

# Bridging the Granularity Gap for Acoustic Modeling

Chen Xu<sup>1</sup>, Yuhao Zhang<sup>1</sup>, Chengbo Jiao<sup>2</sup>, Xiaoqian Liu<sup>1</sup>, Chi Hu<sup>1</sup>,  
Xin Zeng<sup>1</sup>, Tong Xiao<sup>1,3\*</sup>, Anxiang Ma<sup>1,3</sup>, Huizhen Wang<sup>1,3</sup>, Jingbo Zhu<sup>1,3</sup>

<sup>1</sup>School of Computer Science and Engineering, Northeastern University, Shenyang, China

<sup>2</sup>Beijing University of Posts and Telecommunications

<sup>3</sup>NiuTrans Research, Shenyang, China

{xuchennlp,yoohaozhang,liuxiaoqian0319}@outlook.com

{xiaotong,maanxiang,wanghuizhen,zhujingbo}@mail.neu.edu.cn

## Abstract

While Transformer has become the de-facto standard for speech, modeling upon the fine-grained frame-level features remains an open challenge of capturing long-distance dependencies and distributing the attention weights. We propose *Progressive Down-Sampling* (PDS) which gradually compresses the acoustic features into coarser-grained units containing more complete semantic information, like text-level representation. In addition, we develop a representation fusion method to alleviate information loss that occurs inevitably during high compression. In this way, we compress the acoustic features into 1/32 of the initial length while achieving better or comparable performances on the speech recognition task. And as a bonus, it yields inference speedups ranging from  $1.20\times$  to  $1.47\times$ . By reducing the modeling burden, we also achieve competitive results when training on the more challenging speech translation task<sup>1</sup>.

## 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) (Anastasopoulos et al., 2022), how to encode the speech features effectively is an open problem. Different from modeling based on discrete tokens in natural language processing, a standard paradigm for acoustic encoding is taking the continuous features extracted by signal processing as input.

In this process, the framing operation generates a lengthy sequence consisting of fine-grained features. For example, frame-level feature sequences are dozens of times longer than the corresponding

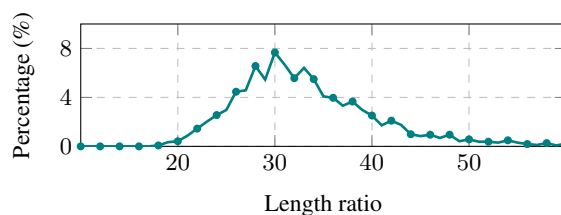


Figure 1: The distribution of the length ratio between the speech features (frame-level) and corresponding transcriptions (subword-level) on Librispeech test-clean set.

subword-level transcriptions (see Figure 1). Such an input leads to the difficulties of capturing long-distance dependencies and distributing the attention weights among semantically incomplete modeling units (Han et al., 2019). Al-Rfou et al. (2019) also demonstrates that the fine-grained character-level modeling yield significantly inferior performance compared with coarse-grained word-level counterparts. In addition, the long sequence also results in prohibitive computation costs due to the quadratic complexity of self-attention.

A popular method is to compress fine-grained features to form coarser-grained modeling units by stacking multiple down-sampling modules before encoding (Dong et al., 2018; Berard et al., 2018). Unfortunately, it does not work well when we increase the down-sampling (DS) ratio like 16 and 32 to attain semantically more complete units. This is in line with the intuition that it is difficult to compress dozens of frames directly into a single unit (Sayood, 2018). It is something like that a few principal components can not preserve all the information in the principal component analysis method (Wold et al., 1987).

We analyze the compression process in the stack method to shed light on the reason for the failure. We find that the similarity of representations in adjacent positions degrades after down-sampling,

\*Corresponding author.

<sup>1</sup>The code is available at <https://github.com/xuchennlp/S2T>.

which leads to the non-trivial issue of information loss and increases the difficulty of subsequent compression. To address this issue, we propose *Progressive Down-Sampling* (PDS), which gradually aggregates the input frame-level representations into semantically more complete text-level representations, like character-level or subword-level units. In this way, the model behaves as a natural language processing (NLP) system, which distributes the attention only among a short sequence and enables reinvesting saved computational resources into a deeper model or a bigger batch. Nevertheless, we find that information loss still occurs as the compression ratio increases. As a remedy, we combine the coarse-grained high-level representation at top layers and fine-grained low-level representations at bottom layers that may preserve information lost during compression.

Our method outperforms the standard Transformer and Conformer on the ASR and ST tasks, and has the potential for application on a wide variety of acoustic modeling tasks. It also makes a trade-off between performance and computational speedup. With an extremely high compression ratio of 32, our method achieves better or comparable performances on the ASR task while yielding inference speedups ranging from  $1.20\times$  to  $1.47\times$ . The lower ratio of 8 or 16 brings remarkable improvement and relatively modest speedup. On the more challenging ST task, PDS also helps convergence and shows competitive results.

## 2 Related Work

Unlike text that has explicit boundaries, audio is in general represented as continuous signals. Although researchers have explored straightforward modeling 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). This short frame shift allows the continuity of the speech signal, and the overlapping segments help to avoid information loss between consecutive frames.

However, the fine-grained frame-level features may not be suitable for the state-of-the-art Transformer architectures (Vaswani et al., 2017). The lengthy sequences composed of semantically incomplete units lead to the difficulties of capturing long-distance dependencies and distributing the attention weights to 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.

Motivated by the work in the efficient models (Beltagy et al., 2020), 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 long sequence modeling and the cross-attention module still suffers from the same issue.

Another line of research is to down-sample the fine-grained features for shorter sequences (Chan et al., 2015; Bahdanau et al., 2016). A popular approach is to pass the features through a stack of strided convolutional layers before encoding (Dong et al., 2018; Berard et al., 2018). But the stack method does not work well in practice due to the loss of information due to 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). The key difference is that the previous studies motivate efficient computation and only explore a modest compression ratio of 8. Recently, Andrusenko et al. (2022) reduce the lengths by 16 times at the intermediate layers. Our work aims to develop a model that takes audio as input but behaves as an NLP style to ease the modeling burden, which requires extremely high compression. Although there is no special design for efficiency, our method still yields a remarkable speedup.

Another related 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. These two methods are complementary and we will explore their combination in future work.

## 3 The Method

### 3.1 Why Is Information Lost?

Down-sampling increases the granularity of modeling units while reducing the sequence length by aggregating the adjacent features. Following previous work (Dong et al., 2018; Berard et al., 2018),

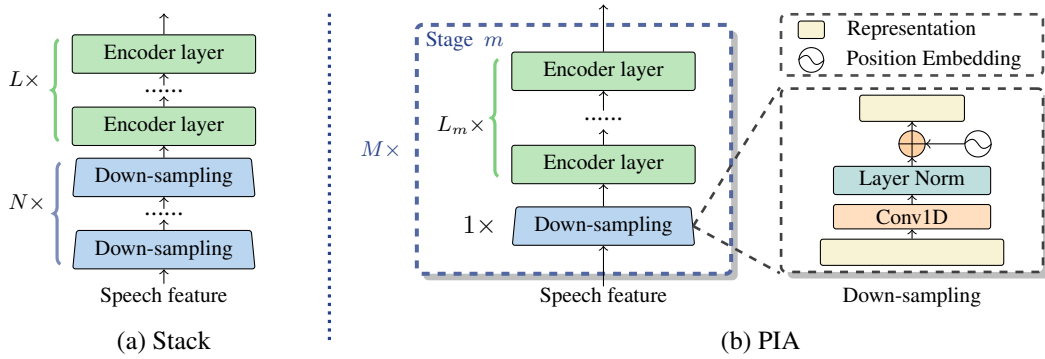


Figure 2: Comparison of the Stack and PDS methods.

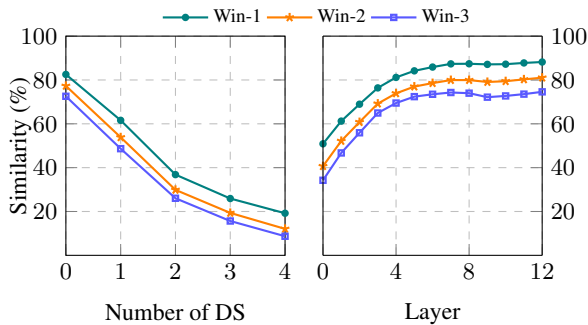


Figure 3: Left: the similarity after each down-sampling. Right: the similarity at each Layer in a standard Transformer-based ASR model. Win- $d$  represents the window size of  $d$ .

input speech features are fed into a stack of 2 convolutions with a stride of 2, followed by a number of encoder layers (see Figure 2 (a)). For generating text-level representations, it is natural to stack more down-sampling layers for a high down-sampling ratio. But it fails in our preliminary experiments (see Table 5).

This motivates us to investigate the changes in representation during down-sampling. We define the *similarity* of representation as the average cosine similarity of each unit to the surrounding units within a small window (Kim et al., 2022). High similarity indicates that the representation is easier for compression.

Firstly, we train a Transformer-based (Vaswani et al., 2017) ASR model with a down-sampling ratio of 16 by 4 stacked compression operations on the 960h LibriSpeech dataset, then show the similarity on the test-clean set. As shown in Figure 3 (Left), the input speech features have an extremely high similarity due to the overlapping framing. However, the similarity degrades drastically after each down-sampling. This indicates

that the subsequent down-sampling processes are difficult to compress the dissimilar representation while fully preserving the information. We refer to this issue as information loss caused by stacked down-sampling.

Now a new question arises: how to increase the representation similarity and alleviate the information loss? An intuitive conjecture is that the context modeling increases the similarity due to the strong preference for short-distance dependency (Sperber et al., 2018; Xu et al., 2021). Figure 3 (Right) shows the similarity at each layer of the encoder in a standard Transformer with a down-sampling ratio of 4. As we expected, the similarity gradually increases from bottom to top. This inspires us to develop a progressive method that encodes context information sufficiently before each down-sampling.

### 3.2 Progressive Down-Sampling

We propose a *Progressive Down-Sampling* (PDS) method to compress the fine-grained frame-level features into semantically more complete units, like text-level representations. See Figure 2 (b) for an overview of PDS.

The encoding is divided into representation down-sampling and context interaction processes. Given the input speech features  $H_0$ , a down-sampling module compresses it by a single 1-D convolution layer. To address varying lengths, position encoding is introduced into the normalized representation. Following the finding in Section 3.1, we use several encoder layers to capture the contextual dependencies for high similarity.

We define each run of down-sampling and interaction processes as a *stage*. The model runs for  $M$  stages and gradually obtains coarser-grained representations  $\{H_1, H_2, \dots, H_M\}$ . Note that the stack

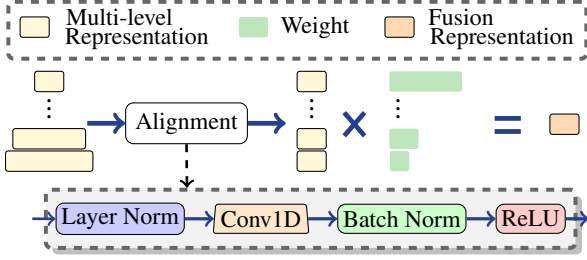


Figure 4: The representation fusion method. It aligns multi-level representations to the same shapes then combines them.

method can be seen as a specific case of PDS: it consists of two stages, where the first stage does not include the context interaction process.

Our method also draws inspirations from the field of computer vision (He et al., 2016; Wang et al., 2021) and NLP (Dai et al., 2020). While they employ a similar design concept, i.e., gradually reducing the sequence length of the representation, it remains an open problem how to compress the fine-grained and lengthy speech features into text-level representations that are easier for modeling.

### 3.3 Representation Fusion

As the inherent nature of compression, information loss still occurs inevitably, especially as the down-sampling ratio increases. Motivated by previous methods to make full use of the multi-level representations (Wang et al., 2019), one way to further alleviate this problem is to fuse the coarse-grained representation and the finer-grained representations (Zhao et al., 2017; Zhang et al., 2020b; Dai et al., 2020). Then the final output representation  $H^o$  can be defined as:

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

where  $\mathcal{F}(\cdot)$  is the fusion function<sup>2</sup>.

Now the key challenge is how to combine the representations with different lengths. The premise is to align them to the length of the coarsest-grained representation at the top of the encoder  $H_M$ . We resort to multiple simple but effective convolution modules to transform the finer-grained representations outputted in the bottom stages to the shape of  $H_M$ . Concretely, the stride and kernel size of convolution for the representation  $H_m$  are set to the  $L_{H_m}/L_{H_M}$ , where  $L_H$  represents the sequence length of the representation  $H$ .

<sup>2</sup>We omit the input feature because it is extracted by signal processing rather than encoding by the model.

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.

Drawing on the design of the convolution module in Conformer, the representation fusion method is shown in Figure 4. After alignment operation  $\mathcal{A}(\cdot)$ , we employ a simple linear combination for fusion:

$$\mathcal{F}(H_1, \dots, H_M) = \sum_{m=1}^M W_k \cdot \text{LN}(\mathcal{A}(H_m)) \quad (2)$$

where  $\text{LN}(\cdot)$  is the layer normalization function.  $W_m \in \mathbb{R}$  is a learnable scalar to weight the aligned representations. The weights are initialized to the same values, then learned together with other parameters during training.

### 3.4 PDS Settings

In this work, we construct 8 settings under different encoder depths and down-sampling ratios (see Table 1) with the following design philosophy<sup>3</sup>:

- During down-sampling, a bigger window size involves more context information while increasing the difficulty of down-sampling due to the lower similarity. Referring to framing, we use an empirical setting of kernel size = 5 and stride = 2.
- Different from the design in the field of computer vision (He et al., 2016), we keep the same hidden dimensions in the whole encoding process. The detailed comparisons and analyses are shown in Appendix B.1.

<sup>3</sup>The sophisticated down-sampling designs, fusion approaches, and optimized settings could potentially perform better, but these are beyond the scope of this paper. We briefly opt for the simple design and expect to bring insights for future studies.



Group	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	3.80	8.90	4.40	9.00	-	-
	Stack-4					30M	3.88	9.26	4.49	9.42	6.80	1.00×
	PDS-Base-8					30M	3.57	8.63	3.85	8.58	<b>6.11 (-0.69)</b>	0.99×
	PDS-Base-16					30M	3.71	8.73	3.74	9.02	6.26 (-0.54)	1.14×
	PDS-Base-32					31M	4.13	9.31	4.21	9.31	6.69 (-0.11)	1.20×
(B)	Stack-4*	12	512	2048	8	71M	3.20	8.00	3.40	7.90	-	-
	Stack-4					71M	3.53	8.15	3.67	7.96	5.78	1.00×
	PDS-Base-8					75M	3.17	7.46	3.47	7.47	<b>5.35 (-0.43)</b>	1.08×
	PDS-Base-16					76M	3.34	7.73	3.37	7.85	5.53 (-0.25)	1.34×
	PDS-Base-32					82M	3.32	7.94	3.64	7.85	5.65 (-0.13)	1.47×
(C)	Stack-4	30	256	2048	4	53M	3.80	8.51	4.33	8.61	6.25	1.00×
	PDS-Deep-8					53M	3.34	7.90	3.50	7.79	5.59 (-0.66)	1.03×
	PDS-Deep-16					54M	3.15	7.83	3.38	7.79	<b>5.50 (-0.75)</b>	1.19×
	PDS-Deep-32					55M	3.26	7.77	3.33	7.88	5.52 (-0.73)	1.27×
Conformer												
(D)	Stack-4†	12	256	2048	4	50M	-	-	3.05	8.36	-	-
	Stack-4					45M	3.02	7.23	3.21	7.29	5.15	1.00×
	PDS-Base-8					46M	2.87	7.07	3.01	7.20	4.98 (-0.17)	0.97×
	PDS-Base-16					46M	2.93	6.97	2.99	7.03	<b>4.93 (-0.22)</b>	1.14×
	PDS-Base-32					47M	2.95	7.15	3.04	7.12	5.02 (-0.13)	1.20×
(E)	Stack-4‡	12	512	2048	8	109M	2.90	6.60	3.00	6.70	-	-
	Stack-4					113M	2.77	6.47	2.82	6.72	4.66	1.00×
	PDS-Base-8					113M	2.72	6.54	3.03	6.38	4.65 (-0.01)	0.97×
	PDS-Base-16					114M	2.73	6.31	2.72	6.40	<b>4.52 (-0.14)</b>	1.23×
	PDS-Base-32					119M	2.70	6.67	2.95	6.81	4.75 (+0.09)	1.25×

Table 2: WER on the 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 attention heads. #Params: the number of parameters. The speedup is computed during inference on the test-clean set with a beam size of 5 and batch size of 100k (except 50k for a bigger Conformer due to the GPU limitation). \*, †, and ‡ stand for the results reported in fairseq<sup>4</sup>, wenet<sup>5</sup>, and espnet<sup>6</sup> respectively.

- We allocate fewer layers at the bottom stage for efficient computation due to the longer sequence. The major computations are concentrated in the intermediate stages. This leads to sufficient encoding and computation acceleration, as shown in Table 6.

## 4 Experiments

We evaluate our method on the LibriSpeech and AISHELL-1 ASR datasets, and the MuST-C En-De ST dataset. Details about the data and model settings are described in Appendix A.

### 4.1 Results of ASR

**LibriSpeech** Table 2 shows the results on the 960h LibriSpeech corpus. We compare methods on the

<sup>4</sup>[https://github.com/pytorch/fairseq/blob/main/examples/speech\\_to\\_text/docs/librispeech\\_example.md](https://github.com/pytorch/fairseq/blob/main/examples/speech_to_text/docs/librispeech_example.md)

<sup>5</sup><https://github.com/wenet-e2e/wenet/blob/main/examples/librispeech/s0/README.md>

<sup>6</sup><https://github.com/espnet/espnet/blob/master/egs/librispeech/asr1/RESULTS.md>

Transformer and Conformer with different encoder layers and hidden dimensions. We use Stack-4 as the baseline model (see Table 1 for the setting). For a fair comparison, the number of model parameters in each group is similar.

For a popular setting of 12 encoder layers with 256 hidden dimensions in group (A), PDS achieves a very high down-sampling ratio of 32 with a slight improvement. As a bonus, it yields a speedup of 1.20×. In real scenarios, the saved computational resource can be reinvested in a bigger batch or improved model capacity. As the down-sampling ratio decreases, the performance improves significantly. Similar phenomena are observed on the wider Transformer with 512 hidden dimensions in group (B), where our method benefits more in terms of speedup.

Interestingly, we find that the deep models with 30 encoder layers in group (C) eliminate the performance gap under different down-sampling ratios. PDS compresses the representation to 1/32 of the

Setting	w/o CTC		w/ CTC	
	dev	test	dev	test
Stack-4	5.42	5.80	4.96	5.55
PDS-Base-8	<b>5.09</b>	<b>5.59</b>	<b>4.72</b>	<b>5.24</b>
PDS-Base-16	5.30	5.78	4.76	5.28
PDS-Base-32	5.43	5.85	5.17	5.63

Table 3: WER on the AISHELL-1 ASR corpus.

initial length while achieving a considerable relative reduction of 0.73 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 also has practical advantages in industrial scenarios where deep models are preferred.

We observe two interesting phenomena in the experiments on Conformer architecture in groups (D) and (E). Firstly, Conformer bridges the performance gap between the stack and PDS method. We speculate that the Conformer integrates the relative position encoding (RPE) to improve the generalization to the variant length, which may be helpful for the long sequence encoding and reduces the benefit of our method. Secondly, PDS with a ratio of 16 outperforms its counterpart with a lower ratio of 8. Conformer enhances local interaction among neighbor contexts by convolution module, which leads to a higher similarity, as shown in Figure 6.

We also notice that PDS works better on clean subsets than other subsets, especially under a high down-sampling ratio. This demonstrates that down-sampling is more challenging on noisy audio, where it is difficult to distinguish the meaningful information. We will explore more robust methods in future work.

**AISHELL-1** We observe that our method with a down-sampling ratio of 32 slightly under-performs the baseline on the AISHELL-1 corpus, as shown in Table 3. As we employ a character vocabulary, the length of the compressed sequence may be less than transcription, making excessive compression and invalid connectionist temporal classification (CTC) computation. This inspires us to explore better solutions, e.g., a self-adaptive compression method that dynamically treats each sample. We further discuss it in Section Limitations.

## 4.2 Results of ST

End-to-end ST has become popular recently (Berrard et al., 2016). However, unlike ASR, annotated

Setting	CTC	#Params	w/o PT	w/ PT
Transformer				
Stack-4		30M	20.2	23.2
	✓	32M	24.0	<b>24.5</b>
PDS-Base-8		29M	23.2	24.5
	✓	32M	24.2	<b>24.8</b>
SATE - Transformer				
Stack-4	✓	40M	24.8	25.3
PDS-Base-8	✓	40M	25.5	<b>25.6</b>
SATE - Conformer				
Stack-4	✓	55M	25.5	25.9
PDS-Base-8	✓	55M	25.8	<b>26.4</b>
SATE - Conformer - Unrestricted				
Stack-4	✓	130M	-	27.9
PDS-Base-8	✓	134M	-	<b>28.7</b>

Table 4: SacreBLEU on the MuST-C En-De ST corpus. "PT" represents the ST models are initialized with the pre-trained ASR and MT models.

speech-to-translation data is scarce, making it challenging for well-trained ST models. Therefore, CTC and pre-training methods are used for sufficient training (Bahar et al., 2019; Zhang et al., 2022). According to the results of ASR, we select PDS-Base-8 to investigate the effects of PDS on both performance and model convergence.

Table 4 shows a substantial performance gap of 3.0 BLEU points between the stack and PDS methods when the auxiliary CTC and pre-training methods are not used. This indicates that bridging the granularity gap helps convergence and improves ST when transcription is not available. With CTC and pre-training, better performance is achieved by strong supervision and better initialization. Also, PDS outperforms the stack method significantly.

Better architecture of SATE (Xu et al., 2021) brings consistent improvements. The encoder of SATE is composed of an acoustic encoder and a textual encoder. We only employ PDS in the acoustic encoder. Although an adaptor is introduced for adaptive representation, the length inconsistency issue is not solved in the original implementation. As a popular method, the shrink mechanism filters the acoustic representation based on the CTC prediction (Dong et al., 2021). However, it also poses the risk of information loss due to inaccurate predictions. PDS provides another approach by generating a length-matched sequence in foundational acoustic encoding.

Combing the CTC and pre-training methods, PDS achieves a competitive performance of 26.4

Group	F	Ratio	Stride	Layer	Avg.
Stack					
(A)	/	2	2	12	7.06
		4	2-2	0-12	<b>6.80</b>
		8	2-2-2	0-0-12	7.43
		16	2-2-2-2	0-0-0-12	9.17
PDS					
(B)	✓	4	2-2-1-1	3-3-3-3	6.01
		4	2-2-1-1	3-3-3-3	<b>5.99</b>
(C)	✓	8	2-2-1-2	3-3-3-3	6.53
		8	2-2-1-2	3-3-3-3	<b>6.11</b>
(D)	✓	8	2-2-2-1	2-2-6-2	6.55
		16	2-2-2-2	2-2-6-2	6.77
		8	2-2-2-1	2-2-6-2	6.28
		16	2-2-2-2	2-2-6-2	<b>6.26</b>
(E)	✓	32	2-2-2-2-2	2-2-3-3-2	7.21
		32	2-2-2-2-2	2-2-3-3-2	<b>6.69</b>

Table 5: Impact of representation fusion. "F" represents the representation fusion method. We report the average WER of all 4 sets of LibriSpeech.

BLEU scores without additional training data. Under the more challenging unrestricted setting, PDS provides a better acoustic representation and yields a remarkable improvement of 0.8 BLEU points over the stack method.

## 5 Analysis

Next, we study a number of interesting problems on LibriSpeech. We present the comparisons with previous work and more analyses in Appendix B.

### 5.1 Impact of Representation Fusion

To investigate the impact of information loss and the importance of representation fusion, we compare the results under different down-sampling ratios (see Table 5).

The standard setting for the stack method is to down-sample the input with a lower ratio of 4. This setting also achieves the best performance in group (A). A lower ratio of 2 leads to inferior WER because long but fine-grained features face the modeling challenges. As the ratio of down-sampling increases, the performance drops drastically. This supports the point that information loss is severe in the stack method.

The PDS method outperforms the stack method under the same setting of ratio = 4. However, we find that the fusion method does not obtain significant improvement. This may be because the lightweight compression 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	6.63
2-2-4-4	3.83	9.11	4.05	9.13	6.49
2-2-6-2	3.71	8.73	3.74	9.02	<b>6.26</b>
5-5-10-10	3.19	7.79	3.57	7.69	5.52
5-5-12-8	3.15	7.83	3.38	7.79	<b>5.50</b>
5-5-15-5	3.27	7.59	3.60	7.83	5.53

Table 6: Impact of the number of layers in each stage. We report the results of Transformer with PDS-Base-16 and PDS-Deep-16 settings.

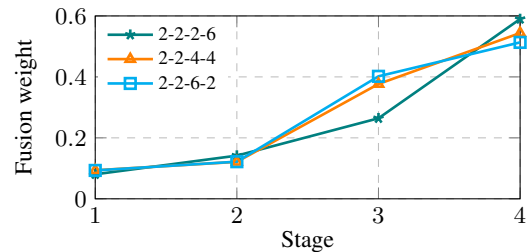


Figure 5: The fusion weights of the output representation in each stage. We consider three settings of the number of layers under PDS-Base-16.

Interestingly, the fusion method achieves consistent improvements when higher down-sampling ratios are employed. To study it further, we design another set of experiments with a special setting in group (D): the down-sampling ratio decreases from 16 to 8 by setting the stride of the final stage to 1. Then, we achieve a better performance of 6.55 WER points, which indicates less information loss under a slighter compression. With the help of the fusion method, two settings achieve similar performances.

### 5.2 Impact of Model Depth

We compare the performance of the different number of layers in each stage. Table 6 shows the results on base and deep models.

For the model of 12 encoder layers, we assign 2 layers in the bottom 2 stages for less computation cost. As described in Section 3.4, PDS achieves better performance as the number of layers increases in the intermediate stages. There are two reasons. Firstly, the information loss in the intermediate stages is less compared with the top stage, and thus the increased encoding layers are helpful. Secondly, sufficient encoding provides high similarity and helps the down-sampling for the next stage, but it is not the case for the top stage. This is consistent with the previous conclusion (Huang et al., 2020).

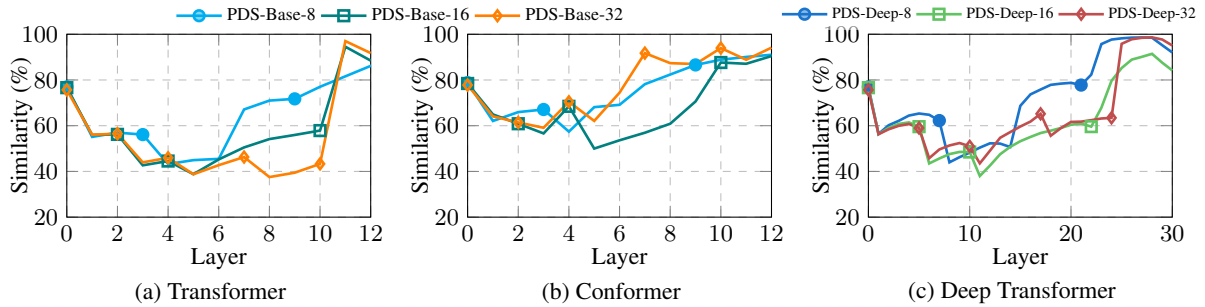


Figure 6: Similarities of window size of 2 in each layer of Transformer, Conformer, and deep Transformer. The marked points represent the similarity before each down-sampling.

We also show the fusion weights in Figure 5. The weight increases as the number of layers increases, and vice versa.

Furthermore, we compare the results of deep Transformer models 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. There is no obvious performance gap across the three different settings. It is very meaningful to combine PDS with the popular deep models (Pham et al., 2019) in the follow-up work.

### 5.3 Impact on Similarity

Unlike the stack method, PDS performs the context interaction process after each down-sampling process. Figure 6 shows the similarity across different model architectures.

In Transformer, high similarity (about 60% ~ 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 similarity and worse performance.

Despite the limited layers in each stage, Conformer always shows high similarities due to explicit local modeling. This also demonstrates the effectiveness of Conformer. The deep Transformer alleviates the issue directly by stacking more encoder layers.

One interesting finding is that the similarity of top layers is very high (> 90%) across all architectures, which may be due to the effect of the direct supervision from the decoder. This inspires us to explore multi-task learning methods by injecting explicit training objectives into intermediate stages.

### 5.4 Distribution of Attention Weights

Our method bridges the granularity gap and generates semantically more complete units, e.g., text-

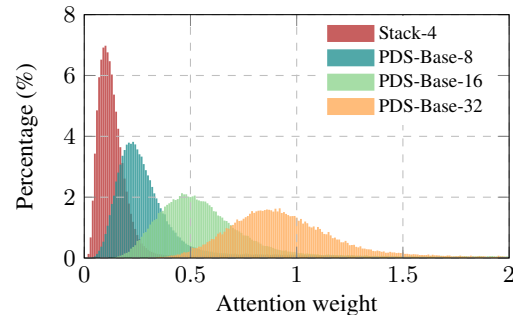


Figure 7: Distribution of summed cross-attention weights for each encoder representation on LibriSpeech test-clean set.

level representations. We suppose that this informative representation has a greater effect on text generation. Referring to Zhang et al. (2020a), Figure 7 shows the distribution of summed cross-attention weights for each encoder representation.

The fine-grained representations in the stack method have a remarkable granularity gap with the text representations. Therefore, the smaller attention weights must spread across multiple relevant representations in cross-attention operation, which makes it hard to capture complete information for text generation. In our method, each representation receives greater weights as the down-sampling ratio increases. This indicates that each unit has a more meaningful contribution to the generation, and it is easier to capture the relevant source information.

## 6 Conclusion

In this paper, we explore how to compress the fine-grained frame-level acoustic features to the semantically complete text-level units, like character-level and subword-level representations. To alleviate this granularity gap, we first investigate the down-sampling process and reveal the risk of information loss in the popular stack method. This



inspires us to propose *Progressive Down-Sampling*, which gradually attains the coarser-grained representation during encoding. Furthermore, we develop a representation fusion method to combine the high-level and low-level information, which is important for high down-sampling ratios. By this, we are the first to compress the acoustic features with a ratio of 32 on the ASR task while achieving comparable or even better performances. The more challenging ST task demonstrates that the alleviated granularity gap facilitates convergence effectively.

## 7 Limitations

Many challenges remain in the follow-up of our work. Here are some limitations that we intend to resolve in the future:

- We need a more robust method for compression. Although our method achieves consistent improvements in most experiments, we notice that the benefit is limited in the noisy sets, especially under a high down-sampling ratio. This drives us to develop a more robust down-sampling method for preserving meaningful information even with high compression.
- Our method compresses all the input acoustic features with the same ratio, where the ratio is determined according to the whole dataset. However, the speed of each audio is different, which results in obstacles to unified down-sampling. Ideally, each sample should be compressed with a self-adaptive ratio.

## Acknowledgement

The authors would like to thank anonymous reviewers for their insightful comments. This work was supported in part by the National Science Foundation of China (No. 62276056), the National Key R&D Program of China, the China HTRD Center Project (No. 2020AAA0107904), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Yunnan Provincial Major Science and Technology Special Plan Projects (No. 202103AA080015), the Fundamental Research Funds for the Central Universities (Nos. N2216016, N2216001, and N2216002), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No. B16009).

## 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.
- Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marceley Zanon Boito, Ondrej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gabbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Vera Kloudová, Surafel Melaku Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Miguel Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. [Findings of the IWSLT 2022 evaluation campaign](#). In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 98–157. Association for Computational Linguistics.
- Andrei Andrusenko, Rauf Nasretidinov, and Aleksei Romanenko. 2022. [Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition](#). *CoRR*, abs/2208.07657.
- 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.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *CoRR*, abs/2004.05150.
- Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. [End-to-end automatic speech translation of audio-books](#). 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.
- Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. [AISHELL-1: an open-source mandarin speech corpus and a speech recognition baseline](#). In *20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment, O-COCOSDA 2017, Seoul, South Korea, November 1-3, 2017*, pages 1–5. IEEE.
- 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.
- Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. 2020. [Funnel-transformer: Filtering out sequential redundancy for efficient language processing](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- 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.
- Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021. [Consecutive decoding for speech-to-text translation](#). In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12738–12748. AAAI Press.
- 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.
- Sehoon Kim, Amir Gholami, Albert E. Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. 2022. [Squeezeformer: An efficient transformer for automatic speech recognition](#). In *NeurIPS*.
- 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: Un-supervised 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.
- 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.
- Yuhao Zhang, Chen Xu, Bojie Hu, Chunliang Zhang, Tong Xiao, and Jingbo Zhu. 2022. [Improving end-to-end speech translation by leveraging auxiliary speech and text data](#). *CoRR*, abs/2212.01778.



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.

## A Experimental Details

### A.1 Datasets and Pre-processing

The datasets are from three benchmarks:

- **LibriSpeech** is a publicly available read English ASR corpus, which consists of 960-hour 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, including dev-clean, dev-other, test-clean, and test-other. The average WER is computed on the concatenation of all four subsets.
- **AISHELL-1** is a publicly available Chinese Mandarin speech corpus, which consists of 170-hour training data (Bu et al., 2017). We select the model on the dev set and report results WER on the dev and test sets.
- **MuST-C** 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 pre-processing, we follow the common recipes in fairseq toolkit<sup>7</sup>, which removes the utterances of more than 3,000 frames or fewer than 5 frames. We only use the speed perturbation in the experiments built on the AISHELL-1. The 80-channel Mel filter bank features are extracted by a 25ms window with a stride of 10ms. We learn SentencePiece<sup>8</sup> segmentation with a size of 10,000 for the LibriSpeech and MuST-C datasets, and use 4231 characters for the AISHELL-1 dataset. For the MuST-C ST dataset, we use a shared vocabulary for the source and target languages.

<sup>7</sup><https://github.com/pytorch/fairseq>

<sup>8</sup><https://github.com/google/sentencepiece>

### A.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 the LibriSpeech ASR task, 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 the same as the encoder. CTC (Graves et al., 2006) multi-task learning is not used due to the very modest improvement in our preliminary experiments.

For the AISHELL-1 ASR task, we evaluate our method on Transformer (Vaswani et al., 2017). The encoder consists of 12 layers and the decoder consists of 6 layers. Each layer comprises 256 hidden units, 4 attention heads, and 2,048 feed-forward hidden units. The weight for CTC multi-task learning is set to 0.3.

For the ST task, we evaluate our method on Transformer and SATE (Xu et al., 2021). Except that the knowledge distillation method is not used for simplicity, we follow the settings of SATE. 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. CTC is employed with a weight of 0.3 for better convergence. Similar to Xu et al. (2021), we also consider both restricted and unrestricted scenarios. Under the restricted setting, the ASR and MT models are pre-trained with the MuST-C En-De data. Under the unrestricted setting, we use the additional LibriSpeech ASR corpus and Opensubtitle En-De MT corpus for pre-training. We also use Conformer as the acoustic encoder and widen the model by increasing the hidden size to 512 and attention heads to 8.

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 on 1 NVIDIA TITAN RTX GPU. The CTC and language model re-scoring methods are not used. We report WER

Group	Arch	Ratio	Layer	Hidden Size	#Params.	dev		test		Avg.
						clean	other	clean	other	
Stack-4										
(A)	CTC Enc-Dec	4	12	256	20M 30M	5.63 3.88	13.53 9.26	5.73 4.49	13.39 9.42	9.50 <b>6.80</b>
PDS-8										
(B)	CTC	2-2-1-2	3-3-3-3	256-256-256-256	20M	5.06	11.97	5.26	11.90	8.49
			3-3-3-3	192-256-256-320		5.16	12.12	5.29	12.00	8.58
			5-3-3-5	192-224-224-256		4.97	11.89	5.14	11.83	<b>8.40</b>
(C)	Enc-Dec	2-2-1-2	3-3-3-3	256-256-256-256	30M	3.57	8.63	3.85	8.58	6.11
			3-3-3-3	192-256-256-320		3.43	8.38	3.71	8.48	<b>5.95</b>
			5-3-3-5	192-224-224-256		3.43	8.30	3.97	8.60	6.03
PDS-16										
(D)	CTC	2-2-2-2	2-2-6-2	256-256-256-256	20M	5.66	12.75	5.89	12.72	9.19
			2-2-6-2	192-224-256-320		5.56	12.50	5.69	12.77	9.07
			3-3-9-3	160-192-224-256		5.23	11.80	5.27	11.81	<b>8.47</b>
(E)	Enc-Dec	2-2-2-2	2-2-6-2	256-256-256-256	30M	3.71	8.73	3.74	9.02	<b>6.26</b>
			2-2-6-2	192-224-256-320		3.92	9.18	4.13	9.00	6.51
			3-3-9-3	160-192-224-256		3.79	8.78	4.19	8.83	6.35

Table 7: Comparison of the settings of hidden dimension. Attention heads are set to 4 and the feed-forward dimension is 4 times the hidden dimension.

and case-sensitive SacreBLEU for ASR and ST tasks, respectively.

## B Additional Analyses

### B.1 Comparison of Dimension Settings

Researchers explore similar pyramid architectures in the field of computer vision (He et al., 2016; Wang et al., 2021) and acoustic encoding (Burchi and Vielzeuf, 2021). A typical setting is that the hidden dimension increases from bottom to top, while the sequence length decreases accordingly. Although this is reasonable to keep the same complexity, the detailed design needs more tuning efforts.

We make a preliminary exploration of the dimension design on the ASR task. The final output dimension is defined as  $d$ , and we consider three settings:

- *Same*. This is the basic setting, and the hidden dimensions across the whole encoder are  $d$ .
- *Width-growth*. The hidden dimensions increase from bottom to top, and the middle dimension is set to  $d$ . This makes a wider model on the topmost stage under the same model parameters.
- *Depth-growth*. The hidden dimensions increase from bottom to top, and the dimension of the topmost stage is set to  $d$ . This makes a

deeper model under the same model parameters.

In Table 7, we evaluate results of two popular ASR architectures: CTC-based and encoder-decoder models. CTC limits that the input length must be longer than the corresponding label, so we do not build the experiments with a down-sampling ratio of 32. For each group of experiments, we list the results of *same*, *width-growth*, and *depth-growth* in turn. In the CTC-based model in group (B) and group (D), the *depth-growth* setting is superior under both two down-sampling ratios, where the parameters are more efficient than other settings. Especially under a high down-sampling ratio of 16, the improvement is more significant due to the more sufficient encoding before down-sampling, as shown in Section 5.3. The *wider-growth* setting only achieves the comparable performance with the *same* setting. This also demonstrates that the deeper models have more potential than the wider models under the same parameters (Wang et al., 2019).

However, different results appear in the encoder-decoder models. As shown in group (C) and group (E), the *width-growth* and *depth-growth* settings can not bring consistent improvements. We conjecture that the CTC-based model suffers from a heavy encoding burden due to the encoder-only modeling. Therefore, the parameter-efficient designs of the growth settings improve the model capacity re-

Model	Ratio	Layer	Hidden Size	#Params.	dev		test		Avg.	Speedup
					clean	other	clean	other		
Conformer <sup>†</sup>	4	16	174	13.0M	5.31	13.60	5.41	13.24	9.33	-
Conformer					4.76	12.07	4.72	12.19	<b>8.36</b>	1.00×
Eff Conformer <sup>†</sup>	2-2-2	5-5-5	120-168-240	13.3M	4.46	11.36	4.61	11.29	7.88	-
Eff Conformer					4.25	11.14	4.35	10.99	<b>7.56</b>	1.12×
PDS	2-2-2	5-5-5	174-174-174	13.0M	4.35	11.66	4.39	11.61	7.93	1.26×
PDS			120-168-240	12.8M	4.04	10.87	4.16	10.71	<b>7.37</b>	1.34×
-Fusion			120-168-240	12.5M	4.29	11.02	4.29	11.18	7.65	1.36×

Table 8: Comparison with effective Conformer (Burchi and Vielzeuf, 2021). Note that <sup>†</sup> represents that the results are reproduced by released code. We run other experiments based on our codebase.

markably. The decoder alleviates the generation burden, which leads to the modest improvement of the growth settings. There are similar results in Burchi and Vielzeuf (2021), where the improvements in RNN-T models reduce compared with the CTC-based models.

According to the above results, we use the *same* setting in our experiments for simplification. We will explore parameter-efficient designs in the future.

## B.2 Comparison with Previous Work

Burchi and Vielzeuf (2021) propose the efficient Conformer, which implements the progressive down-sampling by the modified convolutional module or the strided self-attention module. Compared with the efficient Conformer, our method does not rely on the specific architecture. We use a lightweight and pluggable module for down-sampling, which allows flexible integration with other methods.

To demonstrate the effectiveness of our method, we construct the comparison of our method and efficient Conformer on the ASR task. Following Burchi and Vielzeuf (2021), we learn Sentence-Piece segmentation with a size of 256. The grouped multi-head attention is a general method and is not used for a fair comparison. Except that the models are trained for 100 epochs rather than 450 epochs for fast comparison, we use the same hyperparameters.

As shown in Table 8, our method achieves a reduction of 0.19 WER points compared with efficient Conformer. We also show the result of PDS with the *same* settings. In this more challenging CTC-based Conformer model, the *width-growth* setting is more parameter-efficient. Without the representation fusion method, we only achieve comparable performance with the efficient Conformer. This proves the importance of the fusion method

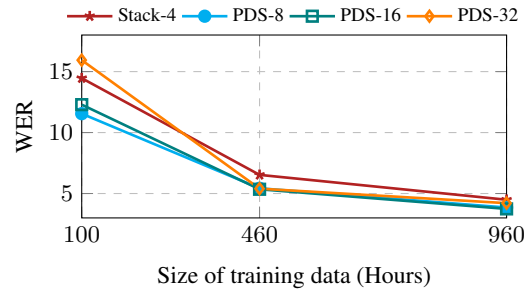


Figure 8: Comparison of varying amounts of training data. We report the WER on the test-clean set.

that alleviates information loss by combining multi-level representations.

In terms of speedup, our method encodes with more efficient computation due to the simplicity of design. It also shows that the representation fusion is lightweight but brings significant improvements. Most efficient attention variants can be integrated into our method, which has enormous potential for fast inference. But we focus on the design of the basic architecture and leave the exploration to the future.

## B.3 Low Resource Setting

Results on ST show that our method helps convergence significantly. To further verify it, we compare the results of varying amounts of training data. We train the base Transformer of 12 encoder layers on the LibriSpeech with common subsets of 100h, 460h, and all 960h training data. We select the appropriate hyperparameter for each model, including learning rate, batch size, and CTC multi-task learning. The WER on the test-clean set is reported in Figure 8.

Under the more challenging setting where only 100 hours are available, the lower down-sampling ratio of 8 yields a remarkable improvement of 2.91 WER. Our method compresses the representations

into coarser-grained units, which eases the burden of attention calculation. But excessive compression leads to degraded performance due to inferior CTC computation. One precondition for CTC loss is that the length of the input must be longer than the length of the corresponding label, which leads to invalid CTC learning for some samples. This is consistent with our previous conclusion: we need to develop a self-adaptive method that down-samples each sample with moderate compression. Increasing the training data also brings consistent improvements across different compression ratios.



## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
*Section 7.*
- A2. Did you discuss any potential risks of your work?  
*We propose a method for acoustic encoding, which does not have any risks.*
- A3. Do the abstract and introduction summarize the paper’s main claims?  
*Section 1.*
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

*Section 4.*

- B1. Did you cite the creators of artifacts you used?  
*Section 4 and Appendix A.*
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*Fairseq is open-sourced under MIT license. We implement our method based on it.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
*Section 4 and Appendix A.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*We use open datasets and cite the related papers.*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*Appendix A.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Appendix A.*

### C Did you run computational experiments?

*Section 4.*

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?  
*Section 4.1.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Section 4.1 We will release the code after review.*

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*Section 4 and 5.*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Section 4.1 and Appendix.*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*No response.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*