Progressive Down-Sampling for Acoustic Encoding

Anonymous ACL submission

Abstract

In acoustic encoding, the fine-grained frame-level features are not suited for capturing global dependencies. But condensing them into a semantically complete representation by stacked down-sampling does not work well. We find that the condensation leads to the degraded correlation of the representations in adjacent positions, which poses the risk of information loss in the stacked method. In this work, we propose a new method, progressive down-sampling (PDS), for encoding the context sufficiently before each condensation. Also, we develop a representation fusion method to alleviate information loss by combining the multi-scale representations. Experimental results on the 960h LibriSpeech automatic speech recognition task show that, for a strong Conformer-based system, our method down-samples the input speech features to 1/32 of the initial length, while yielding an improvement of 0.47 WER with a speedup of 1.42×. It also achieves the state-of-the-art BLEU score (25.8) on the MuST-C En-De speech translation benchmark with no additional training data.

1 Introduction

Despite the success in speech processing tasks like automatic speech recognition (ASR) (Lu et al., 2020; Zhang et al., 2021) and speech translation (ST) (Xu et al., 2021), how to encode the speech features effectively is an open problem. Different from modeling based on discrete units in natural language processing, a standard paradigm for acoustic encoding is taking as input the continuous frame-level features with a very short shift.

Framing generates a very long sequence consisting of fine-grained features. For example, a framing-based feature sequence is in general tens of times longer than the sub-word sequence in a transcription (see Figure 1). For encoding, such a problem leads to the difficulties of capturing long-distance dependencies and distributing the attention weights across semantically incomplete modeling units (Han et al., 2019). Previous work (Al-Rfou et al., 2019) also demonstrates that the fine-grained character-level models yield significantly inferior performance compared with word-level counterparts. In addition, the long sequence also results in prohibitive computation costs due to the quadratic complexity of self-attention.

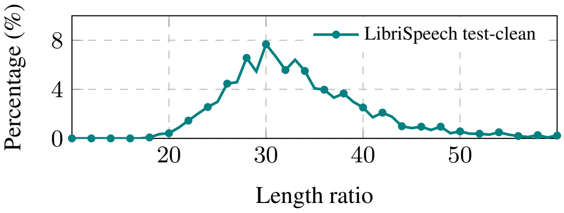


Figure 1: The distribution of the length ratio between the speech features (frame-level) and corresponding transcriptions (sub-word level).

A popular method is down-sampling (DS) fine-grained feature to form a more meaningful representation by stacking multiple strided convolution layers before encoding (Dong et al., 2018; Berard et al., 2018). Unfortunately, it does not work well when the down-sampling ratio increases. Intuitively, it is difficult to condense dozens of frames into one unit straightly (Sayood, 2018). It is something like that a few principal components can not preserve all the information in the classical principal component analysis method (Wold et al., 1987).

For shedding light on the reason of failure, we analyze the condensation process. We find that the correlation of representations in adjacent positions degrades due to down-sampling, which increases the difficulty for subsequent condensation. This leads to the non-trivial issue of information loss in the stacked method.

To address this issue, we propose a *Progressive Down-Sampling* (PDS) method. For the input speech features, a single layer of down-sampling is employed to aggregate the consecutive representations into more informative units. Then the model encodes the context for high correlation of the representation over the sequence. Repeating the above process, we aggregate the frame-level features into more semantically complete units in a progressive manner.

In this way, the multi-scale representations of different granularities are obtained. The fine-grained representations, on the other hand, may contain information that is lost during condensation. To further address the problem, we align the multi-scale representations to the same shape, and then combine them by a lightweight representation fusion method.

PDS is a general method for acoustic encoding. It is easy to make a trade-off between computational speedup and performance. We evaluate it on ASR and End-to-End ST tasks. Experiments on LibriSpeech ASR show that our method achieves a high down-sampling ratio up to 32. Also, it is beneficial to both system speedup and performance improvement. It outperforms the stacked counterparts by 0.47 WER with a speedup of 1.42×. On a more challenging task of ST, our method helps model convergence and achieves the state-of-the-art BLEU score of 25.8 on the MuST-C En-De benchmark without additional resources.

2 Related Work

Unlike text that has explicit boundaries, audio is in general represented in continuous signals. Although researchers have explored models based on the raw audio signal (Schneider et al., 2019), the popular method for segmentation is framing with a frame size of 25ms and a frame shift of 10ms (Oppenheim, 1999). The short frame shift allows the continuity of the speech signal, and the overlapping segments help to avoid the information loss between consecutive frames.

However, the fine-grained frame-level features may not be suitable for the state-of-the-art architectures (Vaswani et al., 2017). The long sequences composed of semantically incomplete units lead to the difficulties of capturing long-distance dependencies and distributing the attention weights across the most related positions. Researchers (Salesky et al., 2019; Salesky and Black, 2020) investigate phoneme-level methods. For example, one can average frame-level features within phoneme-like units. But this needs a non-trivial recognizer for phoneme alignment.

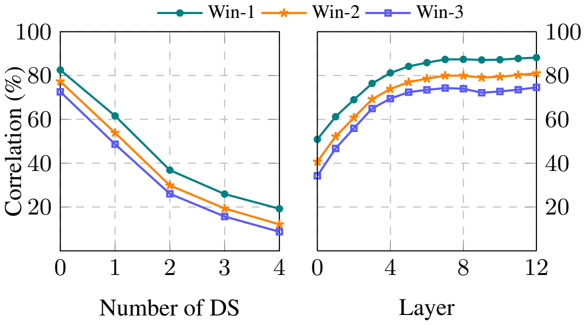


Figure 2: Left: the correlation after each down-sampling. Right: the correlation in each Layer. Win-d represents the window size of d. Note that 0 represents the input speech features and down-sampled features respectively.

Motivated by the work in efficient models, researchers alleviate the modeling difficulty by the improved self-attention mechanisms (Han et al., 2019; Alastruey et al., 2021; Papi et al., 2021). However, they ignore the inherent problem of fine-grained modeling and the cross-attention module still suffers from the same issue.

A natural idea is to down-sample the fine-grained features to generate a more meaningful representation (Chan et al., 2015; Bahdanau et al., 2016). To do this, a popular method is to pass the features through a stack of strided convolutional layers before encoding (Dong et al., 2018; Berard et al., 2018). But the stacked method does not work well in practice due to the loss of information in consecutive convolutional operations. As a way to address this, several research groups use the progressive method to down-sample the acoustic sequence (Peddinti et al., 2018; Huang et al., 2020; Han et al., 2020; Burchi and Vielzeuf, 2021). However, there is still no in-depth analysis on this problem.

Another open problem for acoustic encoding is the variable information caused by silence or noise. Researchers develop adaptive selection (Zhang et al., 2020a) or dynamic down-sampling methods (Na et al., 2019; Zhang et al., 2019) for avoiding useless features. However, the granularity of the filtered representation is still far from ideal. Here we explicitly discuss the problem and focus on effective down-sampling with a fixed ratio.

3 The Method

3.1 Why Is Information Lost?

Down-sampling generates more semantically complete units by aggregating the adjacent frame-

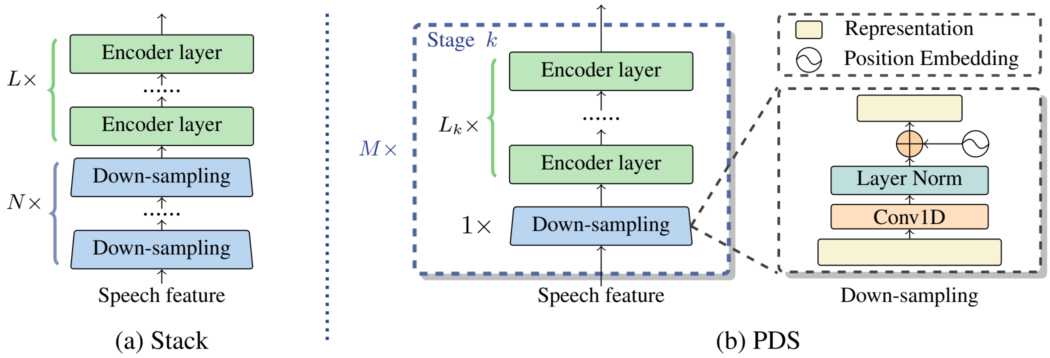


Figure 3: Comparison of the Stack and PDS methods.

level features. Following previous work in down-sampling (Dong et al., 2018; Berard et al., 2018), speech features are fed into a stack of 2 convolutions with a stride of 2. The convolution layers are followed by a number of encoder layers (see Figure 3 (a)). For a large down-sampling ratio, it is natural to stack more down-sampling layers.

We study the changes of representation during down-sampling. We define the *correlation* of representation as the average cosine similarity of each unit to the surrounding units within a small window. High correlation means that the representations of the adjacent positions are similar.

We train a Transformer-based (Vaswani et al., 2017) ASR model with 4 stacked down-sampling on the 960h LibriSpeech dataset and show the correlation of the test-clean test set. As shown in Figure 2 (Left), the input speech features have an extremely high correlation due to the overlapping framing. However, the correlation degrades sharply after each down-sampling. The subsequent down-sampling processes are difficult to condense the diverse representation while preserving the information completely. We call this issue information loss caused by stacked down-sampling.

Now a new question arises: how to increase the correlation of the representation and alleviate the information loss in down-sampling? An intuitive conjecture is that the context modeling increases the correlation due to the strong preference of the short-distance dependency (Sperber et al., 2018; Xu et al., 2021). Figure 2 (Right) shows the correlation in each layer of the encoder of the standard Transformer with a down-sampling ratio of 4. Obviously, the correlation increases from bottom to top, as we expected. This motivates us to develop a progressive method for encoding context information sufficiently after each down-sampling.

3.2 Progressive Down-Sampling

We propose a *Progressive Down-Sampling* (PDS) method to condense the fine-grained features into the semantically complete units. See Figure 3 (b) for an overview of PDS. The encoding is divided to two processes: a representation down-sampling process and a context interaction process.

For the input speech features like MFCC or Mel filter bank, an overlapping down-sampling condenses it by a simple convolution 1D module. Like framing, the overlapping in convolution alleviates the information loss. It also enforces the model to capture local modeling. To deal with varied sequence lengths, the position encoding is introduced into the representation after layer normalization.

Inspired by the finding presented in Section 3.1, the down-sampled representation requires sufficient context interaction for high correlation. Here we simply use the multiple identity layers to capture the dependencies.

Each run of down-sampling and encoding is called a *stage*. The model runs for M stages and obtains more meaningful representations $\{H_1, H_2, \cdots, H_M\}$.

A merit of PDS is that it offers a trade-off between computational efficiency and performance. One can stack more stages for extreme down-sampling. This decreases the computational cost significantly but may lead to the performance drop due to the inevitable information loss. On the other hand, fewer down-sampling processes preserve the information for better performance but cannot provide sufficient speedups. Note that the stacked method can be seen as a specific case of PDS: it consists of two stages and the number of layers in the first stage is zero.

PDS is also similar to the typical backbones in the field of computer vision (CV), like CNN (He

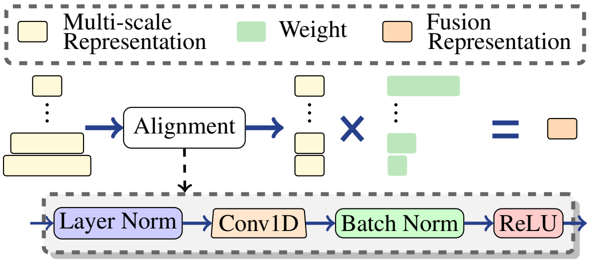


Figure 4: The representation fusion method. It aligns multi-scale representations to the same shapes and combines them.

et al., 2016) and Transformer (Wang et al., 2021). Both of them employ the same design concept, i.e., aggregating the fine-grained input into the more informative representation by progressive down-sampling. This paradigm is widely used in CV tasks. Here we explore it in the field of speech processing.

3.3 Representation Fusion

As the nature of down-sampling, the information loss still occurs inevitably although we use the progressive method. Motivated by previous methods to make full use of the multi-level representations (Wang et al., 2018, 2019), a way to further address the problem is to fuse the finer-grained representations (Zhao et al., 2017; Zhang et al., 2020b). Then the final output representation H^o can be defined as:

$$H^o = \mathcal{F}(H_1, \cdots, H_M) \tag{1}$$

where $\mathcal{F}(\cdot)$ is the fusion function[1]. But this raises a new question: how to combine the multi-scale representations effectively?

The first step is to align the different scales to the same one. We resort to a simple but effective non-overlapping convolution operation to transform the finer-grained representations outputted in bottom stages to the shape of H_M. The stride for the representation H_k is set to the multiplication of the subsequent down-sampling ratios[2].

Drawing on the design of the convolution module in Conformer, the representation fusion method with alignment is shown in Figure 4. We use a simple linear combination of the representations that are already in the same shape. The output $\mathcal{F}(\cdot)$ is defined as:

$$\mathcal{F}(H_1, \cdots, H_M) = \sum_{k=1}^{M} W_k \mathrm{LN}(\mathrm{A}(H_k)) \tag{2}$$

where $\mathrm{A}(\cdot)$ is the alignment function and $\mathrm{LN}(\cdot)$ is the layer normalization function. $W_k \in \mathbb{R}$ is a learnable scalar to weight the aligned representations. The weights are initialized to the same values, and are updated as other parameters during training.

Setting	Stride	Layer
Stack-4	2-2	0-12
PDS-Base-8	2-2-1-2	3-3-3-3
PDS-Base-16	2-2-2-2	2-2-6-2
PDS-Base-32	2-2-2-2-2	2-2-3-3-2
Stack-4	2-2	0-30
PDS-Deep-8	2-2-1-2	7-7-7-9
PDS-Deep-16	2-2-2-2	5-5-12-8
PDS-Deep-32	2-2-2-2-2	5-5-7-7-6

Table 1: Settings of PDS. "Stack-4" represents the standard method. "PDS-Base-R" and "PDS-Deep-R" denote an encoder of 12 layers and 30 layers with a down-sampling ratio of R respectively. "Stride" and "Layer" separated by "-" represent the stride of the down-sampling module and the number of layers in each stage from bottom to top.

3.4 PDS Settings

In this work, we choose 8 settings with different depths and down-sampling ratios (see Table 1). We basically follow the design in He et al. (2016) but do adaptations for acoustic encoding:

- In down-sampling, a bigger window size means that more context information is involved. But this also increases the difficulty for down-sampling due to the lower correlation. We use an empirical setting of kernel size $= 5$.
- We do not increase hidden dimensions as the growth of the model depth, i.e., we use the same hidden dimensions for all stages. Modeling with a small dimension in the bottom stage is obviously not a good idea due to the high dimension of the initial features (typically 80-dimension).
- The bottom stages have fewer layers for efficient computation due to the longer sequence. The major computations are concentrated on the intermediate stages. This leads to computation acceleration and sufficient encoding.

[1] We drop the input feature because it is extracted by signal processing rather than the encoding model.

[2] The stride for H_M is set to 1.

4 Experiments

We evaluate PDS on ASR and ST tasks.

4.1 Datasets and Preprocessing

The datasets are from two benchmarks:

- **LibriSpeech** is a publicly available read English ASR corpus, which consists of 960-hour of training data (Panayotov et al., 2015). The development and test data are divided into clean and other subsets according to the speech quality. We select the model on the dev-clean set and report results on all four subsets.

- **MuST-C En-De** is a multilingual speech translation corpus extracted from the TED talks (Gangi et al., 2019). We train our systems on the English-German speech translation dataset of 400-hour speech. We select (and tune) the model on the dev set and report results on the tst-COMMON set.

For preprocessing, we follow the common recipes in fairseq toolkit[3], we remove the utterances of more than 3,000 frames or fewer than 5 frames. Speed perturbation is not used in all models. The 80-channel Mel filter bank features are extracted by a 25ms window with a stride of 10ms. We learn SentencePiece[4] segmentation with a size of 10,000 for the two datasets. For ST, we use a shared vocabulary for source and target languages.

4.2 Model Settings

We use the encoder-decoder framework and implement the method based on the fairseq toolkit. We use the Adam optimizer and adopt the default learning schedule in fairseq. We apply dropout with a rate of 0.1 and label smoothing $\epsilon_{ls} = 0.1$ for regularization. SpecAugment (Park et al., 2019) is applied in the input speech features for better generalization and robustness.

For ASR tasks, we evaluate our method on Transformer (Vaswani et al., 2017) and Conformer (Gulati et al., 2020). The settings of the encoder for ASR models are shown in Table 2. The decoder consists of 6 Transformer layers and the settings are same to the encoder. CTC (Graves et al., 2006) multitask learning is not used due to the very modest improvement in our preliminary experiments.

For ST tasks, we evaluate our method on Transformer and SATE (Xu et al., 2021). Knowledge distillation is not used for simplicity. The encoder consists of 12 layers for Transformer. SATE has an acoustic encoder of 12 layers and a textual encoder of 6 layers. Each layer comprises 256 hidden units, 4 attention heads, and 2,048 feed-forward hidden units. We use pre-training for more challenging ST task, where the ASR and MT models are pre-trained with the MuST-C En-De data. Different from the ASR model, CTC is employed with a weight of 0.3 for better convergence.

All the models are trained for 100 epochs. We early stop training when there is no performance improvement on the development set for 10 consecutive checkpoints. We use beam search decoding with a beam size of 5 for all models. The CTC and language model rescoring methods are not used. WER and case-sensitive SacreBLEU are reported for ASR and ST respectively.

4.3 Results of ASR

Table 2 shows the results on the 960h LibriSpeech corpus. We compare the methods on Transformer and Conformer with different encoder layers and hidden dimensions. We use Stack-4 as the baseline model (see Table 1 for the setting). The amount of model parameters in each group is similar for a fair comparison.

For the popular setting of 12 encoder layers with 256 hidden dimensions in rows (A), PDS is the first to achieve a very high down-sampling ratio of 32 with no performance drop. It yields a speedup of 1.20×. As the down-sampling ratio decreases, performance improves significantly. Similar phenomena are observed on the wider Transformer with 512 hidden dimensions in rows (B), and this bigger model benefits more in speedup.

Interestingly, we find that the deep models with 30 encoder layers in rows (C) eliminate the performance gap when different down-sampling ratios are employed. PDS condenses the representation to 1/32 of the initial length, while achieving a considerable relative improvement of 0.75 WER points. We conjecture that the deep model allows more sufficient modeling in each stage and preserves the information even in an extreme case of down-sampling. This has a practical advantage in industrial scenarios wherein more speedups are required.

For Conformer, the behavior of the base model with 12 encoder layers in rows (D) is similar to

[3] https://github.com/pytorch/fairseq

[4] https://github.com/google/sentencepiece

	Setting	L	d_h	d_{ff}	h	#Params	dev		test		Avg.	Speedup
							clean	other	clean	other		
Transformer												
(A)	Stack-4	12	256	2048	4	30M	3.88	9.26	4.49	9.42	6.76	1.00×
	PDS-Base-8					30M	3.57	8.63	3.85	8.58	**6.16 (-0.60)**	0.99×
	PDS-Base-16					30M	3.71	8.73	3.74	9.02	6.30 (-0.46)	1.14×
	PDS-Base-32					31M	4.13	9.31	4.21	9.31	6.74 (-0.02)	1.20×
(B)	Stack-4	12	512	2048	8	71M	3.53	8.15	3.67	7.96	5.83	1.00×
	PDS-Base-8					75M	3.17	7.46	3.47	7.47	5.39 **(-0.44)**	1.08×
	PDS-Base-16					76M	3.34	7.73	3.37	7.85	5.57 (-0.26)	1.34×
	PDS-Base-32					82M	3.32	7.94	3.64	7.85	5.69 (-0.14)	1.47×
(C)	Stack-4	30	256	2048	4	53M	3.80	8.51	4.33	8.61	6.31	1.00×
	PDS-Base-8					53M	3.34	7.90	3.50	7.79	5.63 (-0.68)	1.03×
	PDS-Base-16					54M	3.15	7.83	3.38	7.79	**5.53 (-0.78)**	1.19×
	PDS-Base-32					55M	3.26	7.77	3.33	7.88	5.56 (-0.75)	1.27×
Conformer												
(D)	Stack-4	12	256	2048	4	45M	3.64	8.17	3.74	8.10	5.91	1.00×
	PDS-Base-8					46M	3.10	7.41	3.23	7.59	5.33 (-0.58)	0.98×
	PDS-Base-16					46M	3.06	7.27	3.12	7.58	5.26 **(-0.65)**	1.16×
	PDS-Base-32					47M	2.99	7.57	3.17	7.76	5.37 (-0.54)	1.20×
(E)	Stack-4	12	512	2048	8	109M	3.67	7.66	3.79	7.50	5.56	1.00×
	PDS-Base-8					113M	2.90	6.71	3.10	6.70	4.85 **(-0.71)**	1.00×
	PDS-Base-16					114M	3.11	6.74	3.37	6.90	5.03 (-0.53)	1.33×
	PDS-Base-32					119M	2.96	7.10	3.18	7.11	5.09 (-0.47)	1.42×

Table 2: WER on 960h LibriSpeech ASR corpus. L: the number of encoder layers. d_h: the hidden dimension. d_{ff}: the feed-forward dimension. h: the number of the attention heads. #Params: the number of parameters. The speedup is computed during inference on test-clean set with the batch size of 100k and beam size of 5.

that of the deep Transformer models. One important improvement of Conformer is enhancing local dependency by convolution neural networks. It models interaction among neighbor context, which leads to the higher correlation. It is helpful for alleviating the issue of information loss.

Finally, on the wider Conformer in rows (E), our method yields an improvement of 0.47 WER points with a speedup of 1.42×.

4.4 Results of ST

End-to-end ST has become popular recently (Duong et al., 2016; Berard et al., 2016). However, unlike ASR, annotated speech-to-translation data is scarce, which prevents well-trained ST models. Therefore, we use CTC and pre-training for sufficient training (Bahar et al., 2019). According to the experimental results on ASR, we use PDS-Base-8 to investigate the effects of PDS on both performance and model convergence.

Setting	CTC	#Params	w/o PT	w/ PT
Transformer				
Stack-4		30M	20.3	22.9
	✓	32M	23.5	**24.0**
PDS-Base-8		29M	23.0	24.1
	✓	32M	23.9	**24.8**
SATE				
Stack-4	✓	40M	24.3	25.4
PDS-Base-8	✓	40M	24.9	**25.8**

Table 3: SacreBLEU on MuST-C En-De ST corpus. "PT" represents that initializing the ST model with the pre-trained ASR and MT models. All pre-trained models have similar performance for a fair comparison.

Table 3 shows an obvious performance gap of 2.5 BLEU between the stacked and PDS methods when the auxiliary CTC and pre-training methods are not used. This indicates that PDS helps convergence and improves ST when transcription is not available. When pre-training and CTC are available, better performance is achieved by good initialization and strong supervision. Also, PDS always outperforms the stacked method significantly.

On SATE (Xu et al., 2021), consistent improvements are achieved. The encoder of SATE is composed of an acoustic encoder and a textual encoder. We only employ PDS in the first encoder. Although an adaptor is introduced for adaptive representation, the length inconsistency issue is not solved. As a popular method, the shrink mechanism filters the acoustic representation based on the CTC predic-

	F	Ratio	Stride	Layer	Avg.
Stack					
(A)	/	2	2	12	7.11
		4	2-2	0-12	**6.76**
		8	2-2-2	0-0-12	7.48
		16	2-2-2-2	0-0-0-12	9.22
PDS					
(B)		4	2-2-1-1	3-3-3-3	6.06
	✓	4	2-2-1-1	3-3-3-3	**6.03**
(C)		8	2-2-1-2	3-3-3-3	6.58
	✓	8	2-2-1-2	3-3-3-3	**6.16**
(D)		8	2-2-2-1	2-2-6-2	6.59
		16	2-2-2-2	2-2-6-2	6.81
	✓	8	2-2-2-1	2-2-6-2	6.32
	✓	16	2-2-2-2	2-2-6-2	**6.30**
(E)		32	2-2-2-2-2	2-2-3-3-2	7.26
	✓	32	2-2-2-2-2	2-2-3-3-2	**6.74**

Table 4: Impact of representation fusion. "F" represents the representation fusion method. We report the average WER of all 4 sets on LibriSpeech.

tion (Liu et al., 2020). However, this also poses the risk of information loss. PDS provides another approach, which generates a length-matched sequence in foundational acoustic encoding.

Finally, combing the above methods, PDS achieves a new state-of-the-art performance of 25.8 BLEU score with no additional training data.

5 Analysis

Next, we study a number of interesting problems on 960h LibriSpeech.

5.1 Impact of Representation Fusion

To investigate the effect of information loss and the need of representation fusion, we compare the results with different down-sampling ratios (see Table 4).

For the vanilla stacked method in rows (A), the popular setting is to down-sample the input with a lower ratio of 4. This setting also achieves the best performance. Choosing a ratio of 2 leads to inferior WER because long but fine-grained features face the challenges of modeling. As the ratio of down-sampling increases, the performance drops significantly. This supports the point that information loss is serious in the stacked method.

For PDS, the system outperforms the stacked counterpart under the same setting of ratio $= 4$. However, we find that the fusion method does not obtain significant improvement. This might be because that the light condensation is lossless and cannot benefit from the fusion of representations.

Layer	dev		test		Avg.
	clean	other	clean	other	
2-2-2-6	3.91	9.29	4.09	9.38	6.67
2-2-4-4	3.83	9.11	4.05	9.13	6.53
2-2-6-2	3.71	8.73	3.74	9.02	**6.30**
5-5-10-10	3.19	7.79	3.57	7.69	5.56
5-5-12-8	3.15	7.83	3.38	7.79	**5.54**
5-5-15-5	3.27	7.59	3.60	7.83	5.57

Table 5: Impact of the number of layers in each stage. We report the results of Transformer with PDS-Base-16 and PDS-Deep-16 settings.

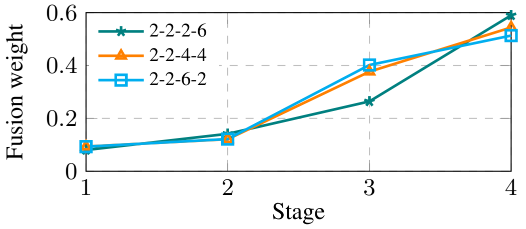


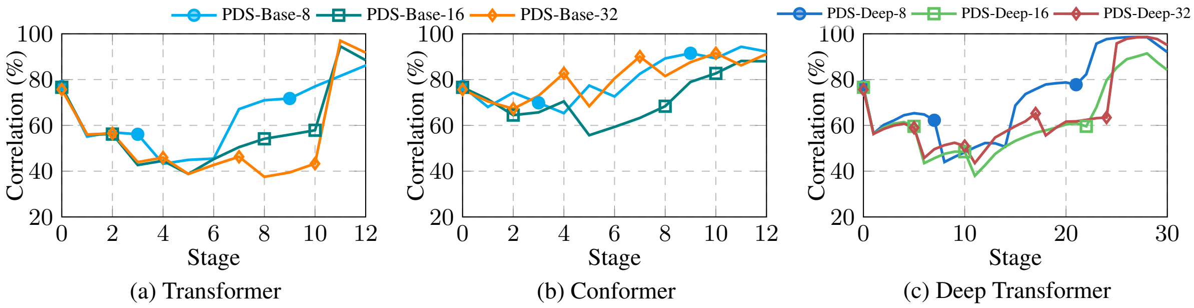
Figure 5: The fusion weights of the output representation in each stage. We consider three settings of the number of layers under PDS-Base-16.

Interestingly, the fusion method achieves consistent improvements when higher down-sampling ratios are adopted. To study it further, we add another set of experiments with a special setting in rows (D): the down-sampling ratio decreases from 16 to 8 by setting the stride of the final stage to 1. Then, we see a better WER of 6.59, which indicates less information loss under the lighter condensation. With the help of the fusion method, they achieve similar performances. This is consistent with what we expected.

5.2 Impact of Model Depth

We compare the results of the different number of layers in each stage. Table 5 shows the results on base and deep models.

For the model of 12 encoder layers, we use 2 fixed layers in the bottom 2 stages for less computation. As we described in Section 3.4, PDS achieves better performance as the number of layers increases in the intermedia stage. We think that there are two reasons. Firstly, the information loss is less compared with the top stages, and thus the encoding layers are more helpful. Secondly, sufficient encoding helps the down-sampling in the next stage, but it is not the case for the top stage. This is consistent with the previous conclusion (Huang et al., 2020). We also show the fusion weights in



(a) Transformer

(b) Conformer

(c) Deep Transformer

Figure 6: The correlation of window size of 2 in each layer of Transformer, Conformer, and deep Transformer. The marked points represent the correlation before each down-sampling.

Figure 5. The weight increases as the number of layers increases, and vice versa. The results are agreed with our design.

We also compare the results in a deep Transformer model with a 30-layer encoder. Due to the sufficient encoding in each stage, the deep model is robust in the design of the number of layers in each stage. There is no obvious performance gap across three different settings. It is very meaningful to combine PDS with the popular deep model (Pham et al., 2019) in the follow-up work.

5.3 Impact on Correlation

Unlike the stacked method, PDS runs a context interaction process after each down-sampling process. Figure 6 shows the correlation across different model architectures.

In Transformer, high correlation (about 60% $\sim$ 80%) alleviates the information loss under the setting of PDS-Base-8. As the down-sampling ratio increases, fewer layers in each stage cannot capture the context information sufficiently and thus make the degraded correlation. This leads to the performance drops, as shown in Section 4.3.

Conformer and deep Transformer show similar performance under the different settings of down-sampling. This motivates us to investigate their behavior. Despite the limited layers in each stage, Conformer always shows a high correlation by the explicit local modeling. This also demonstrates the effectiveness of Conformer. The deep Transformer alleviates the issue straightly by stacking more layers.

One interesting finding is that the correlation of top layers is very high ($> 90\%$) across all architectures. It may be affected by the strong supervision of the decoder. This inspires us to explore multitask learning in the future.

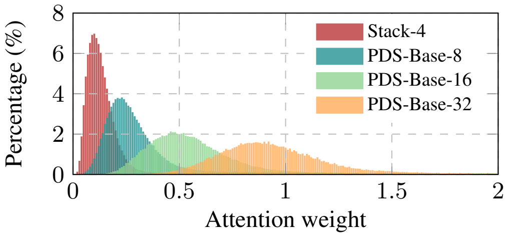


Figure 7: Distribution of summed cross-attention weights for each encoder representation on LibriSpeech test-clean set.

5.4 Distribution of Attention Weights

PDS generates the semantically complete units on the top of the encoder. We suppose that this informative representation has a greater effect on decoding. Refer to Zhang et al. (2020a), Figure 7 shows the distribution of summed cross-attention weights for each encoder representation.

Due to the fine-grained representations in the stacked method, the smaller attention weights spread across multiple relevant representations. In PDS, each representation receives greater attention as the down-sampling ratio increases. Although our method does not explicitly filter uninformative features, we argue that the stronger condensation forces the model to prefer more meaningful representations.

6 Conclusion

In this paper, we investigate the down-sampling process and shed light on the issue of information loss in the popular stacked method. This inspires us to propose a *Progressive Down-Sampling* method, which encodes the context information after each down-sampling. Furthermore, we develop a representation fusion method to combine the multi-scale information. Results on ASR and ST tasks demonstrate the effects of our method.

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3159–3166. AAAI Press.

Belen Alastruey, Gerard I. Gállego, and Marta R. Costa-jussà. 2021. Efficient transformer for direct speech translation. *CoRR*, abs/2107.03069.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019. A comparative study on end-to-end speech to text translation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, pages 792–799. IEEE.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4945–4949. IEEE.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 6224–6228. IEEE.

Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *CoRR*, abs/1612.01744.

Maxime Burchi and Valentin Vielzeuf. 2021. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. *CoRR*, abs/2109.01163.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell. *CoRR*, abs/1508.01211.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5884–5888. IEEE.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 949–959. The Association for Computational Linguistics.

Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2012–2017. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA.

Kyu J. Han, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-stride self-attention for speech recognition. In *Interspeech*

2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019, pages 2788–2792. ISCA.

Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. 2020. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3610–3614. ISCA.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Wenyong Huang, Wenchao Hu, Yu Ting Yeung, and Xiao Chen. 2020. Conv-transformer transducer: Low latency, low frame rate, streamable end-to-end speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5001–5005. ISCA.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the modality gap for speech-to-text translation. *CoRR*, abs/2010.14920.

Liang Lu, Changliang Liu, Jinyu Li, and Yifan Gong. 2020. Exploring transformers for large-scale speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5041–5045. ISCA.

Rui Na, Junfeng Hou, Wu Guo, Yan Song, and Lirong Dai. 2019. Learning adaptive downsampling encoding for online end-to-end speech recognition. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019, Lanzhou, China, November 18-21, 2019*, pages 850–854. IEEE.

Alan V Oppenheim. 1999. *Discrete-time signal processing*. Pearson Education India.

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 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021. Speechformer: Reducing information loss in direct speech translation. *CoRR*, abs/2109.04574.

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, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2613–2617. ISCA.

Vijayaditya Peddinti, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur. 2018. Low latency acoustic modeling using temporal convolution and lstms. *IEEE Signal Process. Lett.*, 25(3):373–377.

Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Müller, and Alex Waibel. 2019. Very deep self-attention networks for end-to-end speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 66–70. ISCA.

Elizabeth Salesky and Alan W. Black. 2020. Phone features improve speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2388–2397. Association for Computational Linguistics.

Elizabeth Salesky, Matthias Sperber, and Alan W. Black. 2019. Exploring phoneme-level speech representations for end-to-end speech translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1835–1841. Association for Computational Linguistics.

Khalid Sayood. 2018. Chapter 1 - introduction. In Khalid Sayood, editor, *Introduction to Data Compression (Fifth Edition)*, fifth edition edition, The Morgan Kaufmann Series in Multimedia Information and Systems, pages 1–10. Morgan Kaufmann.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 3465–3469. ISCA.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. Self-attentional acoustic models. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 3723–3727. ISCA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1810–1822. Association for Computational Linguistics.

Qiang Wang, Fuxue Li, Tong Xiao, Yanyang Li, Yinqiao Li, and Jingbo Zhu. 2018. Multi-layer representation fusion for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3015–3026. Association for Computational Linguistics.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *CoRR*, abs/2102.12122.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2619–2630. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020a. Adaptive feature selection for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2533–2544. Association for Computational Linguistics.

Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. 2020b. Feature pyramid transformer. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pages 323–339. Springer.

Shucong Zhang, Erfan Loweimi, Yumo Xu, Peter Bell, and Steve Renals. 2019. Trainable dynamic subsampling for end-to-end speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1413–1417. ISCA.

Yu Zhang, Daniel S. Park, Wei Han, James Qin, Anmol Gulati, Joel Shor, Aren Jansen, Yuanzhong Xu, Yanping Huang, Shibo Wang, Zongwei Zhou, Bo Li, Min Ma, William Chan, Jiahui Yu, Yongqiang Wang, Liangliang Cao, Khe Chai Sim, Bhuvana Ramabhadran, Tara N. Sainath, Françoise Beaufays, Zhifeng Chen, Quoc V. Le, Chung-Cheng Chiu, Ruoming Pang, and Yonghui Wu. 2021. Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *CoRR*, abs/2109.13226.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239. IEEE Computer Society.