and demonstrate gains similar to what is reported in literature.

2. METHOD

2.1. Contrastive predictive coding

The left part of Figure 1 gives an overview of conventional CPC representation learning approach. Given frames of audio features $\mathbf{x}_t \in \mathcal{X}$, we first apply the feature encoder network $f_{enc} : \mathcal{X} \mapsto \mathcal{Z}$ to map the input sequence to a sequence of latent feature representations $\mathbf{z}_t \in \mathcal{Z} = f_{enc}(\mathbf{x}_t)$. An autoregressive context network $f_{ar} : \mathcal{Z} \mapsto \mathcal{C}$ summarizes all $\mathbf{z}_{\leq t}$ in the latent space and produces a contextual latent representation $\mathbf{c}_t = f_{ar}(\mathbf{z}_{\leq t})$

Both the feature encoder network and the autoregressive context network are trained to optimize the contrastive loss defined in Equation 1 based on Noise-Contrastive Estimation (NCE) [25] for each step k , which equivalently maximizes the mutual information between \mathbf{c}_t and the latent representation \mathbf{z}_{t+k} that is k steps in the future [11].

$$\mathcal{L}_k = -\frac{1}{T-k} \sum_{t=1}^{T-k} \log \frac{\exp(\mathbf{z}_{t+k}^\top h_k(\mathbf{c}_t)/\kappa)}{\sum_{\tilde{\mathbf{z}} \in \mathcal{Z}} \exp(\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_t)/\kappa)} \quad (1)$$

where $\tilde{\mathbf{z}}$ is a set of negative samples sampled from the same audio example to represent the imposter distribution, $h_k(\mathbf{c}_t) = W_k \mathbf{c}_t + \mathbf{b}_k$ is a step-specific affine transformation applied to \mathbf{c}_t for each step k , and κ is the temperature. We optimize the final contrastive loss \mathcal{L}_C by averaging \mathcal{L}_k over the next K steps:

$$\mathcal{L}_C = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_k \quad (2)$$

2.2. Guided contrastive predictive coding model

CPC learns representations from the complete data distribution, which may not be optimal for the downstream ASR task. In this paper, we propose to provide weak guidance for the contrastive loss. This weak guidance is provided in the form of posteriors from a prior-knowledge model learned from a small labeled dataset, and we use a monophone classifier for

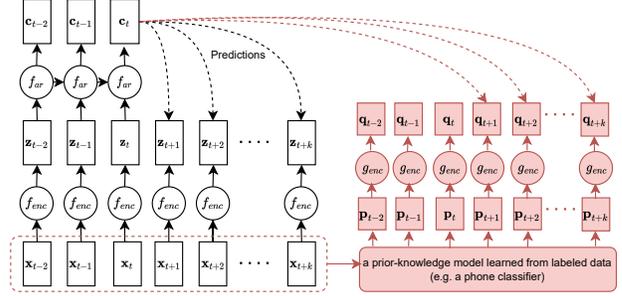


Fig. 1: Illustration of conventional Contrastive Predictive Coding (CPC) representation learning approach (left part) and our proposed Guided CPC (GCPC) method (right part in red). Parameters of the prior-knowledge model are fixed during training. \mathbf{p}_t is a sequence of logits from a monophone classifier in our experiments.

experimentation in the paper. As shown in the right part of Figure 1, we use an additional encoder network $g_{enc} : \mathcal{P} \mapsto \mathcal{Q}$ to map the sequence of unnormalized posteriors (logits) \mathbf{p}_t to a sequence of latent representations $\mathbf{q}_t \in \mathcal{Q} = g_{enc}(\mathbf{p}_t)$, and then optimize the guided contrastive loss \mathcal{L}_C^{guided} defined in Equation 4.

$$\mathcal{L}_k^{guided} = -\frac{1}{T-k} \sum_{t=1}^{T-k} \log \frac{\exp(\mathbf{q}_{t+k}^\top h_k(\mathbf{c}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \in \mathcal{Q}} \exp(\tilde{\mathbf{q}}^\top h_k(\mathbf{c}_t)/\kappa)} \quad (3)$$

$$\mathcal{L}_C^{guided} = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_k^{guided} \quad (4)$$

During training, parameters of the prior-knowledge model are fixed. We hypothesize that representations \mathbf{c}_t learned through this new technique could capture more phone discriminative characteristics since optimizing the guided contrastive loss helps maximizing the mutual information between \mathbf{c}_t and transformation of phone posteriors \mathbf{q}_{t+k} . Thus, \mathbf{c}_t might be more aligned with the downstream ASR task and serve as a better initialization point.

2.3. Contrastive pre-training for RNN-T ASR

We use an RNN-T [24] based ASR system for our experiments. The RNN-T model consists of an encoder, a prediction network and a joint network as shown in Figure 2a. Let $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y})\}$ denote a single example from a training corpus where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ is a sequence of speech features and $\mathbf{Y} = \{y_1, y_2, \dots, y_U\}$, $y_u \in \mathcal{V}$ is a sequence of tokens from the vocabulary \mathcal{V} (e.g. word pieces) representing the labels. The encoder maps each frame of the input speech features \mathbf{x}_t to a hidden state \mathbf{h}_t^{enc} . The prediction network takes the embedding vector of the previous non-blank token y_{u-1} and generates the hidden state \mathbf{h}_u^{pred} . The joint network is a feed-forward network that combines the outputs of the encoder and the prediction network to predict the conditional distribution over the next possible token $\tilde{y}_i \in \mathcal{V} \cup \langle blk \rangle$, where $\langle blk \rangle$ denotes the blank symbol. The RNN-T loss is computed

by marginalizing over all possible blank-augmented token sequences $\tilde{\mathbf{Y}} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{T+U}\}$ aligned with each original token sequence \mathbf{Y} and feature sequence \mathbf{X} :

$$\mathcal{L}_{RNN-T} = - \sum_{(\mathbf{x}, \mathbf{Y}) \in \mathcal{D}} \log \sum_{\tilde{\mathbf{Y}}} \prod_{i=1}^{T+U} P(\tilde{y}_i | \mathbf{X}_{1:t_i}, \mathbf{Y}_{0:u_{i-1}}) \quad (5)$$

where the index i in $\tilde{\mathbf{Y}}$ is mapped to the index u_i in \mathbf{Y} and the index t_i in \mathbf{X} .

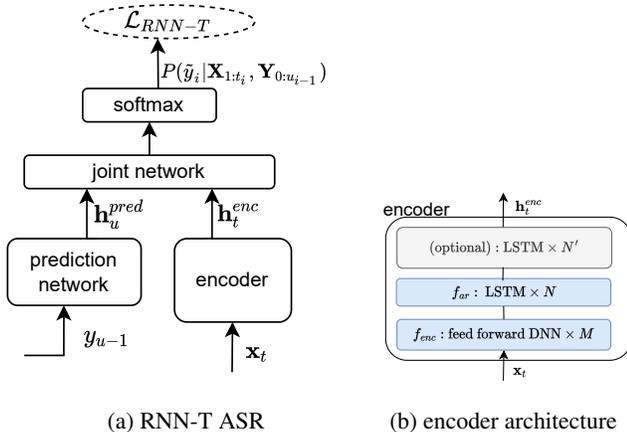


Fig. 2: Overview of the RNN-T based ASR system and architecture of the encoder used for experimentation.

After pre-training with the CPC or GCPC approach, The feature encoder network f_{enc} and the autoregressive context network f_{ar} are used to initialize encoder of the RNN-T ASR model in our experiments as shown in Figure 2b. The remaining parts of the RNN-T model are randomly initialized before the final supervised training stage. When initializing the RNN-T encoder, we experiment with initializing the entire RNN-T encoder as well as initializing part of the RNN-T encoder. During the RNN-T supervised training stage, we additionally experiment with joint training combining the supervised RNN-T loss and the self-supervised contrastive loss.

3. EXPERIMENTAL SETUP

3.1. Datasets

We explore our approach on three datasets in different languages; two in-house far-field corpora of de-identified utterances in German and French from a voice assistant, and one public dataset, Librispeech [26] in English. When experimenting with the Librispeech data, we use the Libri-Light [27] dataset for self-supervised pre-training. The numbers of hours for each dataset are summarized in Table 1 below.

The test sets for the German and French ASR tasks consist of de-identified utterances from a voice assistant, similar to the training data. For the English ASR task, we report results on the Librispeech test-clean and test-other test sets. The amount of test data is also summarized in Table 1.

Table 1: Summary of datasets for experimentation.

Language	Train (hrs)		Evaluation (hrs)
	unlabeled	labeled	
German (in-house)	142k	25k	15
French (in-house)	30k	15k	9.5
English (Librispeech)	60k	960	5.4/5.1

3.2. Model and training details

The encoder of the RNN-T model for our experimentation contains 3 feed-forward (dense) layers of size 512 with ReLU non-linearity, followed by 8 LSTM layers with 1024 units for the German and French ASR tasks, and 6 LSTM layers with 1024 units for the Librispeech ASR task. For experiments that initialize the RNN-T encoder using CPC/GCPC pre-training methods, the feed-forward layers are initialized from the feature encoder network f_{enc} of the CPC/GCPC pre-trained model illustrated in Figure 1, and the LSTM layers are initialized from the autoregressive context network f_{ar} . The RNN-T model also contains a single-layer LSTM prediction network with 1024 units, and a single-dense-layer joint network. When training the RNN-T ASR model, we use a sentence piece model [28] containing 4000 sentence pieces which are trained with the corresponding monolingual dataset for each task.

The acoustic features used are 256-dimensional log short-time fourier transform (log-STFT), computed on a 25ms window with a frame shift of 10ms. The input log-STFT features from 3 consecutive frames are stacked for an effective frame size of 30ms, so that the final RNN-T input feature is of the dimension 768. There is no external language model used for any of the ASR tasks.

When computing the guided contrastive loss, extra dense layers g_{enc} are added on top of phone logits, and are updated during the pre-training stage. Our ablation studies described in Section 4.3.3 will demonstrate the impact of these extra feed-forward layers. The number of steps K (in Equation 2) used for pre-training is 4.

4. RESULTS

4.1. Phone Classification

The phone classification model is a 5-layer LSTM model with 768 units in each layer. It is trained with the standard cross-entropy loss. Frame-level monophone targets for the internal datasets are generated from the 1-best decoding output of a hybrid LSTM-HMM ASR model. For Librispeech, we obtain the frame-level phone targets using the Montreal forced aligner¹. Frame-level accuracy for these phone classifiers are shown in Table 2. These phone classifiers are then used as the prior knowledge models for the guided contrastive loss based pre-training.

¹<https://github.com/CorentinJ/librispeech-alignments>

Table 2: Frame-level phone accuracy (monophones) for the phone classifier built for each language.

Language	# of phones	Accuracy (%)
German	55	75.94
French	45	81.39
Librispeech dev-clean	72	80.58

4.2. ASR results

The results for each ASR task (German, French and English) with different encoder pre-training methods are reported in Tables 3 and 4. The baseline RNN-T ASR models (B_{G_1} , B_{F_1} , B_{L_1}) are trained from scratch. Encoders of M_{G_2} , M_{F_2} and M_{L_2} are initialized with the encoder pre-trained with standard CPC method explained in Section 2.1. Encoders of M_{G_3} , M_{F_3} and M_{L_3} are initialized with our proposed Guided CPC (GCPC) method explained in Section 2.2. For a more comprehensive comparison, we also pre-train the encoder with the cross-entropy loss using phone posteriors (PCE) obtained from the phone classifier described in Section 4.1, and these results are reported for models M_{G_1} , M_{F_1} , M_{L_1} . Note we only report relative WER reduction (WERR) on tasks using the internal data shown in Table 3, but we do report the absolute WER on the Librispeech task shown in Table 4.

Table 3: Relative Word Error Rate Reduction (WERR) w.r.t RNN-T ASR baseline when using different encoder pre-training methods on German and French ASR tasks. Negative WERR indicates a degradation. Best numbers in bold.

Model	RNN-T encoder initialization	WERR%
<i>German ASR task</i>		Test German
B_{G_1} ²	-	0.00
M_{G_1}	PCE	2.11
M_{G_2}	CPC	2.96
M_{G_3}	GCPC	4.44
<i>French ASR task</i>		Test French
B_{F_1}	-	0.00
M_{F_1}	PCE	-0.50
M_{F_2}	CPC	1.01
M_{F_3}	GCPC	6.55

Overall, the standard contrastive pre-training on the RNN-T encoder reduces WER on our internal German and French ASR tasks by 2.96% and 1.01% respectively relative to the baseline, while our proposed guided contrastive pre-training method brings higher relative WER reductions (4.44% and 6.55% on the German and French ASR tasks respectively). The standard phone cross-entropy pre-training leads to a worse WER than both the CPC and GCPC pre-training, which indicates the importance of the contrastive term in the pre-training loss function. On the Librispeech task, we

²Table Notation: B is baseline, M is experimental model, letter in subscript (G/F/E) is language, and numeral in the subscript is the experiment id. The extra letter in subscripts in Section 4.3 refers to the ablation study id.

observe similar trend under the Test-Clean condition. However, under the Test-Other condition, we don’t see benefits of the GCPC pre-training method. From all our experimental results, we observe that for the tasks with larger labeled training dataset, the gain with the pre-training techniques are smaller.

Table 4: Word Error Rate (WER) when using different encoder pre-training methods on Librispeech ASR task. Best numbers in bold.

Model	RNN-T encoder initialization	WER% (WERR%)	
		Test-Clean	Test-Other
<i>Librispeech ASR task</i>			
B_{L_1}	-	6.74	17.44
M_{L_1}	PCE	7.22 (-7.12)	19.25 (-10.38)
M_{L_2}	CPC	5.77 (14.39)	16.05 (7.97)
M_{L_3}	GCPC	5.70 (15.43)	16.21 (7.05)

4.3. Ablation studies

In order to identify the best training scheme, we perform ablation studies on the internal German dataset. We only show results for the German ASR task, but the WER trends for the ASR models for all 3 languages on both the development and test data were similar. The training scheme was tuned on a held out development set, but we show our results on the test set so the numbers in sections 4.2 and 4.3 are comparable.

4.3.1. Two-stage training versus joint training

Since more recent research demonstrates that joint training with both supervised and self-supervised losses can directly optimize the ASR performance [23], we experiment with joint training combining the supervised RNN-T loss \mathcal{L}_{RNN-T} and the self-supervised contrastive loss \mathcal{L}_C as shown in Table 5. Note that we use a different baseline B_{G_2} with a relatively small batch size due to memory limitations of the GPU devices used for experimentation. We demonstrate that joint training from scratch ($M_{G_{A1}}$) brings a relative WERR of 3.32%, which is slightly better than the 2.96% relative WERR obtained from the conventional two-stage training scheme (M_{G_2} in Table 3). On top of this, contrastively pre-training the RNN-T encoder ($M_{G_{A2}}$) doesn’t seem to further improve the WER. Considering similar WERRs from these two training schemes and the high memory consumption by the joint training which causes training instability, we use the conventional two-stage training scheme for our proposed method.

Table 5: Relative Word Error Rate Reduction (WERR) w.r.t RNN-T ASR baseline with joint training on the German ASR task.

Model	RNN-T encoder initialization	Loss	WERR% Test German
B_{G_2}	random	\mathcal{L}_{RNN-T}	0.00
$M_{G_{A1}}$	random	$\mathcal{L}_{RNN-T} + \mathcal{L}_C$	3.32
$M_{G_{A2}}$	CPC	$\mathcal{L}_{RNN-T} + \mathcal{L}_C$	1.95

Table 6: Relative Word Error Rate Reduction (WERR) shown w.r.t RNN-T ASR baseline when experimenting with different pre-training stages on the German ASR task.

Model	RNN-T encoder						WERR%
	$(f_{enc} + f_{ar})$			remaining layers			Test German
	architecture	initialization	trainable	architecture	initialization	trainable	
B_{G_1}	DNN \times 3 + LSTM \times 8	random	Yes	-	-	-	0.00
M_{G_1}	DNN \times 3 + LSTM \times 8	CPC	Yes	-	-	-	2.96
$M_{G_{B_1}}$	DNN \times 3 + LSTM \times 2	CPC	No	LSTM \times 6	random	Yes	-10.36
$M_{G_{B_2}}$	DNN \times 3 + LSTM \times 2	CPC	yes	LSTM \times 6	random	Yes	-0.85
$M_{G_{B_3}}$	DNN \times 3 + LSTM \times 2	$M_{G_{B_1}}$	yes	LSTM \times 6	$M_{G_{B_1}}$	Yes	-1.69

4.3.2. Optimizing the pre-training stage

All the pre-training experiments reported in Section 4.2 utilize an RNN-T encoder with all its layers pre-trained. We also experiment with different pre-training stages and the resulting WERRs are shown in Table 6. For model $M_{G_{B_1}}$, we only pre-train the RNN-T encoder up to the second LSTM layer with the contrastive loss. We then freeze those pre-trained layers and randomly initialize the rest 6 LSTM layers during the supervised RNN-T training stage. The supervised training stage for model $M_{G_{B_2}}$ is similar to that of model $M_{G_{B_1}}$, except that we don't freeze the pre-trained layers. Model $M_{G_{B_3}}$ further tunes model $M_{G_{B_1}}$ by making all layers trainable. According to WERRs reported in Table 6, pre-training the entire RNN-T encoder gives the best WER. Therefore, we adopt this method of pre-training for all other experiments in the paper.

4.3.3. Hyperparameter tuning for contrastive pre-training

We tune two hyperparameters for guided contrastive pre-training. The first hyperparameter is the temperature for the loss function, κ . Second, instead of using phone logits directly to compute the contrastive loss, we experiment with adding trainable feed-forward layers (g_{enc} illustrated in Figure 1) on top of the phone logits before computing the contrastive loss. The intuition behind this is to let the model learn derivative features from the phone posteriors, which might be more suitable for the downstream ASR task. Relative WERRs with different hyperparameters are shown in Table 7. We note that when we use phone logits directly as latent representations, the WER degrades by 16.49% relative to the baseline ($M_{G_{C7}}$ vs B_{G_1}). Using the learnable feed-forward layers g_{enc} to generate latent representations is critical to the GCPC pre-training. Based on the tuning results, using 2 learnable feed-forward layers to generate latent representations for contrastive learning gives the largest WER reduction. Finally, the tuning of the temperature parameter κ shows that 0.01 is the optimal value for the German ASR GCPC pre-training, and 0.1 is the optimal value for the conventional CPC pre-training. Note that we experimented with the outputs of the intermediate layer of the phone classifier as inputs to compute contrastive loss and found that they lead to a worse ASR performance compared to logits followed by trainable feed-forward layers (results not in the Table).

Table 7: Relative Word Error Rate Reduction (WERR) shown w.r.t RNN-T ASR baseline when tuning hyperparameters for contrastive pre-training on the German ASR task.

Model	RNN-T encoder initialization			WERR%
	method	κ	g_{enc}	
B_{G_1}	-	-	-	0.00
M_{G_1}	CPC	0.1	-	2.96
$M_{G_{C1}}$	CPC	0.02	-	0.42
$M_{G_{C2}}$	CPC	0.01	-	-1.06
$M_{G_{C3}}$	GCPC	0.1	DNN \times 2	-18.18
$M_{G_{C4}}$	GCPC	0.02	DNN \times 2	3.59
M_{G_3}	GCPC	0.01	DNN \times 2	4.44
$M_{G_{C5}}$	GCPC	1e-5	DNN \times 2	1.06
$M_{G_{C6}}$	GCPC	0.02	DNN \times 3	2.54
$M_{G_{C7}}$	GCPC	0.02	None	-16.49

4.3.4. Joint contrastive pre-training

We perform an additional experiment where the pre-training objective \mathcal{L}_C^{joint} consists of both the regular contrastive loss (\mathcal{L}_C defined in Equation 2) and the guided contrastive loss (\mathcal{L}_C^{guided} defined in Equation 4).

$$\mathcal{L}_C^{joint} = \mathcal{L}_C + \mathcal{L}_C^{guided} \quad (6)$$

According to results in Table 8, we can see that guided contrastive pre-training performs better than both regular contrastive as well as the joint contrastive pre-training in terms of the final WER.

Table 8: Relative Word Error Rate Reduction (WERR) w.r.t RNN-T ASR baseline when using individual contrastive loss and joint contrastive loss for pre-training on the German ASR task.

Model	RNN-T encoder initialization	WERR%
B_{G_1}	-	0.00
M_{G_2}	CPC	2.96
M_{G_3}	GCPC	4.44
$M_{G_{D1}}$	CPC+GCPC	1.69

4.4. Analysis of representations

To better understand the gains from using the guided contrastive loss, we visualize the output of the RNN-T encoder

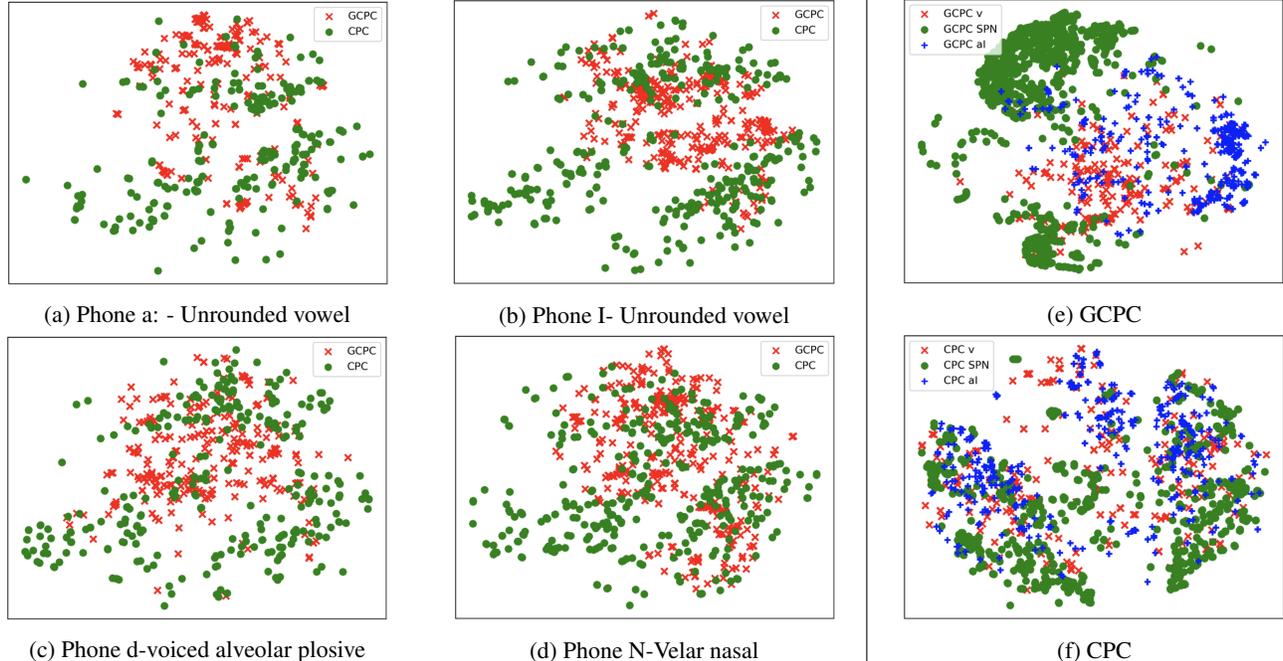


Fig. 3: 3a-3d show the t-SNE visualizations of 4 German phones from CPC and guided CPC pre-trained RNN-T encoders. 3e-3f present the t-SNE visualizations for 3 phones on the same plot (v, al and SPN (speech like noise)) obtained from GCPC and CPC pre-trained RNN-T encoders respectively, indicating better phone separation with GCPC pre-trained encoder.

pre-trained using various methods with the German data. We take the frame level representations for a subset of the development set from the pre-trained RNN-T encoder and apply the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [29] to reduce the embedding dimension from 1024 to 2. We plot these 2-dimensional embeddings obtained from CPC and guided CPC per monophone.

Figure 3a-3d show the t-SNE visualizations for four different monophones for the German ASR task. We use the X-SAMPA phone-set for the experiments. For the visualization, we choose two vowels, one plosive and one nasal phone. From these figures, we observe that the representations for each of the phone classes are more closely clustered when using the guided CPC pre-training as compared to using traditional CPC pre-training. Figure 3e-3f show the t-SNE visualization for three different phones on the same plot from CPC and GCPC pre-trained encoders. Although there is overlap between the different phone clusters, we observe better separation between frames from the different phones with the GCPC pre-trained encoder as compared to the CPC pre-trained encoder. The separation is more pronounced between voiced frames compared to frames with speech like noise (SPN). Table 9 breaks down the relative reduction by substitution, deletion and insertion errors on the German ASR task. The breakdown shows that CPC pre-training degrades the insertion error rate by 3.45% relative, while GCPC pre-training improves the insertion error rate by 3.39%. The better separation between frames with speech like noise and other speech frames explains this improvement in insertion errors.

Table 9: Relative Error Rate Reduction in substitution (SUBR), insertion (INSR) and deletion (DEL) errors w.r.t RNN-T ASR baseline when using different pre-training methods for the German ASR Task.

Model	RNN-T encoder initialization	SUBR%	INSR%	DEL%
B_{G_1}	-	0.00	0.00	0.00
M_{G_2}	CPC	1.45	-3.45	8.57
M_{G_3}	GCPC	2.55	3.39	9.29

5. CONCLUSION

We have shown that injecting prior knowledge in the form of phone posteriors during the contrastive pre-training stage helps improve the performance of three downstream ASR tasks compared to applying regular contrastive pre-training. On the German and French ASR tasks, our method gives a 4.44% and 6.55% relative WERR respectively, while the regular CPC method just brings a 2.96% and 1.01% relative WERR. On the Librispeech ASR task (test-clean), our pre-training method reduces the WER relatively by 15.43% compared to training the ASR model from scratch. From the t-SNE visualizations of the pre-trained embeddings using CPC and guided CPC methods, we observe closer clustering among frames belonging to the same phone with guided contrastive pre-training, and better separation between speech frames and frames with speech-like noise, leading to higher reduction in insertion error rates with the guided contrastive pre-training.

6. REFERENCES

- [1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [2] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of NAACL-HLT*, 2018, pp. 2227–2237.
- [3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, “Improving language understanding by generative pre-training,” 2018.
- [4] Yi-Chen Chen, Sung-Feng Huang, Hung-yi Lee, Yu-Hsuan Wang, and Chia-Hao Shen, “Audio word2vec: Sequence-to-sequence autoencoding for unsupervised learning of audio segmentation and representation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1481–1493, 2019.
- [5] Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord, “Unsupervised speech representation learning using wavenet autoencoders,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.
- [6] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, “An Unsupervised Autoregressive Model for Speech Representation Learning,” in *Proc. Interspeech 2019*, 2019, pp. 146–150.
- [7] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6419–6423.
- [8] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6429–6433.
- [9] Shaoshi Ling, Julian Salazar, Yuzong Liu, Katrin Kirchhoff, and AWS Amazon, “Bertphone: Phonetically-aware encoder representations for utterance-level speaker and language recognition,” in *Proc. Odyssey*, 2020, pp. 9–16.
- [10] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [12] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, “wav2vec: Unsupervised Pre-Training for Speech Recognition,” in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.
- [13] Alexei Baevski, Steffen Schneider, and Michael Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in *International Conference on Learning Representations*, 2019.
- [14] 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*, vol. 33, 2020.
- [15] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu, “W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.
- [16] Chengyi Wang, Yiming Wang, Yu Wu, Sanyuan Chen, Jinyu Li, Shujie Liu, and Furu Wei, “Supervision-guided codebooks for masked prediction in speech pre-training,” *arXiv preprint arXiv:2206.10125*, 2022.
- [17] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [18] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, “Wav2letter: an end-to-end convnet-based speech recognition system,” *arXiv preprint arXiv:1609.03193*, 2016.
- [19] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang, “Unispeech: Unified speech representation learning with labeled and unlabeled data,” in *Proceedings of the ICML*, 2021, pp. 10937–10947.
- [20] Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Yao Qian, Kenichi Kumatani, and Furu Wei, “Unispeech at scale: An empirical study of pre-training method on large-scale speech recognition dataset,” *arXiv preprint arXiv:2107.05233*, 2021.

- [21] Chaitanya Talnikar, Tatiana Likhomanenko, Ronan Collobert, and Gabriel Synnaeve, “Joint masked cpc and ctc training for asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3045–3049.
- [22] Srinivasa Raghavan and Kumar Shubham, “Hybrid unsupervised and supervised multitask learning for speech recognition in low resource languages,” in *Proc. Workshop on Machine Learning in Speech and Language Processing*, 2021.
- [23] Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, and Tara N Sainath, “Joint unsupervised and supervised training for multilingual asr,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6402–6406.
- [24] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [25] Michael Gutmann and Aapo Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.
- [26] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [27] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., “Libri-light: A benchmark for asr with limited or no supervision,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.
- [28] Taku Kudo and John Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 66–71.
- [29] Laurens Van der Maaten and Geoffrey Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.