Improving Automatic Speech Recognition with Decoder-Centric Regularisation in Encoder-Decoder Models

Anonymous ACL submission

Abstract

This paper introduces **De**coder-**C**entric **R**egularisation in **E**ncoder-**D**ecoder (DeCRED) architecture for automatic speech recognition, where auxiliary classifier(s) are introduced in layers of the decoder module. Leveraging these classifiers, we propose two decoding strategies that re-estimate the next token probabilities. Pilot experiments conducted on the independent in-domain datasets identify the suitable placement and weighting of the auxiliary classifiers, resulting in a consistent word-error-rate (WER) reduction of up to 9 % relative across different model sizes. Further experiments on a collection of multi-domain English datasets showed that DeCRED obtained competitive WERs as compared to Whisper-medium and outperformed OWSM v3; while relying only on a fraction of training data and model size. Finally, we also study the generalisation capabilities of DeCRED by evaluating on out-of-domain datasets, where we show an absoulte reduction of 2.7 and 2.9 WERs on AMI and Gigaspeech datasets respectively.

1 Introduction

One of the key challenges in automatic speech recognition (ASR) is the ability of the models to generalise to new or previously unseen domains. Large-scale training on multiple domains (Narayanan et al., 2018; Chan et al., 2021), data augmentation (Park et al., 2019), architecture-specific regularisation (Lee and Watanabe, 2021) are some of the strategies for improving the robustness of ASR systems. In recent years, we have seen a shift towards large-scale training of speech models such as Whisper from OpenAI (Radford et al., 2023). Despite its impressive recognition accuracy on many research datasets, the lack of transparency about the training data has lead the scientific community to build an *open-source equivalent* of Whisper. One such effort, dubbed as OWSM[1] (Peng et al., 2023) is trained on publicly available speech datasets, using an open source toolkit ESPnet (Watanabe et al., 2018). Training such models requires enormous computational resources that are not available for many academic and research organisations.

In this work, we primarily focus on studying architecture-specific regularisation for improving the robustness of ASR systems. More specifically, we introduce auxiliary classifiers in the decoder module of an encoder-decoder based neural architecture for ASR. These auxiliary classifiers not only help regularise the model during training, but also assist during joint-decoding, and acts as light-weight, rapid domain-adaptation modules. This study is done in conjunction with large-scale training[2] of ASR models on multi-domain English datasets.

1.1 Related works

The idea of auxiliary classifiers or intermediate regularisers has been explored in ASR. More specifically, Lee and Watanabe (2021) uses intermediate CTC objectives in the encoder module for ASR, whereas, Wang et al. (2021b) employs similar scheme for training self-supervised speech encoders. Zhang et al. (2022) regularises both the encoder and decoder modules by passing the intermediate representations from the encoder directly to the intermediate layers in the decoder. While these works have shown improvements over their respective baselines, our proposed approach differs in two aspects:

- We introduce auxiliary classifier(s) only in the decoder module of the encoder-decoder architecture, essentially regularising the internal language model.

[1] Open Whisper-style Speech Model.

[2] To the extent supported by the computational budget available for us.

- We further use these auxiliary classifiers in the joint-decoding scheme.

In case of large-scale training of end-to-end ASR models, we mainly take inspiration from prior works such as SpeechStew (Chan et al., 2021) and OWSM (Peng et al., 2023), where we simply mix mulitple publicly available datasets to train our models. It is important to note that simple aggregation from multiple sources (datasets) without text normalising can cause the models to *memorise* dataset-specific annotation styles (Peng et al., 2023); which is not desired for a general purpose ASR system. This also indicates a potential inefficiency, wherein model parameters are allocated towards recognising data sources rather than solving the intended task(s). As it is inevitable, we investigate and quantify the effect of text normalisation on the model's recognition performance.

1.2 Summary and contributions

- The decoder-centric regularisation is formally introduced in Section 2, where we also describe the proposed decoding strategies that exploit the auxiliary classifiers for joint-decoding.

- Experiment protocol is described in Section 3. Pilot experiments, studying DeCRED on single in-domain datasets is presented in Section 4.

- Experiments on large-scale multi-domain training and the effect of text-normalisation are described in Section 5. We show that the proposed DeCRED performs competetively to Whisper medium and outperform OWSM v3 on multiple datasets.

- Experiments on out-of-domain generalisation are presented in Section 6, where we additionally present a lightweight rapid adaptation capability of the auxiliary classifiers.

- Finally, our implementations[3] are built on top of open-source `transformers` library (Wolf et al., 2020), facilitating easy replication of our results. We intent to release of all model checkpoints along with corresponding test hypotheses. Our code also allows for single-line inference within the HuggingFace ecosystem.

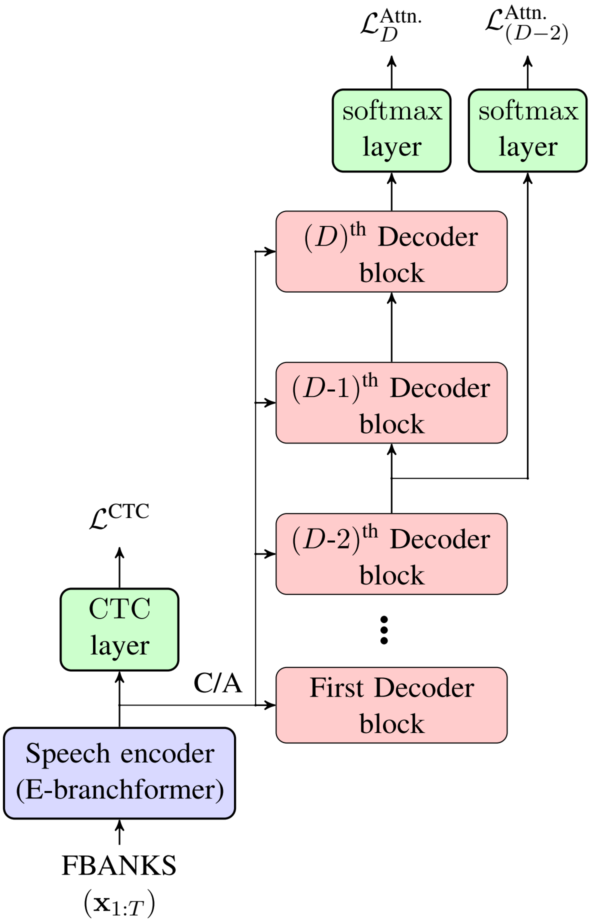
$\mathcal{L}_D^{\text{Attn.}}$
$\mathcal{L}_{(D-2)}^{\text{Attn.}}$
softmax layer
softmax layer
$(D)^{\text{th}}$ Decoder block
$(D\text{-}1)^{\text{th}}$ Decoder block
$(D\text{-}2)^{\text{th}}$ Decoder block
$\mathcal{L}^{\text{CTC}}$
CTC layer
C/A
First Decoder block
Speech encoder (E-branchformer)
FBANKS
$(\mathbf{x}_{1:T})$

Figure 1: Architecture of the proposed DeCRED. In addition to the standard encoder-decoder framework for ASR ($\mathcal{L}_D^{\text{Attn}}$), with the auxiliary CTC objective ($\mathcal{L}^{\text{CTC}}$), DeCRED uses – possibly multiple – auxiliary classifiers ($\mathcal{L}_d^{\text{Attn}}$) attached to the decoder. In the illustration, we shown one auxiliary classifier attached to $(D\text{-}2)$-th decoder block. The embedding and positional encoding layers are not dpeicted for brevity.

2 Decoder-centric regularization

Formally, our approach extends the training objective of encoder-decoder ASR by adding auxiliary cross-entropy loss functions. We explore two additional decoding methods that exploit these auxiliary classifiers.

2.1 Training objective

We build upon the hybrid CTC-attention-based training scheme proposed by Hori et al. (2017). Our objective function $\mathcal{L}$ is defined as:

[3] [to ensure author anonymity, the link to the resource will be added after the review process]

$$\mathcal{L} = \lambda \mathcal{L}^{\text{CTC}} + (1-\lambda)\mathcal{L}^{\text{DeCRED}}, \tag{1}$$

where $\mathcal{L}_{\text{CTC}}$ represents the standard CTC loss (Graves et al., 2006), λ is a hyper-parameter, and $\mathcal{L}_{\text{DeCRED}}$ is defined as:

$$\mathcal{L}^{\text{DeCRED}} = \sum_{d=1}^{D} \beta_d \mathcal{L}_d^{\text{Attn}}, \tag{2}$$

where D represents the number of layers in the decoder, $\mathcal{L}_d^{\text{Attn}}$ is the cross-entropy loss given a classifier layer (linear projection, followed by softmax function) attached to the d-th layer of the decoder, and β_d is the weighting factor of d-th layer. We impose constraints such that $\sum_{d=1}^{D} \beta_d = 1$ and $\beta_d \geq 0$. In practise $[\beta_1 \dots \beta_D]$ is a sparse vector. This definition allows us to explicitly regularise the decoder (internal language model) and force earlier layers to learn discriminative features suitable for the task. Figure 1 illustrates the proposed architecture, where an auxiliary classified is attached to output of $(D$-$2)$-th decoder block.

2.2 Decoding

The decoding follows a typical auto-regressive scheme observed in encoder-decoder ASR systems, where the posterior probability of an output token is obtained by conditioning on previously decoded tokens (partial hypothesis) and the input features.

Formally, let $\mathbf{x}_{1:T}$ be a sequence of input speech (filterbank) features, $\mathbf{h}_d \in \mathbb{R}^{1 \times d_{\text{model}}}$ denote the hidden representation obtained from the d-th layer of the decoder for a given time step, and $\mathbf{W}_d \in \mathbb{R}^{d_{\text{model}} \times V}$ represent linear projection from hidden dimension d_{model} to vocabulary size V. Then, the posterior probability of an output token at a given time step is

$$\begin{aligned} & p(y_n \mid y_{1:n-1}, \mathbf{x}_{1:T}) = \\ & \lambda \, p_{\text{CTC}}(y_{1:n-1} \mid \mathbf{x}_{1:T}) \times \\ & (1-\lambda) \, p_{\text{DeCRED}}(y_n \mid y_{1:n-1}, \mathbf{W}_{1:D}, \mathbf{h}_{1:D}), \end{aligned} \tag{3}$$

where λ is a hyper-parameter. We obtain the following variants by varying the definition of p_{DeCRED} and using auxiliary classifiers:

1. Vanilla joint CTC/attention decoding relying on representations *only* from the last layer $\mathbf{h}_D$:

$$p_{\text{DeCRED}}(\cdot) = \text{softmax}(\mathbf{h}_D \mathbf{W}_D) \tag{4}$$

2. Sum of logits weighted by per-layer learnable scalar β_d:

$$p_{\text{DeCRED}}(\cdot) = \text{softmax}(\sum_{d=1}^{D} \beta_d \mathbf{h}_d \mathbf{W}_d) \tag{5}$$

3. Sum of logits weighted by per-layer learnable vector $\mathbf{v}_d \in \mathbb{R}^{1 \times V}$:

$$p_{\text{DeCRED}}(\cdot) = \text{softmax}\left(\sum_{d=1}^{D} \mathbf{v}_d \odot (\mathbf{h}_d \mathbf{W}_d)\right) \tag{6}$$

Note that to obtain optimal results with methods (5) and (6), an additional held-out set is required for learning the parameters β_d, $\mathbf{v}_d$.

The above schemes can be easily integrated into any of the decoding search algorithms such as greedy and beam-search.

3 Experimental setup

The experiments are organized into three parts. The first one (Sec. 4) focuses on single in-domain datasets, studying the effect of the position (d), weight (β_d) of the auxiliary classifiers, and also their influence in decoding. The second part (Sec. 5) presents the experiments and results on scaling DeCRED to multi-domain English datasets. The selection of datasets is inspired by the ones used for evaluation in OWSM. This relatively larger corpus allows us to fully exploit the proposed decoding alternatives (5) and (6). The third part (Sec. 6) presents the out-of-domain generalisability of DeCRED by evaluating the trained models on AMI, Gigaspeech and FLEURS corpora. We also present a rapid, light-weight domain adaptation technique on the out-of-domain datasets.

All our experiments are built on top of the open-source `transformers` library, accompanied by baseline models built using the ESPnet toolkit.

3.1 Baseline Encoder-Decoder (ED) model

Our baseline ED model employs a feature extraction module consisting of two Conv2d layers with 256 output channels, and a linear projection. This is followed by a 12-layer E-Branchformer encoder with relative positional embeddings (Dai et al., 2019), Macaron-like forward modules, $d_{\text{model}} = 256$, $d_{\text{ff}} = 4 d_{\text{model}}$, four attention heads, and a dropout probability of 0.1. Adhering to the E-Branchformer architecture, we integrate a

Table 1: Effect of the position (d) and weight (β_d) of the auxiliary classifier in DeCRED on WERs of TEDLIUM3 test set. Grey cells indicate configurations deemed reasonable for exploration. Standard deviations (σ) and best WER for the chosen configurations are displayed. For reference, the baseline ED model has a WER of 7.2 %.

Weight	Position				
	1	2	3	4	5
0.1				7.5	6.8
0.2				7.0	7.2
0.3			7.0	7.0	7.0
0.4	7.8	7.5	7.1	**6.8** [$\sigma = 0.15$]	6.9
0.5	7.2	7.1	**6.7** [$\sigma = 0.26$]	7.1	6.9

merge block followed by depth-wise convolution with a kernel size of 31. Subsequently, the encoder is followed by a 6-layer decoder with sinusoidal positional embeddings, maintaining the same number of attention heads, d_{model}, and dropout. The only difference is the fixed $d_{\text{ff}} = 2048$ in the decoder.

Throughout this paper, we will use the triplet (enc. layers, dec. layers, d_{model}) to define the model architecture; the rest of the configuration is fixed unless explicitly stated otherwise. For example, small model $(12, 6, 256)$ has 35.04M parameters.

The model receives 80-dimensional filter-bank features as input. We used a sub-word tokeniser based on the unigram algorithm to build a vocabulary of size $V = 500$. This is our vanilla baseline ED model.

3.2 Training details

If not stated otherwise, models are trained on 4 Nvidia A100 GPUs with bf16 precision using the AdamW optimiser (Loshchilov and Hutter, 2019) with a learning rate of 2×10^{-3}, weight decay of 1×10^{-6}, linear decay scheduler and 15k warm-up steps. As an additional means of regularisation, we use a label smoothing weight of 0.1.

Unlike ESPnet, where some augmentations are applied offline, we implement all augmentations online and allow for postponing some of them until later in the training, resulting in a more stable training process. For instance, while ESPnet adopts a training regime consisting of 50 epochs and three copies of the input data with speed perturbation factors 0.9, 1.0, and 1.1, we train our model for 150 epochs on the original data with speed perturbation factors $\{0.9, 1.0, 1.1\}$ randomly sampled on the fly. After 5k update steps, we apply SpecAug (Park et al., 2019) with two frequency-masks of maximum size 27 and five time-masks with maximum coverage of masked input of 5 %. For all experiments, we select the best-performing model based on the development WER. To speed up the training, samples longer than 20 seconds are discarded from the training set.

Table 2: The effect of incorporating an auxiliary classifier with varied model sizes. The additional classifier is applied with $\beta_{D-2} = 0.4$ to maintain a consistent relative position wrt. the decoder output. Evaluated on the TEDLIUM3 test set.

Configuration	Size [M]	WER [%]	
		ED(4)	DeCRED(4)
(12, 6, 256)	35	7.2	6.8
(16, 8, 256)	56	**7.0**	**6.8**
(12, 6, 384)	73	7.2	7.0

4 Experiments on in-domain dataset

To analyse and understand the proposed regularisation scheme, we select a relatively small dataset, TEDLIUM3 (Hernandez et al., 2018), which allows for faster experiment turnout. The dataset comprises 452 hours of transcribed TED talks, with a test set containing 1155 utterances, roughly translating to 28k words. The size of this dataset enables us to train an ED (12, 6, 256) baseline 35M model to full convergence in approximately 70 A100 hours. An absolute improvement of 0.3 % in WER corresponds to around 100 additional correctly predicted words.

Since we build on top of `transformers` library, to ensure a fair comparison, we adopt hyperparameters and a training setup as close as possible to the ESPnet baseline recipe[4]. For evaluating the models on TEDLIUM3, unless explicitly specified, we follow the ESPnet recipe utilising joint CTC/attention decoding (4) with a beam size of 40 and CTC decoding weight $\lambda = 0.3$.

4.1 Position and weight of auxiliary classifiers

The DeCRED has an identical configuration as the baseline ED, except for the auxiliary classifier attached to specific layers in the decoder. Even with a model with as few as $D = 6$ decoder layers, the definition of the DeCRED objective (2) leaves us with a vast configuration space. We explored this space from configurations with a

[4] https://github.com/espnet/espnet/tree/master/egs2/tedlium3/asr1

Table 3: Comparison of our implementation of ED and proposed DeCRED with ESPnet's baseline on the TEDLIUM3 test split.

		WER [%]	
Model	Size [M]	greedy	beam – width 40
ESPnet ED$^{(4)}$	35.01	8.7	8.1
Our ED$^{(4)}$	35.04	7.6	7.2
DeCRED$^{(4)}$	35.20	**7.0**	**6.8**

Table 4: The performance of different variants of using auxiliary classifiers during decoding. All models have configuration (12, 6, 256). Evaluated on the CommonVoice 13.0 test set.

Model$^{\text{(decoding strategy)}}$	WER [%]
ED$^{(4)}$	19.693
DeCRED$^{(4)}$	19.467
DeCRED$^{(5)}$	19.464
DeCRED$^{(6)}$	19.444

single auxiliary classifier, changing its position and adjusting its weight in increments of 0.1. The additional parameters introduced ($\mathbf{W}_d \in \mathbb{R}^{256\times500}$) by a single auxiliary classifier do not significantly increase the model size.

The results are summarised in Table 1. Compared to our baseline ED model with a WER of 7.2 %, we observe improvements with the additional classifier placed closer to the final layer.

Further experiments with multiple auxiliary classifiers ($\{\beta_3 = 0.2, \beta_4 = 0.3, \beta_6 = 0.5\}$ and $\{\beta_3 = 0.2, \beta_5 = 0.3, \beta_6 = 0.5\}$), did not yield significant improvements, discouraging experiments with more auxiliary classifiers. We avoided exploring very low weights (β_d) in the early layers as gradual adjustments did not yield noticeable improvements. Given the computational resources required for each experiment run, we chose to run the two most promising configurations five times to determine the optimal one. Choosing between the two most promising configurations from Table 1, i. e., $\beta_3 = 0.5$ and $\beta_4 = 0.4$, we opted for the latter for all subsequent experiments. We believe other configurations (indicated with grey colour in the lower triangle of Table 1) could yield similar results.

4.2 Scaling model size

Table 2 compares the effect of regularisation with respect to the width (d_{model}) or depth (D). Although wider baseline ED models have shown minor degradation, likely due to incomplete convergence, intermediate regularisation consistently improved performance.

Finally, Table 3 compares our best-performing DeCRED and ED baseline models with the baseline model from ESPnet. DeCRED consistently outperforms both implementations of ED. The difference is better pronounced in greedy decoding, suggesting the effectiveness of DeCRED in decoding tasks where computational resources are limited.

4.3 Additional classifiers in decoding

Having observed that DeCRED leads to a better-trained model, we move on to explore the effect of the additional classifier heads in decoding. We conduct this experiment on CommonVoice en 13.0 dataset (Ardila et al., 2020). It consists of 2.4k validated hours from 1.1k unique voices of volunteer contributors.

To learn the mixing parameters β^*_d and $\mathbf{v}^*$ of the respective decoding methods (Section 2.2), we split the original 10888 development utterances into new training and development sets in ratio 70:30. Expect for the mixing parameters, the rest of the model is frozen. This training or fine-tuning is very light-weight and took about 30 minutes on a single A100 GPU with a batch size of 512 samples for 10 epochs.

We use the equation number in the superscript of the model to denote the decoding objective, i.e. DeCRED$^{(4)}$ indicates the vanilla decoding method defined by (4), DeCRED$^{(5)}$ indicates mixing the logits by learnable scalars, and DeCRED$^{(6)}$ indicates mixing the logits by learnable vectors.

The comparison across the three variants, along with the baseline ED is shown in Table 4. Overall, using the additional classifiers during decoding does no harm. A 0.25 % WER improvement translates only to 350 additional correctly predicted words. This experiment does not provide any strong evidence in favour of using them. We hypothesize that it could be due to a relatively small model size (36M params with 1000 sub-word vocabulary).

5 Scaling to multi-domain scenario

To fully investigate and leverage the power of the auxiliary classifiers, we chose a mixture of multi-domain datasets that allows for bigger training, development and test sets. The multi-domain dataset is comprised of Fisher

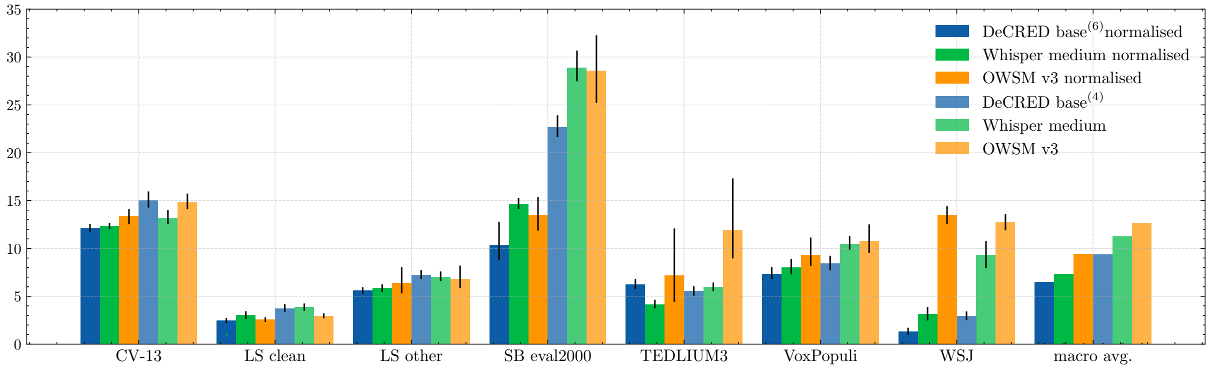

Figure 2: Comparison of the proposed model against publicly available models on original and normalised transcripts using greedy decoding (4) with $\lambda = 0$. DeCRED-base$^{(6)}$ indicates the model with the proposed decoding technique (6), and the mixing parameters $\mathbf{v}$ tuned on development split. To compute confidence intervals, we employed bootstrapping with $\alpha = 0.05$ and $B = 1000$.

(SWITCHBOARD) (Godfrey et al., 1992), WSJ (Paul and Baker, 1992), Common Voice en 13 (Ardila et al., 2020), LibriSpeech (Panayotov et al., 2015), VoxPopuli (Wang et al., 2021a), and TED-LIUM 3 (Hernandez et al., 2018), totalling 6k hours of training data.

5.1 Normalisation of multi-domain data

These datasets have different annotation styles, making learning harder and introducing undesired behaviour in the models, such as memorising the dataset-specific annotations (Peng et al., 2023). We employed a practical approach, where we used the text normalisation scheme from Whisper[5] to standardise the transcripts across all the datasets. We believe this approach allows the model to focus mainly on the recognition task. For practical applications, true casing and punctuation can be restored using a lightweight inverse text normalisation model. In addition to the Whisper text normaliser, we retained the text within parenthesis. Due to inconsistencies across datasets, we removed special tokens such as [breath], [vocalised noise], [pause], [sneeze].

Nevertheless, to enable a fair comparison with prior works, we also report results using the original transcripts. This allows us to quantify the effect of text normalisation on the WER.

5.2 Training setup

We expanded the vocabulary size to $V = 5000$ tokens and trained baseline ED and DeCRED models for 100 epochs with early stopping patience of 10 epochs and 40k warm-up steps with two configurations: *small* (12, 6, 256) with 38.5M parameters and *base* (16, 8, 512) with 172M parameters. Small models were trained on 32 A100 GPUs[6] maintaining an overall batch size of 2048 samples and base on 48 GPUs[7] maintaining batch size 3072. The rest of the settings are identical to the previous setup described in Section 3.2.

Table 5: Macro average of the WERs based on the selected decoding strategy. By default, all auxiliary classifier weights β_d are set to 0. Parameters with an asterisk (e.g., β_d^*, $\mathbf{v}^*$) indicate tuning on a portion of the development split.

DeCRED-base	greedy		beam – width 10	
decoding strategy	$\lambda = 0$	$\lambda = 0.3$	$\lambda = 0$	$\lambda = 0.3$
$\beta_6 = 1^{(5)}$	6.8	6.7	6.5	6.4
$\beta_8 = 1^{(4)}$	6.7	6.4	**6.4**	**6.0**
$\beta_{6,8} = (0.40, 0.60)^{(5)}$	6.7	6.6	**6.4**	6.3
$\beta^*_{6,8} = (0.50, 0.47)^{(5)}$	6.7	6.6	6.5	6.3
$\mathbf{v}^{*(6)}$	**6.5**	**6.3**	6.9	**6.0**

Additionally, we introduced a mechanism to mask special tokens, along with unfinished words[8], during error backpropagation. This strategy aimed to prevent the model from being penalised for unclear inputs.

5.3 Comparison with Whisper and OWSM

Figure 2 presents a comparative analysis of our best-to-date model, DeCRED-base$^{(4)}$ (172M parameters), and DeCRED-base$^{(6)}$, incorporating additional 10k parameters in $\mathbf{v}^*$, against the

[5] https://github.com/huggingface/transformers/blob/main/src/transformers/models/whisper/english_normalizer.py

[6] consuming approximately 840 A100 hours per run

[7] consuming approximately 2240 A100 hours per run

[8] e.g. transcript “[hesitation] to re- to re- renew" is transformed into “[MASK] to [MASK] to [MASK] renew"

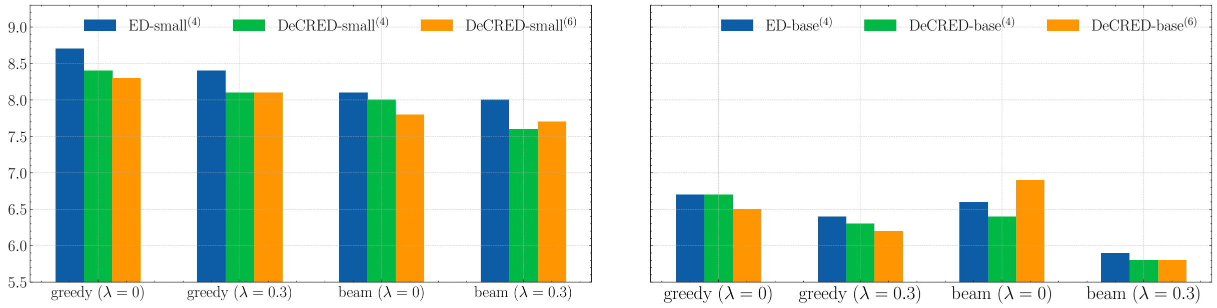

Figure 3: The impact of employment of the proposed training strategy, along with the enhanced decoding (6) on small (12, 6, 256) and base (16, 8, 512) models using greedy and beam decoding.

Whisper-medium (Radford et al., 2023) (700M) and OWSM v3 (Peng et al., 2023) (889M) models. Although most of our models were trained on the normalised transcriptions, to highlight the effect of text normalisation, we also trained the DeCRED base (16, 8, 512) model on original transcriptions. When evaluating the Whisper and OWSM models in a normalized setup, we post-processed their hypotheses using the same text normalization as used in our training pipeline. This ensures the fairest comparison possible.

5.4 Performance vs. speed

Table 5 presents a comparison of different decoding methods in terms of macro WER over the aforementioned datasets. We observed that integrating intermediate representations with per-layer learnable weights (5) led to minor improvement only in the greedy decoding scenario without hybrid CTC decoding. Notable enhancements were observed with the incorporation of per-token-specific mixing (6), except for beam decoding on the SB eval2000 dataset, where we observed a degradation of 5.3 % in WER, primarily attributed to insertions. Interestingly, we did not observe the same behaviour with the small model. For completeness, macro WERs are also provided for early-exiting ($\beta_6 = 1$, i.e., decoding directly from 6-th layer, while the model has 8 layers), where only minor degradations were observed.

Figure 3 showcases the improvements in macro WER achieved by DeCRED-small (12, 6, 256) and DeCRED-base (16, 8, 512) compared to the respective ED variants with different decoding methods.

Figure 4 presents the relative WER reductions of our multi-domain models on TEDLIUM3 in relation to the relative slowdown caused by additional decoding overhead. The slowdown factor is measured relatively to our fastest model ED small$^{(4)}$. It is calculated as an average time over the TEDLIUM3 test set required to emit 20 tokens on an A100 GPU with maximum VRAM memory consumption[9]. We fixed a number of decoding steps to normalise different hypothesis lengths across models.

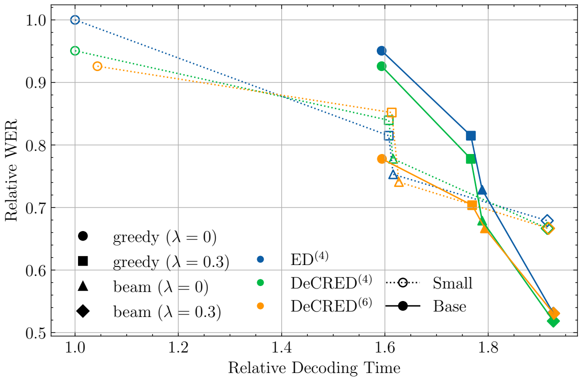

Figure 4: The impact of model size and decoding approach on the average time needed to transcribe an utterance (TEDLIUM3) and WER (macro average across datasets).

In this setup, there is no speed difference between ED$^{(4)}$ and DeCRED$^{(4)}$. However, as shown in Figure 4, regularised models significantly reduce the WER. When using DeCRED$^{(6)}$, the only overhead is computing softmax $\left(\sum_{d=1}^{D} \mathbf{v}_d \odot (\mathbf{h}_d \mathbf{W}_d)\right)$, where $\mathbf{h}_D \mathbf{W}_D$ is already computed. It is worth noting that when using greedy decoding, DeCRED small performs similarly to ED base, being much smaller, thus consuming less computation resources and speeding up decoding significantly.

[9] For example, with greedy decoding and ED small, we can fit 240 samples in a batch. In contrast, with ED base and joint CTC/attention decoding with a beam size of 10, we are only able to fit 20 samples.

Table 6: Unseen datasets used to evaluate the generalization ability of our models.

Dataset	Domain	# test utterances
FLEURS	Read speech	0.6k
AMI	Conversational	12.6k
Gigapeech	Mixture	25.3k

5.5 Effect of text normalisation

To further understand the effect of text normalisation, we trained standalone models ED-small (12, 6, 256) on the TEDLIUM3 and Voxpopuli datasets with and without normalised transcripts. Notably, we observed an improvement from 9.8 % to 9.0 % WER on VoxPopuli and from 7.2 % to 6.7 % on TEDLIUM3. The normalisation process effectively resolved contraction errors and also led to fewer errors in the most frequent confusion pairs (e.g., "the" vs "a", "in" vs "on", "in" vs "and"). By normalising, we reduced the number of words from 44.3k to 44.1k for Voxpopuli and increased this number from 27.5k to 28.2k for TEDLIUM3, which also influenced the WER.

6 Out of domain performance

To study the generalisation capabilities of our models, we evaluate our best-to-date ED and DeCRED models on three unseen datasets (Table 6).

The performances are summarized in Table 7. Overall, all our models outperform the much larger OWSMv3, which has also been trained on the corresponding training data, showing that our models do generalize to unseen domains well. With the exception of the FLEURS dataset, where the difference in WER is the smallest anyway, DeCRED models outperform the ED baseline significantly, suggesting that the decoder-centric regularisation enhances the generalisation ability of the model.

Additionally, we take this as an opportunity to evaluate the effect of tuning the mixing weights $\mathbf{v}^*$ on the corresponding domain. For this, we utilised FLEURS train split, and development splits of AMI and Gigaspeech, respectively, following the training protocol described in Section 4.3. In Table 7, these models are denoted as DeCRED base$^{(6)\dagger}$. In all cases, adapting $\mathbf{v}^*$ leads to decrease in WER, and with the exception of FLEURS dataset, this decrease is considerable, confirming that the mixing weight does do provide a rapid adaptation capability.

Table 7: Comparison of ED and DeCRED models on out-of-domain test sets. WERs are obtained using greedy decoding with $\lambda = 0$. † denotes models where $\mathbf{v}^*$ was tuned on each of the datasets separately.

Model	FLEURS	AMI-ihm	Gigaspeech
ED base$^{(4)}$	6.4	24.8	19.8
DeCRED base$^{(4)}$	6.7	22.1	16.9
DeCRED base$^{(6)}$	6.9	21.9	17.0
DeCRED base$^{(6)\dagger}$	6.8	21.4	16.4
OWSM v3	8.6	35.8	34.1
Whisper medium	**5.5**	**16.6**	**14.9**

7 Conclusion

We introduced the DeCRED regularization scheme, which effectively integrates auxiliary classifiers within the decoder of an encoder-decoder-based architecture. We further proposed decoding methods that exploit these auxiliary classifiers, which led to a significant decrease in the word error rates. We observed that DeCRED consistently improves the results when employing simple greedy decoding scheme as compared to the baseline models. Our experiments on multi-domain datasets show that DeCRED is scalable and performs competetively to much larger Whisper medium and outperform OWSM v3. Finally, we show that DeCRED enhances the generalisation to out-of-domain datasets, where we observed a reduction of 2.7 and 2.9 WERs on AMI and Gigaspeech respectively. Using a light-weight rapid domain adaptation scheme enabled by DeCRED, the out-of-domain WERs were further reduced by 0.7 and 0.5 % absolute on the respective datasets. In future, we intend to study DeCRED in multilingual and multi-task scenario.

8 Limitations

We identify a few of limitations in our work. Firstly, due to our computational budget, we were only able to scale our setup to 6k hours of training data and 172M model parameters. Secondly, our models were trained on English data only, which makes the comparison with multilingual models tricky, as these models had to invest a part of their capacity into modeling other languages as well. Yet, due to the first point, our models are exposed to one (OSWM) or even two (Whisper) orders of magnitude less English data, therefore we believe the comparison is not unfair. Also, our models use considerably smaller vocabulary; however, while this might limit model performance on domain-specific words present for example in the FLEURS dataset, we do not observe performance degradation there. Next, some of the improvements from introducing DeCRED diminish when employing beam-search decoding with a wider beam, which however comes at a computational cost at inference time. Finally, while the proposed decoder-centric regularization is independent of the backbone architecture, we have only analysed our approach using an E-branchformer speech encoder.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. 2021. SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 369–376, New York, NY, USA. Association for Computing Machinery.

Francois Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings. pages 198–208.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–529, Vancouver, Canada. Association for Computational Linguistics.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate Loss Regularization for CTC-Based Speech Recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Arun Narayanan, Ananya Misra, Khe Chai Sim, Golan Pundak, Anshuman Tripathi, Mohamed Elfeky, Parisa Haghani, Trevor Strohman, and Michiel Bacchiani. 2018. Toward Domain-Invariant Speech Recognition via Large Scale Training. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 441–447.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. ISSN: 2379-190X.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, pages 2613–2617. ISCA.

Douglas B. Paul and Janet M. Baker. 1992. The Design for the Wall Street Journal-based CSR Corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.*

Yifan Peng, Jinchuan Tian, Brian Yan, Dan Berrebbi, Xuankai Chang, Xinjian Li, Jiatong Shi, Siddhant Arora, William Chen, Roshan Sharma, Wangyou Zhang, Yui Sudo, Muhammad Shakeel, Jee-weon Jung, Soumi Maiti, and Shinji Watanabe. 2023. Reproducing Whisper-Style Training Using an Open-Source Toolkit and Publicly Available Data. *arXiv preprint*. ArXiv:2309.13876 [cs, eess].

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518. PMLR. ISSN: 2640-3498.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary

Williamson, Juan Pino, and Emmanuel Dupoux. 2021a. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Chengyi Wang, Yu Wu, Sanyuan Chen, Shujie Liu, Jinyu Li, Yao Qian, and Zhenglu Yang. 2021b. Self-supervised learning for speech recognition with intermediate layer supervision. *Preprint*, arXiv:2112.08778.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-End Speech Processing Toolkit. pages 2207–2211.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jicheng Zhang, Yizhou Peng, Haihua Xu, Yi He, Eng Siong Chng, and Hao Huang. 2022. Intermediate-layer output regularization for attention-based speech recognition with shared decoder. *Preprint*, arXiv:2207.04177.