Optimizing Speech Multi-View Feature Fusion through Conditional Computation

Weiqiao Shan[1], Yuhao Zhang[2*], Yuchen Han[1], Bei Li[3], Xiaofeng Zhao[4], Yuang Li[4], Min Zhang[4], Hao Yang[4], Tong Xiao[1,5†], Jingbo Zhu[1,5]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]The Chinese University of Hong Kong, Shenzhen, China
[3]Meituan, Beijing, China
[4]Huawei Translation Services Center, Beijing, China
[5]NiuTrans Research, Shenyang, China

***Abstract*—Recent advancements have highlighted the efficacy of self-supervised learning (SSL) features in various speech-related tasks, providing lightweight and versatile multi-view speech representations. However, our study reveals that while SSL features expedite model convergence, they conflict with traditional spectral features like FBanks in terms of update directions. In response, we propose a novel generalized feature fusion framework grounded in conditional computation, featuring a gradient-sensitive gating network and a multi-stage dropout strategy. This framework mitigates feature conflicts and bolsters model robustness to multi-view input features. By integrating SSL and spectral features, our approach accelerates convergence and maintains performance on par with spectral models across multiple speech translation tasks on the MUSTC dataset.**

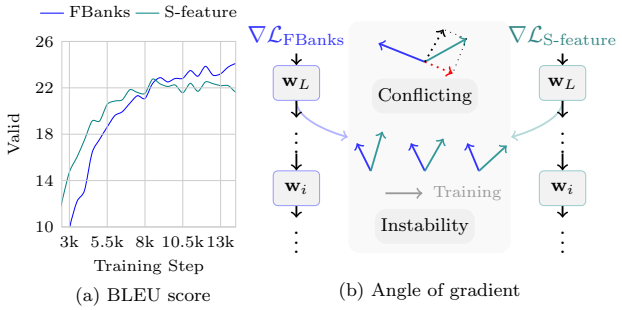
***Index Terms*—Speech Translation, Conditional Computation, Feature Fusion**

I. Introduction

Conventional speech processing methods rely on spectral features (e.g., MFCC, FBanks) [1]–[3], while recently, features learned through self-supervised learning (SSL) methods have garnered the attention of researchers [4]–[6]. The SSL-based features (S-feature) alleviate the variability of signals in a fixed representation space using the vector quantization (VQ) method [7]. This type of feature is easy to learn and operate, and offers multi-view representations for speech processing tasks [8], [9]. Furthermore, the S-feature demonstrates a strong capacity for integrating contextual information, making it particularly effective for downstream tasks [7], [10], even beneficial for building speech large language models [11].

Although the S-feature incorporates more in-context information learned from the pre-training stage, it suffers from information loss during the VQ process. This makes it difficult to achieve the strongest performance compared to spectral features in complicated tasks. Fig. 1(a) shows an example in the Speech Translation (ST) task. The S-feature demonstrates rapid convergence but fails to outperform the spectral feature. Inspired by the work that fuses multi-view features to enhance the performance of downstream tasks [12]–[14], we naturally raised the question: can we leverage the strengths of the two views of feature to improve complex speech-to-text tasks?

Though multi-view feature fusion has been achieved for low-resource cases and emotion recognition in separate studies [15], [16], there remains a lack of discussion on how to accomplish this under more general conditions. Our primary analysis reveals a significant divergence in the update directions between FBanks and S-feature as shown in Fig. 1(b). This gradient conflict explains why simple fusion methods fail to effectively integrate multi-view features (Tab. III). To overcome this limitation, we introduce conditional computation which dynamically activates parts of the model parameters based on the inputs through a gating network [17]. This dynamic activation property allows the model to better adapt to the input characteristics, similar to the widely discussed early exit models [18], [19] and the mixture of experts used in multitask learning [20], [21].



(a) BLEU score (b) Angle of gradient

Fig. 1. The BLEU scores of the model using only FBanks or S-feature (a). When we freeze all the model parameters during training and sequentially input the two features, we observe that up to 32% gradients generated by the two features contain conflicting components, and the angle between the two gradients increases as training progresses (b).

Specifically, we propose a gradient-sensitive gating network (GSGN) that implements conditional computation for the gradient of parameters, enhancing the adaptability of our approach by dynamically computing the correlation between heterogeneous features and adjusting the weights of different feature branches to achieve optimal fusion. We carry on the experiences in MuST-C for three languages (En-De, En-Es, and En-Fr) in ST tasks. Our proposed method effectively resolves the conflict between the two features. For models trained from scratch, our method can provide comparable performance and average 1.24 times training speedup compared to the model with the FBanks feature. As opposed to the pre-trained ST model, our approach is also able to provide a corresponding training acceleration effect when the S-feature is sufficiently effective and guarantees the performance of the model.

II. Method

A. Architecture

We adopt an end-to-end ST system following previous work [22], [23]. The system consists of an acoustic encoder (A-enc), a textual

This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

*Equal contribution.
†Corresponding author

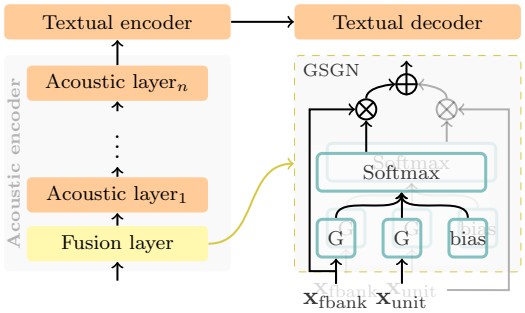


Fig. 2. ST model architecture.

encoder (T-enc), and a decoder (dec). The forward process of the network can be denoted as follows:

$$\begin{aligned}\mathbf{h}_{\text{A}} &= \text{A-enc}(\mathbf{x})\\ \mathbf{h}_{\text{T}} &= \text{T-enc}(\mathbf{h}_{\text{A}})\\ \mathbf{h}_{\text{out}} &= \text{dec}(\mathbf{h}_{\text{T}})\end{aligned} \tag{1}$$

During the training phase, the model generates the final prediction based on the decoder's output and obtains the final loss via the subsequent cross-entropy function:

$$\mathcal{L} = \text{CrossEntropy}(\mathbf{w}_{\text{out}}\mathbf{h}_{\text{out}}, \hat{\mathbf{y}}) \tag{2}$$

where $\hat{\mathbf{y}}$ is the target text. Conventional ST tasks typically use the FBanks feature as input, denoted as $\mathbf{x} = \mathbf{x}_{\text{fbank}}$. In this paper, we introduce a representation with general and contextual information as a multi-view speech representation, the S-feature feature, which is represented as $\mathbf{x}_{\text{unit}}$, we incorporate new modules that dynamically fuse the two features through an additional fusion layer as shown in Fig. 2.

B. Gradient-sensitive Gating Network for Conflicting Gradient

We first present the specific formulation of our gradient-sensitive gating network. It accepts two inputs with embedding dimension size D, $\mathbf{x}_{\text{fbank}}, \mathbf{x}_{\text{unit}} \in \mathbb{R}^{1\times D}$. Each input undergoes a linear mapping, resulting in the gating vector $\mathbf{g} \in \mathbb{R}^{1\times D}$ as follows:

$$\mathbf{g}_{[\cdot]} = \text{Sigmoid}(\text{Linear}^1_{[\cdot]}(\mathbf{x}_{\text{fbank}}) + \text{Linear}^2_{[\cdot]}(\mathbf{x}_{\text{unit}}) + \text{bias}_{[\cdot]}) \tag{3}$$

For input $\mathbf{x}_{\text{fbank}}$ and $\mathbf{x}_{\text{unit}}$, we compute the gating vector $\mathbf{g}_{\text{fbank}}$ and $\mathbf{g}_{\text{unit}}$ as the weight separately. Finally we fuse the two features by Hadamard product:

$$\mathbf{x}_{\text{fusion}} = \mathbf{g}_{\text{fbank}} \circ \mathbf{x}_{\text{fbank}} + \mathbf{g}_{\text{unit}} \circ \mathbf{x}_{\text{unit}} \tag{4}$$

Given that the network structure in use is a residual network [24], we express the transfer of each layer using the following equation:

$$\mathbf{h}_{i+1} = \underbrace{F(\mathbf{h}_i)}_{\text{residual}} + \underbrace{\mathbf{h}_i}_{\text{shortcut}} \approx \underbrace{\mathbf{w}_i\mathbf{h}_i + \mathbf{b}_i}_{\text{residual}} + \underbrace{\mathbf{h}_i}_{\text{shortcut}} \tag{5}$$

where $\mathbf{h}_i$ is the hidden state of the i-th layer and $F(\cdot)$ denotes the function of this layer. For simplicity, we assume a linear model for $F(\cdot)$, omitting nonlinear operations and layer normalization.

Taking the input $\mathbf{x} = \mathbf{x}_{\text{fusion}}$ as an example, the gradient for the parameter $\mathbf{w}_i$ in layer i can be expressed as:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_i} = \underbrace{(\mathbf{g}_{\text{fbank}} \circ \mathbf{x}_{\text{fbank}} + \mathbf{g}_{\text{unit}} \circ \mathbf{x}_{\text{unit}})^{\text{T}}}_{\text{Input-dependent}} \times \underbrace{\sum_{j=0, j\neq i}^{n} (\mathbf{w}_j^{\text{T}} + \mathbf{1}) \frac{\partial\mathcal{L}}{\partial\mathbf{h}_{\text{out}}}}_{\text{Input-independent}} \tag{6}$$

We can derive the gradient $\frac{\partial\mathcal{L}_{\text{fbank}}}{\partial\mathbf{w}_i}$ when we only use the FBanks feature $\mathbf{x}_{\text{fbank}}$ to replace the input-dependent term in Eq. (6). Similarly, we use $\frac{\partial\mathcal{L}_{\text{unit}}}{\partial\mathbf{w}_i}$ to denotes the gradient when $\mathbf{x}_{\text{unit}}$ is the only input. Given that the Hadamard product scales the elements directly, Eq. (6) can be transformed into a weighted summation of two gradients when different features are input separately under frozen parameters.

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_i} = \mathbf{g}_{\text{fbank}} \circ \frac{\partial\mathcal{L}_{\text{fbank}}}{\partial\mathbf{w}_i} + \mathbf{g}_{\text{unit}} \circ \frac{\partial\mathcal{L}_{\text{unit}}}{\partial\mathbf{w}_i} \tag{7}$$

As illustrated in Fig. 1, the two gradients $\frac{\partial\mathcal{L}_{\text{fbank}}}{\partial w_i}$ and $\frac{\partial\mathcal{L}_{\text{unit}}}{\partial w_i}$ contain conflicting components when $\cos(\theta) < 0$. Such conflicts in gradients are commonly observed in multitask learning and can lead to the seesaw phenomenon [25]. To mitigate this issue, an efficient strategy is to eliminate the components of one gradient $\overrightarrow{b}$, that conflict with the gradient of the main feature $\overrightarrow{a}$ [26].

$$\text{Deconflict}(\overrightarrow{a}, \overrightarrow{b}) = \overrightarrow{b} - \frac{\|\overrightarrow{b}\|\cos(\theta)}{\|\overrightarrow{a}\|}\overrightarrow{a} \tag{8}$$

Therefore the gradient of the model should be corrected to the summation of the two gradients without conflicting components.

$$\begin{cases}\overrightarrow{a} + \overrightarrow{b}, & \text{if } \cos(\theta) >= 0,\\ \overrightarrow{a} + \text{Deconflict}(\overrightarrow{a}, \overrightarrow{b}) & \text{else}\end{cases} \tag{9}$$

Based on the Eq. (8), the final gradient of the model is as follows:

$$\begin{cases}\mathbf{1}\circ\overrightarrow{a} + \mathbf{1}\circ\overrightarrow{b}, & \text{if } \cos(\theta) >= 0,\\ (\mathbf{1} - \frac{\|\overrightarrow{b}\|\cos(\theta)}{\|\overrightarrow{a}\|})\circ\overrightarrow{a} + \mathbf{1}\circ\overrightarrow{b} & \text{else}\end{cases} \tag{10}$$

Building on the previous description, we observe that when we set $\overrightarrow{a} = \frac{\partial\mathcal{L}_{\text{fbank}}}{\partial\mathbf{w}_i}$ and $\overrightarrow{b} = \frac{\partial\mathcal{L}_{\text{unit}}}{\partial\mathbf{w}_i}$, we only need to adjust $\mathbf{g}_{[\cdot]}$ in Eq. (7) to satisfy Eq. (10), and in turn eliminate the conflicting components of the gradients produced by the two features. Consequently, we can derive the ideal gradient by constraining $\mathbf{g}_{\text{unit}} = \mathbf{1}$ and introducing an additional loss term for $\mathbf{g}_{\text{fbank}}$.

$$\mathcal{L}_{\text{gate}} = \begin{cases}\text{MSE}(\mathbf{g}_{\text{fbank}}, \mathbf{1}) & \text{if } \cos(\theta) >= 0,\\ \text{MSE}(\mathbf{g}_{\text{fbank}}, (\mathbf{1} - \frac{\|\frac{\partial\mathcal{L}_{\text{unit}}}{\partial\mathbf{w}_i}\|\cos(\theta)}{\|\frac{\partial\mathcal{L}_{\text{fbank}}}{\partial\mathbf{w}_i}\|})) & \text{else}\end{cases} \tag{11}$$

So the final loss of the model is:

$$\mathcal{L}_{\text{final}} = \mathcal{L} + \mathcal{L}_{\text{gate}} \tag{12}$$

C. Multi-stage dropout for Instability Gradient

For the instability gradient shown in Fig. 1, the $\cos(\theta)$ between the two gradients from each feature is not constant during training. The variation in gradient directions highlights the inconsistency in the roles of different features during the training process. Initially, the roles of these features may be highly similar. However, these updating directions change significantly during the training process, suggesting that the branches adapt to focus on different aspects of the data at various stages.

To enhance the model's robustness to each input feature and ensure that all features can be fully utilized, we dynamically sample from three features during the training stage and input them into the next layer. Specifically, we sample the FBanks, S-feature, and fusion features with two fixed threshold $\delta_{\text{fbank}}, \delta_{\text{unit}}$ and a random probability $p \in (0, 1)$.

$$\mathbf{x} = \begin{cases}\mathbf{x}_{\text{fbank}}, & \text{when} \quad p < \delta_{\text{fbank}}\\ \mathbf{x}_{\text{unit}}, & \text{when} \quad \delta_{\text{fbank}} <= p < \delta_{\text{fbank}} + \delta_{\text{unit}}\\ \mathbf{x}_{\text{fusion}}, & \text{when} \quad \delta_{\text{fbank}} + \delta_{\text{unit}} <= p\end{cases} \tag{13}$$

TABLE I
MAIN RESULT OF OUR METHOD.

Model	En-De		En-Fr		En-Es	
	Test	Epoch	Test	Epoch	Test	Epoch
S2T(FBanks-only)	25.08	40	36.22	35	30.04	37
S2T(S-feature-only)	23.33	27	33.00	25	26.52	23
S2T-GSGN	24.18	28	33.91	25	28.61	24
S2T-GSGN+Drop	25.26 (+0.18)	30 (×1.33)	36.24 (+0.02)	30 (×1.17)	30.24 (+0.20)	30 (×1.23)

III. EXPERIMENTS

A. Datasets

We mainly experimented on the MuST-C dataset (MuSTC) including three benchmarks [23], [27], English-German (En-De), English-French (En-Fr) and English-Spanish (En-Es), and we report results on the test sets for all models. For the experiment based on the pre-trained model, we obtain the pre-trained ASR model on the LibriSpeech [28] dataset to initialize the acoustic encoder. For pre-training textual encoder and decoder, pre-training data include WMT16, WMT14, and WMT13 for En-De, En-Fr, and En-Es three tasks respectively. The detailed training setup throughout the pre-training phase follows [29]. To obtain the S-feature, we use the pre-trained and fine-tuned ASR Hubert models [7], and setting 500 for k-means cluster center based on the release model[1].

B. Experimental Settings

Based on the Transformer model [30], our model consists of an acoustic encoder with 12 layers, and a textual encoder and decoder with 6 layers. We set the model hidden size to 512 and the feedforward size to 2048, with 8 head attention. We applied dropout with a value of 0.1 to attention weights and 0.2 to residual connections, respectively. We also used label smoothing of 0.1 to handle overfitting. The Adam (β_1 = 0.9, β_2 = 0.98) optimizer with a warmup step of 4K was adopted. Additionally, we use the early stopping strategy set the patience = 10.

For GSGN, we follow the setting in Eq. 4 to compute a weight matrix $\mathbf{g}_{[\cdot]}$. The matrix $\mathbf{g}_{[\cdot]}$ has the same dimension with each input feature, where each $\mathbf{g}_{[\cdot]}$ is given by $\text{Linear}_{[\cdot]} : \mathbb{R}^{1\times D} \rightarrow \mathbb{R}^{1\times D}$, and we set D to 512 following the model hidden size.

For multi-stage dropout, we empirically divide the training process into three stages, when the current epoch $<$ 10, we set the $\delta_{\text{fbank}} = 0.3, \delta_{\text{unit}} = 0$ for better performance of gradient descent. During the next stage, when $10 \leq$ epoch < 25 we increase the $\delta_{\text{fbank}} = 0.5, \delta_{\text{unit}} = 0.3$ to ensure the model fully leverages the different features. Finally, for $25 \leq$ epoch and we revert the thresholds to $\delta_{\text{fbank}} = 0.3, \delta_{\text{unit}} = 0$, allowing the fusion branch to achieve better performance during the inference stage.

During the inference stage, we average the best 10 checkpoints on the valid set for evaluation with beam size = 5 for beam search, and length penalty = 1.0, we provide the last epoch at the average 10 checkpoints to indicate the convergence speed. We do a down-sample for the S-feature for the same length as the FBanks feature, and we use the same setting as the third stage in multi-stage dropout to sample from three branches. We evaluate translation quality with sacreBLEU [31].

[1]https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960_L9_km500.bin

TABLE II
RANDOM NOISE EXPERIMENT.

Noise	Model	Test	Epoch
Sum	S2T(FBanks-only)	25.37	38
Replace	S2T-GSGN	25.19	37
Replace	S2T-GSGN+Drop	25.3	39

C. Results

Tab. I shows the performance of models based on GSGN and GSGN+Drop. The GSGN+Drop method achieves better results and faster convergence compared to the baseline model. We find that the GSGN method effectively improves performance while maintaining a comparable convergence rate to the model with the S-feature. Furthermore, GSGN+Drop enhances the robustness of the model to the two input features while maintaining an acceptable training speedup[2].

IV. ANALYSIS

A. Effectiveness of S-feature

To demonstrate the effectiveness of the S-feature, we incorporated random noise into the model in the EN-DE direction. The random noise sampling from a uniform distribution has the same maximum and minimum values as the S-feature across the entire training set, the results are shown in Tab. II. We employed two noise addition methods 1)Sum: adding noise directly to the current input, and 2)Replace: replacing the S-feature with noise. While the model with random noise achieves better results, it requires more training epochs. This suggests that even adding noise directly to the baseline model with FBanks can enhance performance by improving robustness and mitigating overfitting. When replacing the S-feature with random noise in the S2T-GSGN model, we observed better performance but slower convergence, indicating that the S-feature indeed provides a fast convergence capability for the ST model, rather than behaving like inaccurate noise. Furthermore, in the S2T-GSGN+Drop model, the results suggest that the multi-stage dropout method enhances the model's ability to leverage different features, rather than solely serving as a robust training method.

B. Effectiveness of GSGN

We also investigated the effectiveness of GSGN by comparing a simple concatenation-based gating network on En-De direction based on the model trained from scratch. The new gating concat two features $\mathbf{x}_{\text{concat}} = [\mathbf{x}_{\text{fbank}}; \mathbf{x}_{\text{unit}}] \in \mathbb{R}^{1\times 2D}$ and maps them back to original dimensions by $\text{Linear}(\mathbf{x}_{\text{concat}}) : \mathbb{R}^{1\times 2D} \rightarrow \mathbb{R}^{1\times D}$. We find that GSGN makes more efficient use of the S-feature and achieves faster and better convergence as shown in Tab. III. This suggests that

[2]Our code is available at https://github.com/shanweiqiao/GSGN.git

TABLE III
CONCATENATION-BASED GATING NETWORK VS GSGN.

En-De	Test	Epoch
S2T-Concat	23.8	32
S2T-GSGN	24.18	28

TABLE IV
EXPERIMENTS OF OUR METHOD ON THE PRE-TRAINED MODELS.

En-De	Test	Epoch
S2T(FBanks-only)	27.98	33
S2T(S-feature-only)	25.60	25
S2T-GSGN	27.61	25
S2T-GSGN+Drop	28.19 (+0.21)	27 ($\times$1.22)

the simple concatenation-based gating network fails to resolve the conflicting gradient problem between two features, thereby limiting its ability to fuse them effectively.

C. Effectiveness under the pre-trained model

We conducted the same experiment on the ST model initialized with a pre-trained model, as shown in Tab. IV. The results indicate that the GSGN is more effective on the pre-trained models than on the models trained from scratch. We find that this difference is due to the varying frequencies of gradient conflicts and instability in the trained from scratch and pre-trained models (Fig. 3). In pre-trained models, gradient conflict is particularly common at the beginning of training, and GSGN effectively mitigates this issue, leading to improved results. In contrast, gradient instability is more prominent in models trained from scratch, making the multi-stage dropout method more beneficial.

D. Weight Analysis for GSGN

Additionally, we analyzed the weights assigned by the GSGN to the FBanks features in the S2T-GSGN+Drop model. As shown in Fig. 4, for the model trained from scratch, we observed that the GSGN pays more attention to the S-feature in the initial phase of the training to leverage the rapid convergence properties. During training, GSGN gradually fuses more FBanks to achieve a more stable and effective descent process.

For the pre-trained model, we observe that GSGN assigns a higher weight to $\mathbf{g}_{\text{fbank}}$. This is because, when gradient conflicts occur, the $\cos(\theta)$ is less than 0, and the term $(\mathbf{1} - \frac{\|\vec{b}\|\cos(\theta)}{\|\vec{a}\|})$ in Eq. 10, or equivalently the mean of $\mathbf{g}_{\text{fbank}}$ in Eq. 4 would be greater than 1. We find that 89.53% of the elements in $\mathbf{g}_{\text{fbank}}$ exceed 1 in the pre-trained model. This indicates that the conflict gradient in the pre-trained model is severe, which aligns with our observation in Fig. 3 (right).

It is important to emphasize that both instability gradient and conflict gradient co-exist in models trained from scratch and pre-trained models. Although the conflict gradient is particularly common in the pre-trained models, there are still 10.46% of the elements in $\mathbf{g}_{\text{fbank}}$ that are less than 1. Conversely, in models trained from scratch, about 8.63% of the elements in $\mathbf{g}_{\text{fbank}}$ exceed 1. This demonstrates that GSGN, in combination with the multi-stage dropout method, effectively addresses the gradient-related issues in both cases. After correctly handling the gradient problem, we find that our method achieves faster convergence speedup compared to the model using only FBanks, and superior performance compared to the model using

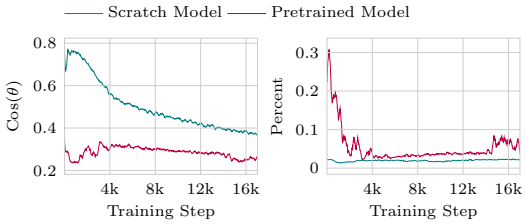
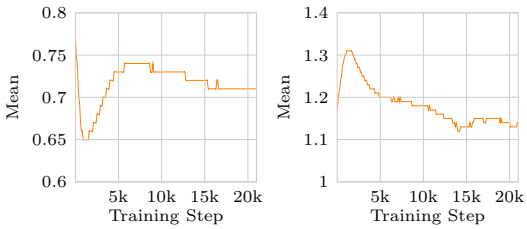


Fig. 3. The $\cos(\theta)$ between two gradients generated by each feature. We find **Instability**(left): The $\cos(\theta)$ becomes smaller and smaller with training for $\cos(\theta) > 0$, and **Conflicting**(right): Percent of conflicting gradients($\cos(\theta) < 0$) among all gradients in an input batch.



(a) Scratch Model. (b) Pretrained Model.

Fig. 4. The mean value of $\mathbf{g}_{\text{fbank}}$ in the model training from scratch (left) and the pre-trained model (right).

only the S-feature. Our fusion method successfully leverages the strengths of both features, as shown in Fig. 5.

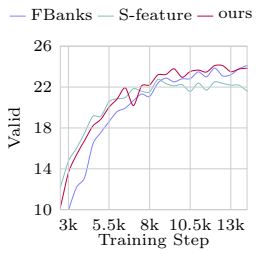


Fig. 5. The final result of our method.

V. CONCLUSION AND FUTURE WORK

Recent studies have demonstrated the effectiveness of S-features across various speech-related tasks, offering lightweight and general-purpose multi-view speech representations. However, we observed that while S-features exhibit rapid convergence, they fail to outperform FBanks features due to conflicts in their update directions. To address this, we propose a conditional computation method with a specialized gradient-sensitive gating network, which dynamically computes the correlation between heterogeneous features and adjusts the weights of different feature branches. Our method effectively mitigates conflicts between the two features while exploiting their complementary aspects. As a result, we achieve comparable performance and average 1.24 times training speedup compared to models using only FBanks features, for both models trained from scratch and pre-trained models in the MuST-C En-De, En-Es, and En-Fr speech translation tasks. In the future, we aim to explore more effective fusion approaches by incorporating varied unit representations from more pre-trained models, as well as exploiting the pre-trained models directly.

REFERENCES

[1] Lawrence R Rabiner, Ronald W Schafer, et al., "Introduction to digital speech processing," *Foundations and Trends® in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.

[2] Pierre Thodoroff, Joelle Pineau, and Andrew Lim, "Learning robust features using deep learning for automatic seizure detection," in *Machine learning for healthcare conference*. PMLR, 2016, pp. 178–190.

[3] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[4] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[5] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[6] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[7] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.

[8] Jiatong Shi, Yun Tang, Hirofumi Inaguma, Hongyu Gong, Juan Pino, and Shinji Watanabe, "Exploration on HuBERT with Multiple Resolutions," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2023, vol. 2023, pp. 3287–3291.

[9] Jiatong Shi, Hirofumi Inaguma, Xutai Ma, Ilia Kulikov, and Anna Sun, "Multi-resolution HuBERT: Multi-resolution speech self-supervised learning with masked unit prediction," *arXiv preprint arXiv:2310.02720*, 2023.

[10] Heeseung Kim, Sungwon Kim, Jiheum Yeom, and Sungroh Yoon, "Unitspeech: Speaker-adaptive speech synthesis with untranscribed data," *arXiv preprint arXiv:2306.16083*, 2023.

[11] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," *arXiv preprint arXiv:2305.11000*, 2023.

[12] Zhibin Wan, Changqing Zhang, Pengfei Zhu, and Qinghua Hu, "Multi-view information-bottleneck representation learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, pp. 10085–10092.

[13] Xiaoqiang Yan, Shizhe Hu, Yiqiao Mao, Yangdong Ye, and Hui Yu, "Deep multi-view learning methods: A review," *Neurocomputing*, vol. 448, pp. 106–129, 2021.

[14] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola, "The platonic representation hypothesis," *arXiv preprint arXiv:2405.07987*, 2024.

[15] Dan Berrebbi, Jiatong Shi, Brian Yan, Osbel López-Francisco, Jonathan Amith, and Shinji Watanabe, "Combining Spectral and Self-Supervised Features for Low Resource Speech Recognition and Translation," *Interspeech 2022*, 2022.

[16] Jonghwan Hyeon, Yung-Hwan Oh, Young-Jun Lee, and Ho-Jin Choi, "Improving speech emotion recognition by fusing self-supervised learning and spectral features via mixture of experts," *Data & Knowledge Engineering*, vol. 150, pp. 102262, 2024.

[17] Yoshua Bengio, "Deep learning of representations: Looking forward," in *International conference on statistical language and speech processing*. Springer, 2013, pp. 1–37.

[18] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli, "Depth-adaptive transformer," *arXiv preprint arXiv:1910.10073*, 2019.

[19] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al., "Layer skip: Enabling early exit inference and self-speculative decoding," *arXiv preprint arXiv:2404.16710*, 2024.

[20] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1930–1939.

[21] Shashank Gupta, Subhabrata Mukherjee, Krishan Subudhi, Eduardo Gonzalez, Damien Jose, Ahmed H Awadallah, and Jianfeng Gao, "Sparsely activated mixture-of-experts are robust multi-task learners," *arXiv preprint arXiv:2204.07689*, 2022.

[22] Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Qi Ju, Tong Xiao, Jingbo Zhu, et al., "Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders," *arXiv preprint arXiv:2105.05752*, 2021.

[23] Yuhao Zhang, Kaiqi Kou, Bei Li, Chen Xu, Chunliang Zhang, Tong Xiao, and Jingbo Zhu, "Soft Alignment of Modality Space for End-to-End Speech Translation," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11041–11045.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong, "Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations," in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 269–278.

[26] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[27] Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi, "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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 2012–2017.

[28] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[29] Yuchen Han, Chen Xu, Tong Xiao, and Jingbo Zhu, "Modality adaption or regularization? a case study on end-to-end speech translation," *arXiv preprint arXiv:2306.07650*, 2023.

[30] Ashish Vaswani, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[31] Matt Post, "A call for clarity in reporting BLEU scores," *arXiv preprint arXiv:1804.08771*, 2018.