DYNAMIC INTERMEDIATE LOSS FOR FINE-TUNING SELF-SUPERVISED SPEECH MODELS

Khai Le-Duc[*1], *Christoph Lüscher*[*1,2], *Ralf Schlüter*[1,2]

[1]Lehrstuhl Informatik 6 - Machine Learning and Human Language Technology Group,
Computer Science Department, RWTH Aachen University, 52074 Aachen, Germany
[2]AppTek GmbH, 52062 Aachen, Germany
dle@i6.informatik.rwth-aachen.de, {luescher,schlueter}@hltpr.rwth-aachen.de

ABSTRACT

Self-Supervised Learning (SSL) has expedited Automatic Speech Recognition (ASR) progress in low-resource, real-world settings. In this work, we propose a simple regularization method, called Dynamic Intermediate Loss (*DiLoss*) for fine-tuning wav2vec 2.0 pre-trained models. We show the robustness of *DiLoss* on 3 types of loss functions: cross-entropy, focal, and PolyLoss. To the best of our knowledge, we are the first to transfer the novel PolyLoss from computer vision tasks to ASR application. Also, we propose 4 different training schedules for *DiLoss*: Dynamic Layer, Dynamic Loss Scale, Dynamic Loss Removal and Parameter Sharing. We then combine some of them to show the suitability. We show the robustness of our method by reporting the improvement of WERs on all 3 pre-trained models on VietMed - a public real-world medical ASR dataset. Notable is the reduction of WER from 49.5% to 29.0% (a relative reduction of 42% compared to the baseline).

***Index Terms*—** One, two, three, four, five

1. INTRODUCTION

ASR systems have made significant progress, achieving notable performance, particularly in tasks with abundant training data. Nonetheless, challenges persist in the form of diverse acoustic and recording conditions, speaking styles, and a shortage of labeled training data, hindering the development of precise models. Recently, SSL has made it possible to leverage unlabeled audio data, which is more affordable, considerably reducing the reliance on transcribed data [1, 2, 3]. These pre-trained models are typically quite large, trained on substantial amounts of unlabeled data, demanding advanced hardware, several weeks or months, and a substantial financial budget for training. So, deciding to directly fine-tune large publicly available pre-trained models instead of going through the whole process of pre-training a new one for specific needs is seen as a more cost-effective and time-saving choice.

However, simply vanilla fine-tuning does not always ensure a good result, especially in a challenging real-world task where problems like domain shift, natural flow of speech, various acoustic conditions and recoding devices, medical domain, and sampling rate mismatch. [4] showed modifications of the feature extractor and a freezing scheme in fine-tuning pre-trained models to tackle sampling rate mismatch for low-resource ASR. [5] studied individual components of wav2vec 2.0 architectures which motivated a modification to the fine-tuning protocol in the very low-resource setting.

Deep transformers always suffer from gradient vanishing, especially for paramete [6]. Previously, the concept of intermediate loss has been suggested to incorporate additional loss functions after some layers, aiming to enhance gradient updates. This is particularly emphasized through the use of intermediate CTC loss to improve the performance of the CTC model [7, 8]. For HMM-based hybrid ASR systems, intermediate cross-entropy loss has been employed in the training of deep Transformer-based acoustic models [9], and Conformer-based models [10].

In this work, we propose Dynamic Intermediate Loss (*DiLoss*) - a new variant of intermediate loss - to help fine-tuning wav2vec 2.0 pre-trained models. We then empirically evaluate our proposed *DiLoss* on the public medical ASR dataset - *VietMed*. Our key contributions are summarized as follows:

- To the best of our knowledge, we are the first to evaluate the novel loss function in computer vision - PolyLoss - in the ASR task.
- We show the robustness of *DiLoss* on 3 types of loss functions: cross-entropy, focal, and PolyLoss.
- We propose 4 different training schedules for *DiLoss*: *Dynamic Layer*, *Dynamic Loss Scale*, *Dynamic Loss Removal* and *Parameter Sharing*, and combine some of them to show the suitability.

[*]Equal contribution

All code are published online[1].

2. METHODS

In the process of constructing and refining the *DiLoss* for the wav2vec 2.0-based model, we explore various training methods inspired by existing literature. This exploration has proven beneficial in enhancing the Word Error Rate (WER) in the section Experimental Results.

Time down-sampling in feature extractor: A main challenge that often occurs when fine-tuning publicly available pre-trained models on existing dataset is the sampling rate mismatch. In our situation, the feature extractor of 16 kHz pre-trained models can be modified to handle 8 kHz sampled data while still producing representations with the same 20 ms frame shift. This adjustment involves halving the stride of a convolutional layer in the feature extractor. Then we will receive features at the desired frame rate while reducing the down-sampling factor from the waveform to the feature frames by a factor of 2.

Time up-sampling in DiLoss: Training deeper neural networks demands thoughtful strategies to achieve convergence. Incorporating intermediate loss at various layers has proven to be beneficial in enhancing training stability [7, 8]. However for wav2vec 2.0 architecture, a modified design is needed in order to successfully integrate intermediate loss. In both scenarios, whether there is a sampling rate mismatch or not, the final Transformer layer must be upsampled before being fed into a Softmax layer for classification. Not doing so would result in a mismatch in the number of targets when calculating the frame-wise loss objective function. The feed-forward layer (FFW) in *DiLoss* therefore needs to be upsampled.

Loss function in DiLoss: Loss function in *DiLoss* is easy to be interchanged by different types of loss function. In the scope of this paper, we only evaluate on 3 types of loss function: cross-entropy, focal loss [11], and PolyLoss [12]. Focal loss is a technique that adjusts the cross-entropy objective function to reduce the emphasis on well-classified targets, redirecting the model's attention toward mis-classified targets. PolyLoss is a novel loss motivated by how functions can be approximated via Taylor expansion. By incorporating just one additional hyperparameter and including a single line of code, the Poly-1 formulation surpasses the performance of cross-entropy loss and focal loss in tasks such as 2D image classification, instance segmentation, object detection, and 3D object detection.

Dynamic Layer: [5] has demonstrated that information in intermediate layers of pre-trained models proves to be more useful for ASR than the output from the final layers. Inspired by this findings, we want to leverage different information contained in different intermediate layers to help the convergence of the model. Therefore, we propose a dynamic training schedule by placing intermediate loss on different layers while training.

Dynamic Loss Scale: Also inspired by the findings above, we want to increase or decrease the weight of information in intermediate layers that contribute to the total loss. To do so, we simply change the loss scale while training the network.

Dynamic Loss Removal: At some epoch, the existence of intermediate loss might hurt the generalization of the model. Therefore, we remove the intermediate loss for some epochs during training. In our experiments, we only conduct *Dynamic Loss Removal* at final epochs in an attempt to not substantially increase the trainable parameters in the final model as the vanilla [7, 8].

Parameter Sharing: Just like *Dynamic Loss Removal*, *Parameter Sharing* can help decrease the size of the model. We explore how *Parameter Sharing* in both the intermediate loss layers and the wav2vec 2.0 FFW layers impacts the overall performance.

3. EXPERIMENTS

We used RETURNN [13] for supervised training and Fairseq [14] for self-supervised wav2vec 2.0 training. Decoding was performed with RASR [15]. Fairseq models were converted to RETURNN models with our PyTorch-to-RETURNN toolkit[2].

3.1. Data and Pre-trained Models

We used *VietMed* - a public real-world medical ASR dataset for our training [16]. We also employed 2 models pre-trained on large-scale 8kHz Vietnamese data from [16], *w2v2-Viet* and *XLSR-53-Viet*. Moreover, we used *XLSR-53* model [2] pre-trained on 56k hours of 16kHz multilingual data. All 3 baselines have been presented in [16].

3.2. DiLoss Setups

For focal loss, we used the default focal value 2.0. For PolyLoss, we used Poly-1 formulation. To find the right PolyLoss value (ϵ), we simply tried grid search, with $\epsilon \in \{-1, 0, 1, ..., 7\}$. We found that $\epsilon \in \{0, 1, ..., 5\}$ generated almost the same good results, so we sticked to $\epsilon = 2$ for all our experiments.

4. EXPERIMENTAL RESULTS

4.1. Studies on intermediate loss design

As shown in Table 1, putting the intermediate loss on layer 3 yields the best result. Putting the intermediate loss on layer

[1] https://github.com/rwth-i6/returnn-experiments

[2] https://github.com/rwth-i6/pytorch-to-returnn

	Layer	Loss scale	dev	test
CE	3	0.1	26.1	29.1
		0.3	26.7	29.9
Focal	0	0.1	26.2	29.0
	3		26.1	28.9
	7		28.1	31.2
	3	0.3	26.7	29.5
PolyLoss	3	0.1	26.1	28.9

Table 1. Results of CE, focal and PolyLoss using *XLSR-53-Viet* pre-trained model for different layer positions and loss scale values.

7 negatively affects the model convergence. This negative effect also happened when we trained on our in-house dataset. Here we conclude that the intermediate loss should be put in the middle layer of the encoder to gain the best result.

In original settings where Transformer supervised-only training was done [7, 8], the CTC intermediate loss used a default loss scale of 0.3. However, as shown in Table 1, loss scale of 0.1 works better compared to loss scale of 0.3 on wav2vec 2.0 architecture. The effect of loss scale 0.1 has been confirmed on both CE and focal loss.

4.2. Dynamic Layer

	Phase 1	Phase 2	Lr schedule	dev	test
Focal	Layer 3, scale = 0.1	Layer 0	Lr reset	26.2	29.3
		Layer 2		26.1	-
		Layer 5		26.1	-
		Layer 0	Continued Lr	26.0	-
		Layer 2		26.1	-
		Layer 5		26.4	-
		Layer 7		27.1	-

Table 2. Dynamic Layer results, using *XLSR-53-Viet* pre-trained model, fine-tuning with focal loss. We only changed the layer in phase 2 of the training, in which the models were tuned with either learning rate reset or continued learning rate.

As shown in Table 2, we report all experiments' performance degraded when moving the intermediate loss to another layer in the 2nd phase of training. In the 2nd phase, the weights of the intermediate loss were re-initialized using Xavier Initialization [17], which probably was the reason why it hurt the performance. Therefore, we expect in the 2nd phase of training, learned representations should be exactly transfered from the 1st phase in order to leverage the learned information. We leave this expectation for future work.

4.3. Dynamic Loss Scale

As shown in Table 3, compared to constant-loss-scale training, we report all experiments' performance improved using

	Phase 1	Phase 2	Lr schedule	dev	test
Focal	Layer 3, scale = 0.1	Layer 3, scale=0.3	Lr reset	26.0	28.9
		Layer 3, scale=0.3	Continued Lr	25.9	28.8
		Layer 3, scale=0.5		26.0	28.8

Table 3. Dynamic Loss Scale results: using XLSR-53-Viet pre-trained model, fine-tuning with focal loss. We only changed the loss scale in phase 2 of the training, in which the models were tuned with either learning rate reset or continued learning rate.

Dynamic Loss Scale. In our many experiments which we do not show here, we found that loss scale = 0.3 or 0.5 is a good value to directly tune without so much tuning efforts in the future.

4.4. Parameter sharing

	Layer	Loss scale	Paramshare	dev	test
Focal	0	0.1	Upsample + FFW	26.2	29.0
	3			25.9	28.8
	7			27.7	30.5
	3		Upsample	26.0	28.8
			FFW	26.0	28.7

Table 4. Parameter Sharing results: using XLSR-53-Viet pre-trained model, fine-tuning with focal loss. We applied Parameter Sharing in the single-phase training, either on Upsample layer or FFW layer or on both.

As shown in Table 4, we report all experiments' performance improved or unchanged with *Parameter Sharing* compared to not applying this strategy. In the case of not applying the intermediate loss to an optimal layer (e.g. to Layer 0 and Layer 7), the *Parameter Sharing* still boosted the model performance. Also shown in Table 4,the *Parameter Sharing* should be applied on both Upsample and FFW layer in order to achieve the best performance.

4.5. Dynamic Loss Removal

As shown in Table 6, *Dynamic Loss Removal* with continued learning rate even boosted performance of both *Parameter Sharing* and *Dynamic Loss Scale*. The learning rate reset negatively affected the *Dynamic Loss Removal*. We therefore conclude that, *Dynamic Loss Removal* could be effectively combined with *Parameter Sharing* or *Dynamic Loss Scale*.

Pre-trained model	Strategy	dev	test
XLSR-53-Viet	None	26.8	29.6
	+DiLoss	26.1	28.9
	+Paramshare+Dyn.Loss Rem.	**25.7**	28.8
	+Dyn.Loss Scale+Dyn.Loss Rem.	25.9	**28.6**
w2v2-Viet	None	45.3	49.5
	+DiLoss+Paramshare+Dyn.Loss Rem.	25.9	29.0
XLSR-53	None	45.2	51.8
	+DiLoss+Paramshare+Dyn.Loss Rem.	40.7	47.6

Table 5. Overall results: using 3 different pre-trained models, fine-tuning with focal loss. We combined some strategies that led to improvements of model performance.

	Training schedule	Lr schedule	dev	test
Focal	Paramshare	Reset Lr	26.3	28.9
		Continued Lr	25.7	28.8
	Dynamic Loss Scale		25.9	28.6

Table 6. Dynamic Loss Removal results: using XLSR-53-Viet pre-trained model, fine-tuning with focal loss. We only removed the intermediate loss in phase 2 of the training, in which the models were tuned with either learning rate reset or continued learning rate

4.6. Overall Results

As shown in Table 5, our proposed *DiLoss* improved WERs on all 3 baselines. Notable is the improvement on w2v2-Viet pre-trained model, which experienced the WER reduction from 45.3%, 49.5% to 25.9%, 29.0% (a relative reduction of more than 40%). For the XLSR-53 pre-trained model, even in a sampling rate mismatch scenario, we still report a WER reduction from 45.2%, 51.8% to 40.7%, 47.6% (a relative reduction of 8.1%). For the very well pre-trained model, we also report a WER reduction from 26.8%, 29.6% to 25.7%, 28.8% or 25.9%, 28.6% (a relative reduction of 3.8%).

5. CONCLUSION

In this work, we propose a novel regularization method, called Dynamic Intermediate Loss (*DiLoss*) for fine-tuning wav2vec 2.0 pre-trained models. We show that *DiLoss* could be well used with 3 types of loss functions: cross-entropy, focal, and PolyLoss. Also, we propose 4 different training schedules for *DiLoss*: *Dynamic Layer*, *Dynamic Loss Scale*, *Dynamic Loss Removal* and *Parameter Sharing*. We show the negative effect of *Dynamic Layer* and we propose a solution for future work. Furthermore, we show *Dynamic Loss Scale*, *Dynamic Loss Removal* and *Parameter Sharing* could all boost the WER performance of pre-trained models. The performance even becomes better thanks to the combination of *Dynamic Loss Removal* with either *Dynamic Loss Scale* or *Parameter Sharing*. Finally, we show the robustness of our method by reporting the improvement of WERs on all 3 pre-trained models on VietMed dataset. Notable is the reduction of WER from 49.5% to 29.0% (a relative reduction of 42% compared to the baseline).

6. REFERENCES

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, NIPS'20.

[2] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2426–2430.

[3] 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*, 2021.

[4] Peter Vieting, Christoph Lüscher, Julian Dierkes, Ralf Schlüter, and Hermann Ney, "Efficient utilization of large pre-trained models for low resource asr," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2023.

[5] Ankita Pasad, Ju-Chieh Chou, and Karen Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021.

[6] Ruchao Fan, Wei Chu, Peng Chang, and Abeer Alwan, "A ctc alignment-based non-autoregressive transformer for end-to-end automatic speech recognition,"

IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 1436–1448, 2023.

[7] Andros Tjandra, Chunxi Liu, Frank Zhang, Xiaohui Zhang, Yongqiang Wang, Gabriel Synnaeve, Satoshi Nakamura, and Geoffrey Zweig, "Deja-vu: Double feature presentation and iterated loss in deep transformer networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[8] Jaesong Lee and Shinji Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[9] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al., "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[10] Mohammad Zeineldeen, Jingjing Xu, Christoph Lüscher, Wilfried Michel, Alexander Gerstenberger, Ralf Schlüter, and Hermann Ney, "Conformer-based hybrid asr system for switchboard dataset," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[12] Zhaoqi Leng, Mingxing Tan, Chenxi Liu, Ekin Dogus Cubuk, Jay Shi, Shuyang Cheng, and Dragomir Anguelov, "Polyloss: A polynomial expansion perspective of classification loss functions," in *International Conference on Learning Representations*, 2021.

[13] Albert Zeyer, Tamer Alkhouli, and Hermann Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Annual Meeting of the Assoc. for Computational Linguistics*, 2018.

[14] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 48–53.

[15] David Rybach, Stefan Hahn, Patrick Lehnen, David Nolden, Martin Sundermeyer, Zoltán Tüske, Simon Wiesler, Ralf Schlüter, and Hermann Ney, "RASR - the RWTH Aachen University open source speech recognition toolkit," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011.

[16] Khai Le-Duc, Christoph Lüscher, Minh-Nghia Phan, Ralf Schlüter, and Hermann Ney, "Vietmed: A dataset and benchmark for automatic speech recognition of vietnamese in the medical domain," 2023.

[17] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.