Multi-Encoder Learning and Stream Fusion for Transformer-Based End-to-End Automatic Speech Recognition

Timo Lohrenz, Zhengyang Li, Tim Fingscheidt

Technische Universität Braunschweig
Institute for Communications Technology
Schleinitzstr. 22, 38106 Braunschweig, Germany

t.lohrenz@tu-bs.de, zhengyang.li@tu-bs.de, t.fingscheidt@tu-bs.de

Abstract
Stream fusion, also known as system combination, is a common technique in automatic speech recognition for traditional hybrid hidden Markov model approaches, yet mostly unexplored for modern deep neural network end-to-end model architectures. Here, we investigate various fusion techniques for the all-attention-based encoder-decoder architecture known as the transformer, striving to achieve optimal fusion by investigating different fusion levels in an example single-microphone setting with fusion of standard magnitude and phase features. We introduce a novel multi-encoder learning method that performs a weighted combination of two encoder-decoder multi-head attention outputs only during training. Employing then only the magnitude feature encoder in inference, we are able to show consistent improvement on Wall Street Journal (WSJ) with language model and on Librispeech, without increase in runtime or parameters. Combining two such multi-encoder trained models by a simple late fusion in inference, we achieve state-of-the-art performance on transformer-based models on WSJ with a significant WER reduction of 19% relative compared to the current benchmark approach.

Index Terms: End-to-end speech recognition, information fusion, multi-encoder learning, transformer, phase features

1. Introduction
In recent years a paradigm shift in automatic speech recognition (ASR) research is seen towards the replacement of established hybrid hidden Markov model (HMM) based approaches [1] by end-to-end trained neural networks making pronunciation dictionaries and phonetic modeling techniques obsolete [2]. Proposed methods for end-to-end training include connectionist temporal classification (CTC) [3], recurrent neural network transducers (RNN-T) [4] and recent attention-based encoder-decoder (AED) models, namely the listen-attend-and-speak (LAS) [5] and the transformer model [6]. While the LAS models employ recurrent connections in the typical encoder-decoder structure of the end-to-end models, transformer models rely entirely on the attention mechanism to capture temporarily relevant information in speech [7]. On large datasets such as Librispeech [8], transformer models outperform hybrid speech recognition already by a large margin [9].

For the well-established hybrid speech recognition stream fusion approaches can be classified into three categories based on which stage in the system fusion is performed: early fusion—combination in the input feature domain [10] [11] [12], middle fusion—combination of an intermediate information representation [13] [14] [15] (e.g., state likelihoods), or late fusion—combination of system outputs (e.g., word hypotheses [16], output posterior, confusion networks [17], or lat-
tics [18]). A prominent task for fusion is audiovisual automatic speech recognition (AV-ASR) [19] [20], employing additional visual sensors to increase robustness in noisy conditions. In single-channel settings, usually different feature representations are used for fusion (e.g., filterbank and iMLLR features [12], different short-time Fourier transform window sizes [15]), or as here, standard magnitude features with phase features [21], or multiple acoustic models [22] [23].

Concerning the fusion of additional information into end-to-end transformer models, the few existing approaches stem from audiovisual automatic speech recognition [24] [25] and neural machine translation [26], where additional encoders are used to gather visual speech information or contextual information, respectively. Recent successful non-fusion techniques for end-to-end models are multi-task learning, e.g., by using a combination of CTC and attention-based losses [27] [28], and augmentation techniques such as spectral augmentation [29]. Those methods improve neural networks by adding more variety to the trained models either by composite losses or by randomly withholding information in the input features. An unexplored approach to add such variety, strongly related to fusion, is to use multiple encoders during training of the transformer model. Some context-aware approaches in neural machine translation use additional context encoders [30] to incorporate previous context of a sentence to achieve a better translation, while in [26] it has been found that results improve even if such context is ignored during inference.

In this paper we adopt and modify fusion techniques from hybrid ASR to transformer-based end-to-end speech recognition on an exemplary audio-only fusion task by combining the common magnitude-based feature representations with additional phase-based features. To elaborate the best possible fusion method we apply simple input feature and output posterior combination methods as well as middle fusion schemes, that use two different encoders to perform fusion of the respective encoder-decoder multi-head attention outputs. For this middle fusion approach we investigate several variants comprising an optional sharing of the encoder-decoder attention parameters as well as different paradigms combining the outputs thereof. In addition we explore a novel method which we dub multi-encoder learning (MEL) that uses both individual encoders only during training, thereby increasing robustness of the standard non-fusion transformer even during single encoder inference.

The paper is structured as follows: In Section 2, we introduce known and novel fusion and learning approaches to end-to-end model architectures. Section 3 describes the fusion experiment setup on Wall Street Journal (WSJ) and Librispeech, while corresponding results are reported and discussed in Section 4. The paper is concluded in Section 5.
2. Fusion Methods for End-to-End ASR

2.1. Early Fusion

When it comes to fusion in end-to-end systems, the simplest approach is early fusion as it is often applied in hybrid systems by stacking the individual feature vectors \( o_t \) and \( u_t \) to a joint feature representation \( x_t = (o_t^\text{mag}, u_t^\text{phase})^\top \), with \( (.)^\top \) being the transposed. When using filterbank features, it has become a common technique to use convolutional neural networks (CNNs) \( \mathcal{C} \) in the input layer. In our Fusion- Early approach the additional feature stream is treated as an additional input channel, yielding an input tensor to the CNN block of size \( B \times (2 \times C) \times T \times F \) with \( B, C, T, F \) being the batch size, channel depth, feature sequence length, and feature dimension, respectively. After the input layer, the processing follows the standard transformer model architecture using a single attention-based encoder and a single decoder as in [6].

2.2. Middle Fusion

For the middle fusion approaches we use two individual stream encoders for each feature sequence as shown by the green boxes in Figure 1. Based on the previous output token \( c_{t-1} \in \mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(D)}\} \) and the entire feature sequences \( o_t^\text{mag}, u_t^\text{phase} \), where \( t \in \{1, \ldots, T\} \) and \( \ell \in \{1, \ldots, L\} \) are time instants of the input feature vector and output token sequences, respectively, the transformer outputs a vector \( P_x \) with output token probabilities for the current sequence time instant \( \ell \). Each of the stream encoders comprises a total of 12 identical encoder blocks, each consisting of the multi-head self-attention mechanism and position-wise fully connected layers as in [6]. The output of the last encoder block is then passed on to each of the in total 6 decoder blocks, which are detailed in Figure 2. We investigate two different strategies for middle fusion, both using two separate encoder-decoder multi-head attention blocks (shown in yellow) for each stream encoder. Fusion is then applied to the hidden entities \( h_t^\text{mag} \) and \( h_t^\text{phase} \) after the two encoder-decoder multi-head attention block outputs. The same setup is used for the Multi-Encoder Learning approaches (MEL-t-mag and MEL-t-phase) only during training, while only one encoder is active in inference. Dropout layers [23] are in dashed line boxes.

\[
h^\text{middle} = \alpha h^\text{mag}_t + (1 - \alpha) h^\text{phase}_t
\]

with \( \alpha \in [0,1] \) being a fusion weight to balance the influence of each of the encoder-decoder multi-head attention blocks. In addition, for the Fusion-t-Mid-WS approach, we tied ("-") the parameters of both involved encoder-decoder multi-head attention blocks.

The second variant dubbed Fusion-Mid-CC is the straightforward concatenation of both entities according to \( h^\text{middle} = (h^\text{mag}_t, h^\text{phase}_t)^\top \) as it has been used for audiovisual speech recognition in [25]. To still allow residual connections, in this case it becomes necessary to halve the dimension of both encoder-decoder multi-head attention block outputs to \( d/2 \) and add the residual from the self-attention after the concatenation, where the previous model dimension is restored.
2.3. Late Fusion
As late fusion we investigate the fusion of output token probability vectors \( P_{f}^{\text{mag}} \) and \( P_{f}^{\text{phase}} \) stemming from separately trained transformer networks for each feature stream \( \mathbf{s}_f \) and \( \mathbf{u}_f \). The final output token probability in the log domain for each time instant \( t \) is then computed as (Fusion-Late):

\[
\log P_{f}^{\text{late}} = \beta \log P_{f}^{\text{mag}} + (1 - \beta) \log P_{f}^{\text{phase}}
\]

with \( \beta \in [0, 1] \) being a posterior fusion weight and \( \log(\cdot) \) operating element-wise. One major advantage of the late fusion approach is that it uses independently trained models, and the balancing hyperparameter \( \beta \) can be easily set during inference time if one feature stream deteriorates.

2.4. Novel Multi-Encoder Learning (MEL)
In addition to the previous fusion paradigms, we employ a novel yet simple multi-encoder learning (MEL) approach to investigate if the additional information during training helps to increase robustness without using any additional parameters in inference. For this method, we train the middle fusion transformer model exactly as for the Fusion-t-Mid-WS approach using both encoders, but tie ("-t") all parameters of both multi-head attention blocks (shown as yellow blocks in Figure 1). During inference, however, only one of the encoders is active and the decoder uses one instance of the jointly trained multi-head attention. For the MEL-t-mag and MEL-t-phase approaches, only the magnitude or the phase encoder is active during inference, respectively, while the fusion weight \( \alpha = 0.9 \) during training is biased towards the inference encoder. Both models trained with the MEL method can also be subject to late fusion, dubbed MEL-t-Fusion-Late in Tables 1 and 3.

2.5. Language Model and Decoding
For all investigated approaches including non-fusion baselines Baseline-mag and Baseline-phase, we use beam-search decoding during inference and slightly deviate from the standard transformer architecture in [6] by using layer normalization before each attention or stack of fully connected layers according to the implementation in [33]. During decoding, the final output \( P_{f}^{\text{final}} \) of all approaches can optionally be computed as

\[
\log P_{f}^{\text{final}} = \log P_{f} + \lambda \log P_{f}^{\text{LM}},
\]

adding logarithmic character probabilities from the language model \( P_{f}^{\text{LM}} \), with the standard language model weight \( \lambda \) chosen according to the shallow integration technique [34]. For experiments on the Wall Street Journal task we report all results both without and with additional language model in Table 1.

3. Experimental Setup
3.1. Databases
We evaluate our fusion approaches on the 81-hour Wall Street Journal (WSJ) dataset [55] using the dev93 and eval92 splits to evaluate system performance in terms of word error rate WER = 1 - \( \frac{N - D - I - S}{N} \), as well as w.r.t. character error rate (CER), where the number of units \( N \), deletions \( D \), insertions \( I \), and substitutions \( S \) are calculated on character-level instead of on word-level as for the WER. To investigate our approaches also on a large-scale dataset, all experiments are repeated on Librispeech [8] using the 960 h training set along with the clean and other portions of the dev and test datasets. All used speech signals are sampled at 16 kHz and analyzed with a 25 ms window and a frame shift of 10 ms.

3.2. Acoustic Frontends
For the middle fusion approaches, each of the encoders receives a sequence of \( T \) feature vectors of dimension \( F = 83 \). As magnitude features \( \mathbf{s}_f \) we use standard 80-dimensional filterbank features extended with 3-dimensional pitch features extracted with the Kaldi toolkit [39]. For the phase features \( \mathbf{u}_f \) we follow the processing of [33], [38] and use the group-delay representation extracted from an all-pole model, and also apply an 80-dimensional mel-filterbank. For details on the processing, please refer to [38]. The convolutional neural networks (CNNs) at the input layers, shown as CNN blocks in Figure 1 consist of a total of four convolutional layers each using 3x3 filter kernels. The second and forth convolutional layer use a stride of 2 in both temporal and frequency direction thus compressing the input sequence length to \( T/4 \). We note that it might be beneficial to apply separate convolutions to the pitch features but follow [30] for comparability.

3.3. Acoustic and Language Model Configuration
As shown in Figure 1 the used transformer architecture for the acoustic model follows the standard architecture from [6] employing a total of 12 encoder blocks for each used encoder, while the decoder stacks 6 decoder blocks. For WSJ, the model dimension is set to \( D = 256 \) and multi-head attention blocks use 4 attention heads, while for Librispeech we use a larger model, where both values are doubled to \( D = 512 \) and 8 attention heads. Transformer models were trained using the Adam optimizer with label-smoothed cross-entropy loss [40]. We follow [37] for learning rate scheduling. For Librispeech experiments we additionally used spectral augmentation [29].

For language modeling in WSJ experiments, we apply a 3-layer LSTM network with a size of 1200 each, which is trained on word-level but yields character-level probabilities \( P_{f}^{\text{LM}} \) for a total of \( D = 52 \) characters, using the lookahead method proposed in [41]. As language model weight we follow [39] and choose \( \lambda = 0.9 \). For Librispeech we use SentencePiece for word tokenization with an output token dimension size of \( D = 5000 \) embeddings [42] and use a 4-layer LSTM as token-based LM with each layer having a size of 1024. The language model weight for Librispeech is set to \( \lambda = 0.4 \) following [39].

For the Fusion-Late approach, the posterior fusion weight \( \beta \) (only applied during inference) has been optimized on the respective development sets. For all middle fusion approaches with weighted sum we set the fusion weight \( \alpha = 0.9 \) without further tuning. The same value was used for the MEL approaches during training for the respective primary encoder (magnitude encoder for MEL-t-mag and phase encoder for MEL-t-phase).

All models were trained using the fairseq and espresso toolkits based on PyTorch [39, 43, 44]. WSJ models were trained on a single GTX1080Ti GPU, while Librispeech models used 4 Tesla P100 GPUs. All experiments use the same random seed.

4. Recognition Results and Discussion
Results of all approaches on the WSJ task are shown in Table 1. For the single-encoder approaches, we note that the Baseline-mag transformer performs slightly better than the Baseline-phase approach (about 1% absolute in terms of WER on the eval92 set with language model). Comparing results without and with language model (LM) for both Baseline approaches, we note that with LM the error rates are significantly reduced (especially for word errors), showing the effectiveness of the word-based lookahead LM [41].
Table 1: Transformer-based approaches on the WSJ task. Best results are bold, second best results are underlined.

| Approach          | Inference complexity | Without language model | With language model |
|-------------------|----------------------|------------------------|---------------------|
|                   |                      | dev93 | eval92 | dev93 | eval92 | dev93 | eval92 |
| Baseline-mag      |                      |       |        |       |        |       |        |
| Baseline-phase    |                      |       |        |       |        |       |        |
| Fusion-Early      |                      |       |        |       |        |       |        |
| Fusion-Mid-CC     |                      |       |        |       |        |       |        |
| Fusion-Mid-WS     |                      |       |        |       |        |       |        |
| Fusion-t-Mid-WS   |                      |       |        |       |        |       |        |
| Fusion-Late       |                      |       |        |       |        |       |        |
| MEL-t-mag         |                      |       |        |       |        |       |        |
| MEL-t-phase       |                      |       |        |       |        |       |        |
| MEL-t-Fusion-Late |                      |       |        |       |        |       |        |

Table 2: WER comparison of recent transformer-based end-to-end ASR approaches with language model on WSJ.

| Approach          | Inference complexity | WER   | CER   |
|-------------------|----------------------|-------|-------|
| Tsunoo et al. 2019 | 69.8M                | 6.90  | 4.20  |
| Karita et al. 2019 | 69.8M                | 6.80  | 4.40  |
| Moriya et al. 2020 | 69.8M                | 6.90  | 4.20  |
| Ours              |                      | 5.50  | 3.40  |

Table 3: Transformer-based approaches on Librispeech.

| Approach          | # of inference param. | WER   | test clean | other clean |
|-------------------|------------------------|-------|------------|-------------|
| Baseline-mag      | 69.8M                  | 6.90  | 4.05       | 8.14        |
| Baseline-phase    | 69.8M                  | 6.90  | 4.05       | 9.62        |
| Fusion-Early      | 69.8M                  | 7.82  | 3.78       | 8.14        |
| Fusion-Mid-CC     | 114.0M                 | 7.98  | 4.08       | 8.25        |
| Fusion-Mid-WS     | 115.6M                 | 8.02  | 3.96       | 8.34        |
| Fusion-t-Mid-WS   | 109.3M                 | 7.26  | 3.77       | 7.68        |
| Fusion-Late       | 139.6M                 | 6.91  | 3.63       | 7.34        |
| MEL-t-mag         | 69.8M                  | 7.37  | 3.87       | 7.90        |
| MEL-t-phase       | 69.8M                  | 6.86  | 4.05       | 9.03        |
| MEL-t-Fusion-Late | 139.6M                 | 6.63  | 3.34       | 7.15        |

5. Conclusion

In this contribution we introduced several fusion mechanisms to transformer-based end-to-end speech recognition. In addition, we apply a novel multi-encoder learning method (MEL), that uses the additional information from a second encoder only during training, while just a single encoder is used during inference. Compared to standard transformer approaches our novel MEL achieves a consistent WER reduction on all investigated tasks at the same runtime and number of parameters. By performing additional fusion, we achieve a WER reduction of 19% relative on the Wall Street Journal task compared to state of the art, thereby defining a new benchmark for transformer-based ASR on that task.

6. Acknowledgements

The research leading to these results has received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for project number 414091002, as well as from the Bundesministerium für Wirtschaft und Energie (BMWi) under funding code 01MK20011T.
