Revisiting Interpolation Augmentation for Speech-to-Text Generation

Chen Xu[1], **Jie Wang**[2], **Xiaoqian Liu**[2], **Qianqian Dong**[3], **Chunliang Zhang**[2,4],
Tong Xiao[2,4], **Jingbo Zhu**[2,4] **Dapeng Man**[1*] and **Wu Yang**[1]
[1]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[2]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[3]ByteDance
[4]NiuTrans Research, Shenyang, China
{chen.xu, mandapeng, yangwu}@hrbeu.edu.cn
{wangjienlp, liuxiaoqian0319}@outlook.com, dongqianqian@bytedance.com
{zhangchunliang, xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Speech-to-text (S2T) generation systems frequently face challenges in low-resource scenarios, primarily due to the lack of extensive labeled datasets. One emerging solution is constructing virtual training samples by interpolating inputs and labels, which has notably enhanced system generalization in other domains. Despite its potential, this technique's application in S2T tasks has remained under-explored. In this paper, we delve into the utility of interpolation augmentation, guided by several pivotal questions. Our findings reveal that employing an appropriate strategy in interpolation augmentation significantly enhances performance across diverse tasks, architectures, and data scales, offering a promising avenue for more robust S2T systems in resource-constrained settings.[1]

1 Introduction

Recently, neural network-based end-to-end systems have achieved impressive improvements and become the de facto modeling method for speech-to-text (S2T) generation tasks, such as automatic speech recognition (ASR) (Karita et al., 2019) and automatic speech translation (AST) (Xu et al., 2023b). These deep learning models typically comprise millions or even billions of parameters and require vast amounts of training data to achieve state-of-the-art performance (Zhang et al., 2022). For example, leading ASR models demand thousands of hours of training data (Lu et al., 2020). However, the labeling of such extensive datasets leads to significant costs, and models trained on limited data are prone to overfitting, resulting in suboptimal generalization to unseen samples (Ying, 2019).

To enhance generalization capabilities, data augmentation has become a key strategy (Shorten and Khoshgoftaar, 2019). Existing approaches in S2T can be broadly classified into two categories: online and offline augmentation. Online methods, such as SpecAugment (Park et al., 2019), enhance regularization by transforming the input representation during training. By introducing random noise into input features, these techniques have become standard in S2T tasks. Offline methods, on the other hand, boost data diversity by creating large amounts of pseudo-data through original audio distortion (Ko et al., 2015) or synthesis (Rosenberg et al., 2019). Though effective, these offline techniques are separate from the training process, often requiring additional steps and computational resources. This creates a demand for more efficient solutions.

We resort to interpolation augmentation (IPA), also known as Mixup, a notable method first introduced in image classification (Zhang et al., 2017). IPA mitigates overfitting by constructing virtual samples through linear interpolation of both input features and labels from two randomly selected samples. This approach has achieved impressive success across diverse domains, including speech processing (Medennikov et al., 2018; Lam et al., 2020; Meng et al., 2021; Kang et al., 2023), natural language processing (Guo et al., 2019; Sun et al., 2020; Xie et al., 2023), and computer vision (Verma et al., 2019; Wang et al., 2023).

In the specialized field of speech processing, preliminary studies have explored IPA in speech separation (Lam et al., 2020; Alex et al., 2023) and classification tasks (Snyder et al., 2017; Liu et al.,

[*] Corresponding author.

[1] The source code is available at https://github.com/xuchennlp/S2T.

Findings of the Association for Computational Linguistics ACL 2024, pages 9488–9499
August 11-16, 2024 ©2024 Association for Computational Linguistics

2023). However, its application in S2T tasks remains limited and largely unexplored (Medennikov et al., 2018; Meng et al., 2021; Cheng et al., 2022; Zhou et al., 2023). The existing work has not yet established clear guidelines on when and how IPA can be optimally leveraged in S2T tasks, leaving a substantial gap in our understanding and application of this promising technique.

In this paper, we examine this question more closely, conducting a series of experiments to answer the following questions:

Q1 What is the **appropriate interpolation strategy**, and what distinctions arise between interpolating speech features and text embeddings? (§3)

Q2 How can IPA create an effective **combination with existing augmentation techniques**, such as the well-established method SpecAugment? (§4)

Q3 Are there **specific issues** in applying IPA to S2T tasks, and how can they be addressed? (§5)

Q4 How does IPA perform **across various scenarios**? (§6)

By probing these questions, we develop an effective IPA method that achieves consistent improvements across two S2T tasks (including ASR and AST), various architectures (including encoder-decoder and encoder-CTC), and diverse data scales (ranging from LibriSpeech 10h to 960h).

2 Experimental Settings

Data augmentation methods typically demonstrate greater potential in low-resource scenarios. In light of this, we conduct analyses using the LibriSpeech 100h ASR dataset and subsequently apply our findings to various scenarios. We report results mainly on the test-clean and test-other sets. The average word error rate (WER) is calculated on the concatenation of all four subsets.

Various existing data augmentation techniques, such as SpecAugment and speed perturbation, have achieved excellent results. SpecAugment (Park et al., 2019), the most widely employed method in S2T tasks, introduces random noise to the input features through time warping, frequency masking, and time masking. Speed perturbation (Ko et al., 2015), on the other hand, commonly expands the dataset by generating three variations of raw audio with speed factors of 0.9, 1.0, and 1.1, facilitating its integration. In our work, the goal of IPA is to not only lead to isolate improvements but to also work orthogonally with these methods. Therefore, we first examine scenarios without other augmentations and then explore the effects of their combination.

In the field of S2T, common architectures encompass both encoder-decoder (**Enc-Dec**) and encoder-CTC (**Enc-CTC**) designs. The Enc-Dec model consists of an encoder with 12 Conformer layers and a decoder with 6 Transformer layers, each containing 256 hidden units, 4 attention heads, and 2048 feed-forward sizes. Connectionist Temporal Classification (CTC, Graves et al., 2006) multi-task learning is applied on top of the encoder, introducing an additional loss with a weight of 0.3. The Enc-CTC model can be viewed as a variant of the Enc-Dec model, containing only an 18-layer Conformer encoder for comparable parameters of about 30M. It predicts the text purely through CTC, where the weight of the CTC loss is 1. We initially investigate the effects of IPA on the Enc-Dec model before extending the method to the Enc-CTC model. More details about the datasets and model settings are described in Appendix A.

3 Q1: Choice of Interpolation Strategy

In this section, we begin with an overview of the basic implementation of IPA. Subsequently, we investigate the appropriate interpolation strategy tailored specifically for the field of S2T generation.

3.1 Definition of IPA

IPA, commonly known as Mixup (Zhang et al., 2017), constructs virtual samples in a vicinal distribution by linearly interpolating both the inputs and labels of two randomly selected samples, thereby enhancing the model's generalization capability. Considering two samples (x_i, y_i) and (x_j, y_j), where x denotes the input features and y represents the corresponding label. IPA assembles the new sample as follows:

$$x_m = \lambda \cdot x_i + (1 - \lambda) \cdot x_j \tag{1}$$

$$y_m = \lambda \cdot y_i + (1 - \lambda) \cdot y_j \tag{2}$$

where $\lambda \in [0, 1]$ is a weighting factor drawn from a Beta distribution $\lambda \sim \text{Beta}(\alpha, \alpha)$.

A value of α approaching 0 implies that the generated samples closely resemble either (x_i, y_i) or

(x_j, y_j), while a value of α approaching $+\infty$ leads to a more balanced interpolation between the two. In practical applications, IPA randomly replaces a subset of samples with the interpolated versions in each mini-batch, while leaving the remaining samples untouched. The selection ratio γ is typically set to 1, indicating the model is trained completely on the interpolated samples. Both α and γ serve as essential hyper-parameters, and finding their optimal values often requires careful empirical exploration.

3.2 IPA Strategy in S2T

Building upon the aforementioned framework, we extend our investigation to the application of IPA within the domain of S2T generation, focusing specifically on ASR and AST tasks.

Let a training sample be denoted as (s, x, y), where s denotes the speech features, x denotes the transcription of s, and y denotes the translation in the target language in AST, or the transcription in the case of ASR. When employing an Enc-Dec model, the training objectives encompass the utilization of joint CTC loss to model x at the encoder level, coupled with cross-entropy (CE) loss to model y within the decoder. Thus, it can be formulated as:

$$\mathcal{L}_{\text{CTC}}(h, x) = -\log \text{P}_{\text{CTC}}(x|h; \theta_{Enc}) \quad (3)$$

$$\mathcal{L}_{\text{CE}}(h, z, y) = -\log \text{P}_{\text{CE}}(y|h, z; \theta) \quad (4)$$

where h is the output of the encoder, and z is the input embedding of the decoder. θ_{Enc} and θ are the model parameters of the encoder and the whole network. Two hyper-parameters w_{CTC} and w_{CE} are introduced to balance CTC and CE loss components:

$$\mathcal{L} = w_{\text{CTC}} \cdot \mathcal{L}_{\text{CTC}} + w_{\text{CE}} \cdot \mathcal{L}_{\text{CE}} \quad (5)$$

To apply the IPA in S2T tasks, several significant distinctions must be noted when compared to conventional classification tasks:

- In typical classification models, the architecture usually comprises only the encoder, whereas in the S2T model based on the Enc-Dec architecture, the decoder processes the embedding sequence as input. The feasibility of directly interpolating word embeddings remains an open question.
- The label in classification tasks often takes the form of a one-hot category, thereby simplifying the interpolation process, while the S2T tasks present a more complex scenario. Specifically, the training objectives for CTC and CE are discrete text sequences, and the method to interpolate and learn the label effectively remains an open question.

To address these challenges, we first design the interpolation strategy grounded in previous studies, followed by an exploration of specific issues. Consider two arbitrary samples in a batch, denoted as (s_i, x_i, y_i) and (s_j, x_j, y_j). We interpolate the input according to Eq. (1):

$$s_m = \lambda \cdot s_i + (1 - \lambda) \cdot s_j \quad (6)$$

where we pad the shorter features with zeros to achieve the same length for interpolation. After obtaining the representation h_m outputted by the encoder, we calculate the CTC loss with respect to both labels and interpolate them as follows:

$$\begin{aligned}\mathcal{L}_{\text{CTC}}(h_m, x_i, x_j) = \lambda \cdot \mathcal{L}_{\text{CTC}}(h_m, x_i) \\ +(1 - \lambda) \cdot \mathcal{L}_{\text{CTC}}(h_m, x_j)\end{aligned} \quad (7)$$

Employing the widely proven interpolation strategy Zhang et al. (2017) in the encoder is natural due to similar designs. Thereby we focus on the interpolation strategy within the decoder. A straightforward implementation is similar to the operation in the encoder, which involves interpolating the embeddings z_i and z_j in the input layer of the decoder:

$$z_m = \lambda \cdot z_i + (1 - \lambda) \cdot z_j \quad (8)$$

Next, we calculate losses with two labels y_i and y_j for interpolation. The whole procedure is formalized as:

$$\begin{aligned}\mathcal{L}_{\text{CE}}(z_m, h_m, y_i, y_j) = \lambda \cdot \mathcal{L}_{\text{CE}}(h_m, z_m, y_i) \\ +(1 - \lambda) \cdot \mathcal{L}_{\text{CE}}(h_m, z_m, y_j)\end{aligned} \quad (9)$$

For simplicity, we refer to this strategy as embedding interpolation (*EIP*).

However, the preceding approach may lead to a disparity between training and decoding. During training, the decoder takes the interpolated embedding sequence as input, whereas it receives only a single embedding sequence during inference. To bridge this gap, we investigate an alternative strategy that solely interpolates the encoder input while preserving the original input in the decoder (Meng et al., 2021). The loss in this context is calculated as follows:

$$\begin{aligned}\mathcal{L}_{\text{CE}}(h_m, z_i, z_j, y_i, y_j) = \lambda \cdot \mathcal{L}_{\text{CE}}(h_m, z_i, y_i) \\ +(1 - \lambda) \cdot \mathcal{L}_{\text{CE}}(h_m, z_j, y_j)\end{aligned} \quad (10)$$

Method	α	γ	*EIP*	clean	other	Avg.
Baseline	-	-	-	11.85	30.78	20.81
IPA	0.2	1.0		10.31	25.12	17.37
	2.0	0.3		10.31	26.44	18.00
	2.0	1.0		10.14	**22.45**	**15.99**
	0.2	1.0	√	10.29	25.53	17.48
	2.0	0.3		10.40	26.67	18.02
	2.0	1.0		**9.91**	22.90	16.35

Table 1: WER of **IPA** method applied to the **Enc-Dec** model **without** SpecAugment on the LibriSpeech 100h dataset.

In summary, two interpolation strategies have distinct characteristics. The first approach leverages simple interpolation operations akin to those in the encoder and contributes to the regularization of the decoder. Conversely, the second approach focuses on consistent modeling, bypassing interpolation in the decoder, and may facilitate more stable learning.

We construct experiments to validate these two strategies. For the hyper-parameters, we select α from the set $\{0.2, 0.5, 1.0, 2.0\}$ and γ from $\{0.3, 1.0\}$. The partial results presented in Table 1 illustrate that enhancing noise by increasing either α or γ serves to reinforce generalization, leading to significant improvements, particularly on the more noisy test-other set. In addition, the *EIP* strategy exhibits slightly inferior performance. This observation aligns with our initial conjecture.

4 Q2: Combination of Augmentation Techniques

Although the straightforward application of the IPA method has yielded noticeable improvements, our exploration seeks to combine it with existing data augmentation techniques.

4.1 Preliminary Results

Table 2 presents the results obtained when using SpecAugment. Compared with the IPA method, SpecAugment is more effective in enhancing performance. However, excessive interpolation intensity inversely affects the results, leading to performance degradation. Reducing the values of α and γ alleviates this issue, though it yields only modest gains.

Another noteworthy observation is that the *EIP* strategy promotes a more stable training process, despite a decline in performance. This phenomenon might be attributed to the inherent sensitivity of the original model to noise, coupled with an apparent deficiency in handling complex input within the decoder. The enhanced robustness introduced by SpecAugment appears to mitigate this issue, empowering the decoder to handle interpolated input and extract information from the noisy encoder output.

Method	α	γ	*EIP*	clean	other	Avg.
Baseline	-	-	-	8.51	19.05	13.63
IPA	0.2	0.3		8.45	**18.68**	**13.46**
	0.2	1.0		8.75	19.51	13.89
	2.0	0.3		9.19	19.88	14.27
	2.0	1.0		11.01	23.48	16.88
	0.2	0.3	√	**8.29**	18.97	13.53
	0.2	1.0		8.73	19.39	13.80
	2.0	0.3		8.71	19.01	13.65
	2.0	1.0		10.36	20.24	15.07

Table 2: WER of **IPA** method applied to the **Enc-Dec** model **with** SpecAugment on the LibriSpeech 100h dataset.

4.2 Why Does the Combination Fail?

To optimize the combination between SpecAugment and IPA, it is crucial to shed light on the influence of SpecAugment on the IPA approach. Both two methods function by introducing regularization into the encoder input, targeting a balance to improve the model's ability to generalize without causing it to under-fit. With the right amount of noise, the model may take longer to reach its best performance but eventually perform better.

However, too much noise may result in troubles, leading to training failures or poor results. We think that the noise added by SpecAugment might mess up the interpolation, synthesizing samples that stray too far from the desired vicinal distribution. As the original samples are replaced with the interpolated version, it leads to poor learning of the actual data distribution.

To validate our conjecture, we visualize the data distribution of original and interpolated samples by t-SNE. Figure 1 (Top) shows that interpolated samples maintain a similar distribution to that of the original samples when SpecAugment is not employed, even with a large interpolation weight. However, this distribution uniformity is disrupted with the introduction of SpecAugment, giving rise to an evident discrepancy in distribution, as illustrated in Figure 1 (Middle).

Although the initial input representations (the first column) appear similar thanks to the cepstral

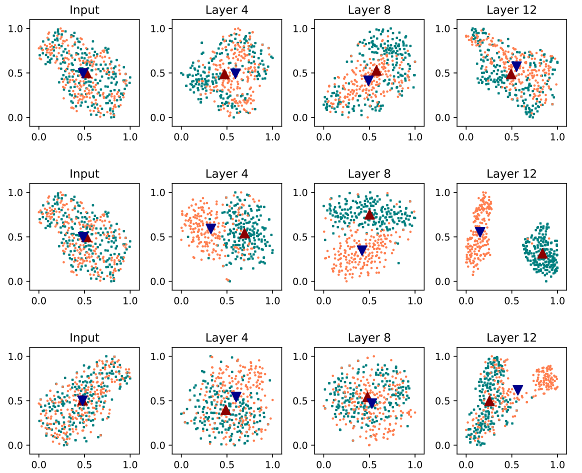

Figure 1: Visualization of encoder representations of both original (depicted as green squares) and interpolated (depicted as pink circles) samples in the IPA method. The upper triangle and lower triangle represent the centers of two data distributions, respectively. The experiment is conducted using the LibriSpeech 100h dataset with an interpolation ratio of $\gamma = 0.3$. Top: without SpecAugment and $\alpha = 2.0$. Middle: with SpecAugment and $\alpha = 2.0$. Bottom: with SpecAugment and $\alpha = 0.2$.

mean and variance normalization operation, the excessive perturbation caused by SpecAugment leads to a deviation of the interpolated samples from the original empirical distribution during encoding. This phenomenon, referred to as *distribution shift*, can be slightly mitigated by diminishing the intensity of the interpolation, thus narrowing the divergence between the two data distributions, as shown in Figure 1 (Bottom). However, traces of the distribution shift persist in the representation at the top layers. This inconsistency with the middle layers stems from the influence of the decoder, which we discuss subsequently.

4.3 Appending-based IPA

To mitigate the problem of distribution shift identified previously, the key is to prevent the interpolated samples from disturbing the stable learning of the original data distribution. The "*replace*" operation within the conventional IPA method is revealed to be suboptimal, constraining the magnitude of permissible regularization techniques. As an alternative, we introduce an "*appending*" operation into the IPA methodology, referred to as **AIPA**. Specifically, for an original batch comprising n samples, AIPA synthesizes $\lceil n \times \gamma \rceil$ interpolated samples. These are concatenated with the original batch, resulting in a new batch size of $\lceil n \times (1+\gamma) \rceil$ for training. This simple approach preserves all original samples and generates interpolated ones, thereby safeguarding stable training while simultaneously enabling robust regularization.

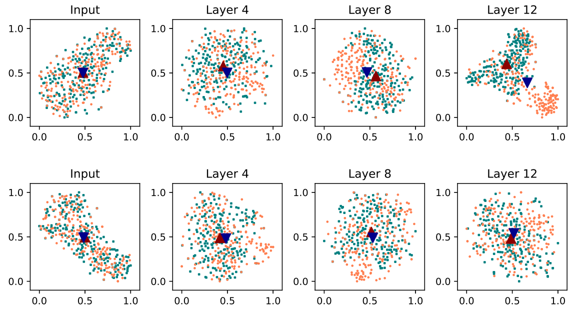

Figure 2: Similar to Figure 1, visualization of encoder representations in the **AIPA** method. Top: Enc-Dec model with SpecAugment, $\alpha = 0.2$. Bottom: Enc-CTC model with SpecAugment, $\alpha = 0.2$.

Method	α	γ	*EIP*	clean	other	Avg.
Baseline	-	-	-	8.51	19.05	13.63
AIPA	0.2	0.3		8.13	18.95	13.36
	0.2	1.0		8.01	18.52	13.05
	2.0	0.3		8.26	18.39	13.16
	2.0	1.0		8.48	18.91	13.48
	0.2	0.3	√	8.45	18.72	13.25
	0.2	1.0		7.91	**18.14**	**12.79**
	2.0	0.3		8.30	18.57	13.27
	2.0	1.0		**7.88**	18.17	12.95

Table 3: WER of **AIPA** method applied to the **Enc-Dec** model **with** SpecAugment on the LibriSpeech 100h dataset.

Moreover, AIPA guarantees exhaustive learning of both the original and vicinal distributions, bridging the divergence between training and inference, as the original samples remain unaltered. As depicted in Figure 2 (Top), the distance between the two classes of samples has been significantly minimized.

The experimental results in Table 3 further validate these findings. AIPA yields modest and consistent improvements under the augmentation of varying intensities. Notably, the *EIP* operation appears to be advantageous. This phenomenon can be interpreted as an additional benefit conferred by AIPA, which serves to enhance the robustness of the decoder by introducing controllable regularization. Based on these results, we select $\alpha = 0.2$ and $\gamma = 1.0$ as the default hyper-parameters and employ *EIP* operation for the subsequent experiments.

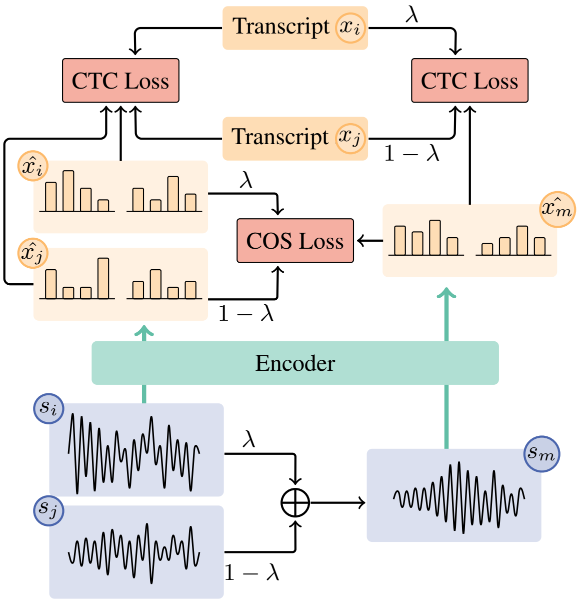

Figure 3: Encoding process of the AIPA method with COS training.

Model	dev		test		Avg.
	clean	other	clean	other	
Baseline	8.20	19.13	8.51	19.05	13.63
AIPA	7.56	17.95	7.91	18.14	12.95
+ CTC COS	**7.11**	**17.66**	**7.49**	17.85	**12.43**
+ CTC COS*	7.20	17.74	7.57	17.90	12.51
+ CE COS	7.41	17.92	7.82	17.99	12.69
+ Both COS	7.26	17.75	7.61	**17.80**	12.52

Table 4: WER of **AIPA** method applied to the **Enc-Dec** model **with** SpecAugment on the LibriSpeech 100h dataset. The α and γ are set to 0.2 and 1, respectively. COS* indicates using the hard labels.

5 Q3: Resolution of Specific Issues

While the current method does achieve stable effects, the enhancements are relatively modest. This section delves into further optimization by addressing the specific issues when employing AIPA in S2T tasks.

In the standard implementation, interpolated samples are given the dual responsibility of predicting two corresponding text sequences in both CTC and CE losses. However, this strategy might introduce a risk of ambiguity in the decision boundaries, potentially leading to an over-smoothed model. This risk is notably amplified during CTC learning, where the likelihood of a particular transcript x given the hidden representations h is obtained by summing over the probabilities of all feasible alignment paths $\Phi(x)$ between the speech s and x:

$$\mathrm{P}_{\mathrm{CTC}}(x|h) = \sum_{\pi \in \Phi(x)} \mathrm{P}(\pi|h) \tag{11}$$

This implies that each representation is required to cater to a multiplicity of labels, which substantially complicates the ideal predicted distribution, making it challenging to converge and somewhat counter-intuitive. Therefore, the design of appropriate training objectives for interpolated samples is pivotal.

We propose constraint objective space (COS), which facilitates CTC learning by replacing the complex traversal with deterministic labels. Rather than computing the best alignment by the model (Xu et al., 2023a), we take the predicted distribution of the original samples as the objective of the interpolated samples for efficiency. Specifically, we calculate the COS loss as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{CTC}}^{\mathrm{COS}}(h_m, h) = &-\sum_{m=1}^{T}\sum_{k=1}^{|V|} \mathrm{P}(\pi_m = v^k|h) \\ &\times \log \mathrm{P}(\pi_m = v^k|h_m)\end{aligned} \tag{12}$$

where T represents the length of h, and V denotes the vocabulary. Drawing a parallel to learning on text labels, we formulate the interpolation of the losses as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{CTC}}^{\mathrm{COS}}(h_m, h_i, h_j) = &\ \lambda \cdot \mathcal{L}_{\mathrm{CTC}}^{\mathrm{COS}}(h_m, h_i) \\ &+ (1-\lambda) \cdot \mathcal{L}_{\mathrm{CTC}}^{\mathrm{COS}}(h_m, h_j)\end{aligned} \tag{13}$$

In this framework, the original samples act as a *teacher*, guiding the more accessible learning process of the interpolated *student* (Hinton et al., 2015). This distribution offers detailed information across the entire vocabulary, and importantly, the training objective becomes more deterministic, thereby simplifying the learning process. The final design of AIPA with COS is depicted in Figure 3.

Similarly, this strategy can be extended to the cross-entropy (CE) loss, denoted by $\mathcal{L}_{\mathrm{CE}}^{\mathrm{COS}}$. The final training objective thus takes the form:

$$\begin{aligned}\mathcal{L} &= w_{\mathrm{CTC}} \cdot \mathcal{L}_{\mathrm{CTC}} + w_{\mathrm{CTC}}^{\mathrm{COS}} \cdot \mathcal{L}_{\mathrm{CTC}}^{\mathrm{COS}} \\ &+ w_{\mathrm{CE}} \cdot \mathcal{L}_{\mathrm{CE}} + w_{\mathrm{CE}}^{\mathrm{COS}} \cdot \mathcal{L}_{\mathrm{CE}}^{\mathrm{COS}}\end{aligned} \tag{14}$$

where $w_{\mathrm{CTC}}^{\mathrm{COS}}$ and $w_{\mathrm{CE}}^{\mathrm{COS}}$ are weights of two COS losses.

We present the results in Table 4. Utilizing COS for CTC training yields an average significant reduction of 1.35 WER points, as this approach simplifies CTC learning by eliminating the need for

Method	dev		test		Avg.
	clean	other	clean	other	
Baseline	9.58	23.07	9.99	23.84	16.50
+ InterCTC	8.18	20.19	8.47	20.73	14.28
+ PAE	8.09	19.85	8.32	20.76	14.15
AIPA	8.77	21.41	9.07	21.75	15.14
+ CTC COS	7.16	17.93	7.39	18.17	12.57
+ InterCTC	7.74	19.73	8.12	20.09	13.82
+ CTC COS	7.03	17.43	7.37	17.80	12.31
+ Both COS	6.73	17.07	6.99	17.35	11.94
+ PAE	**6.44**	**16.49**	**6.70**	**16.67**	**11.49**

Table 5: WER of **AIPA** method applied to the **Enc-CTC** model **with** SpecAugment on the LibriSpeech 100h dataset. The α and γ are set to 0.2 and 1, respectively.

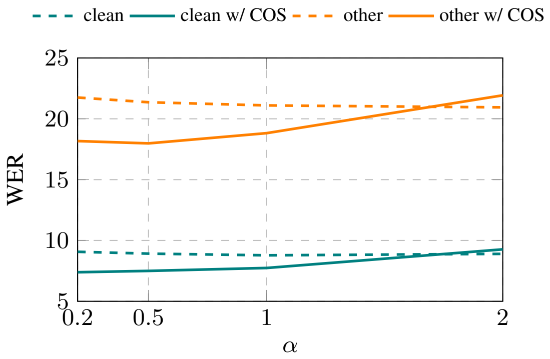

Figure 4: Effects of the hyper-parameters α on Enc-CTC models trained with LibriSpeech 100h dataset.

complex dynamic programming. Note that the soft training objective is not necessary. The main motivation is to provide simplified labels for stable learning. Replacing the distribution with the one-hot labels by *argmax* operation also achieves obvious effects. However, the application of COS in CE adversely affects performance. We speculate that the CE objective is more straightforward to learn, whereas the COS method might introduce errors.

6 Q4: Effect under Various Scenarios

We have obtained numerous valuable insights from the ablation studies conducted on the LibriSpeech 100h dataset. We now extend the application of the aforementioned settings to a broader array of scenarios.

6.1 Model Architectures

Combining the above efforts, we develop an effective interpolation augmentation method, which achieves significant improvements in the Enc-Dec architecture. The effects are further explored on the Enc-CTC model, with results presented in Table 5.

Due to the inherent conditional independence assumption of CTC modeling, the baseline model struggles to converge well. To build more robust configurations, we employ popular techniques to enhance the model's performance. Utilizing InterCTC (Lee and Watanabe, 2021), additional CTC supervisions are introduced into the intermediate layers, effectively bridging the gap. Meanwhile, the prediction-aware encoding (PAE) method (Xu et al., 2023a) integrates self-predicted information, yet only achieves slight improvements due to the limited accuracy of the CTC prediction.

AIPA yields more substantial improvements on the Enc-CTC model, addressing its inherent fragility. The COS method significantly aids CTC learning, resulting in a reduction of 2.57 WER points. This result demonstrates the appropriate training objective facilitates convergence effectively. Within the AIPA method, the intermediate CTC loss is computed similarly to the standard CTC, but its direct use has limited impact. However, when coupled with joint COS methods, it achieves gains of 1.88 WER points. Finally, thanks to the improved prediction of intermediate CTC, the PAE method also exhibits notable effects. Combining these methods achieves a remarkable reduction of 2.66 WER points over the baseline model.

Beyond merely improving performance, we also examine the data distribution within the Enc-CTC model, as depicted in Figure 2 (Bottom). Except for applications on various architectures, the settings are consistent with those in the preceding figure. In the Enc-CTC model, both the original and interpolated samples share the same representation space in the top layers. This observation suggests that the distribution shift in the Enc-Dec model is attributable to the behavior of the decoder. We speculate that the decoder must differentiate between two data distributions to capture information effectively, whereas the CTC objective diminishes this need, thereby maintaining a similar distribution.

Hyper-parameter α has significant influences on the final performance. We illustrate the results of the AIPA method both with and without the COS method in Figure 4. AIPA achieves stable results by preserving the original data distribution, and variations in α have only a minor impact. However, increasing α negatively affects the efficacy of the COS method. A possible explanation is that a larger α results in a more balanced sample interpolation between two original samples, leading to increased COS loss and poor convergence.

In summary, our findings indicate that the IPA

Dataset	Method	dev		test		Avg.
		clean	other	clean	other	
10h	Baseline	35.34	51.89	35.13	53.20	43.74
	AIPA	**28.34**	**43.89**	**28.46**	**44.76**	**36.22**
50h	Baseline	13.10	28.40	13.48	29.46	21.03
	AIPA	**10.54**	**22.50**	**10.84**	**23.12**	**16.64**
960h	Baseline	3.47	9.34	3.61	9.02	6.31
	AIPA	**2.91**	**7.61**	**3.01**	**7.51**	**5.21**

Table 6: WER of **AIPA** method applied to the **Enc-CTC** model **with** SpecAugment on the LibriSpeech 10h, 50h, and 960h dataset. InterCTC is used for all models and the COS technique is used in AIPA.

Method	Transformer	Conformer
Baseline	6.06	7.16
+ InterCTC	5.67	5.87
+ PAE	5.32	5.81
AIPA	5.58	6.14
+ CTC COS	5.12	4.55
+ InterCTC	5.15	4.53
+ InterCTC COS	5.05	4.35
+ PAE	**4.62**	**4.27**

Table 7: WER of **AIPA** method applied to the **Enc-CTC** model **with** SpecAugment on the AiShell-1 dataset.

Method	dev	tst-COMMON
Baseline	25.42	26.31
+ InterCTC	26.35	26.56
+ PAE	**26.62**	**26.62**
AIPA	25.85	26.38
+ CTC COS	26.04	26.75
+ CE COS	26.13	26.64
+ Both COS	26.79	26.88
+ InterCTC	26.48	26.68
+ All COS	**26.92**	**27.50**
+ PAE	26.69	27.39

Table 8: BLEU of **AIPA** method applied to the **Enc-Dec** model **with** SpecAugment on the MuST-C En-De ST dataset.

technique is particularly well-suited for the Enc-CTC architecture. This suitability may stem from multiple factors, such as the baseline model's inherent fragility, the compatibility of continuous features with interpolation, and the elimination of the decoder's influence. We will explore these reasons further in future research.

6.2 Data Scales

By integrating our proposed strategies, we achieve more significant improvements, especially on the noisy other test sets. Under the extreme low-resource scenarios of 10h and 50h data, our method achieves substantial reductions of about 4 $\sim$ 6 WER points and boosts the convergence speed effectively. Even under the high-resource scenario of 960h, AIPA still delivers further improvements. These findings indicate that the optimized IPA settings are not only effective in low-resource environments but also demonstrate their efficacy in high-resource scenarios.

6.3 Model Backbones

We explore the effects of our method with different model backbones on the AiShell-1 ASR dataset, incorporating speed perturbation. The results, displayed for both Transformer and Conformer models in Table 7, reveal some new insights. Interestingly, the base Conformer model underperforms its Transformer counterpart, potentially due to underfitting associated with larger model parameters. Despite incorporating auxiliary techniques, the Conformer model struggles to converge optimally.

Our proposed method effectively addresses this convergence issue. Notably, employing the COS method specifically for CTC learning offers outstanding regularization and significantly enhances the model's convergence. This observation underscores the advantages of our interpolation augmentation method over SpecAugment. Across both model architectures, our interpolation strategy yields stable and substantial improvements, illustrating its broadly applicable effectiveness.

6.4 AST Task

The AST task presents unique challenges due to the substantial modeling complexity involved in handling both cross-modality and cross-lingual mapping. In this demanding context, the fundamental AIPA method delivers only modest improvements, as shown in Table 8. However, with the application of our proposed learning objectives for the interpolated samples, we observe more substantial gains.

A notable distinction is the effectiveness of the COS method for CE loss. This likely stems from the increasing task complexity, where the distribution may be more readily learned by the decoder, thereby easing the training process. Remarkably, without resorting to intricate designs, our method achieves a BLEU score of 27.50. This performance is highly competitive, approaching current state-of-the-art results where no additional training data are employed.

7 Conclusion

In this paper, we develop a comprehensive exploration of the interpolation augmentation (IPA) method's application in S2T generation. Our findings provide actionable insights for the effective application of IPA in S2T: (1) Utilizing IPA alone may not surpass the effectiveness of SpecAugment; a careful combination of both lies in mitigating distribution shift and preserving the learning of original data distribution. (2) Defining an appropriate training objective for interpolated samples is of paramount importance. (3) IPA demonstrates particular compatibility with the Enc-CTC model. (4) The appropriate IPA strategy significantly enhances performance across diverse scenarios.

Limitations

Although our method demonstrates exceptional performance in various scenarios, there are still some underlying challenges that remain in the follow-up of our work. We outline key limitations and propose future directions for improvement:

- Enhancing stability with diverse hyperparameters: As depicted in Figure 4, a larger value of α leads to the generation of excessively noisy interpolated samples, adversely affecting the WER. This underscores the need for a more robust IPA method and the determination of universally effective hyperparameters to ensure broader applicability.
- Adapting to pre-trained models: The S2T field boasts several influential open-source, pretrained models such as Wav2vec2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and Whisper (Radford et al., 2023). Integrating our IPA method with these established models is a promising avenue that requires thorough validation and exploratory research.

Acknowledgements

The authors would like to thank anonymous reviewers for their insightful comments. This work was sponsored by research team project supported by Natural Science Foundation of Heilongjiang (grant no.TD2022F001), NSFC-Xinjiang Joint Fund Key Program (grant no.U2003206), NSFC-Regional Joint Fund Key Program (grant no.U20B2048), National Natural Science Foundation of China (grant no.U21B2019), NSFC-Regional Joint Fund Key Program (grant no.U22A2036), and National Natural Science Foundation of China (grant no.62272127).

References

Ashish Alex, Lin Wang, Paolo Gastaldo, and Andrea Cavallaro. 2023. Data augmentation for speech separation. *Speech Communication*.

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

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.

Xuxin Cheng, Qianqian Dong, Fengpeng Yue, Tom Ko, Mingxuan Wang, and Yuexian Zou. 2022. M3ST: mix at three levels for speech translation. *CoRR*, abs/2212.03657.

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.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Lei Kang, Lichao Zhang, and Dazhi Jiang. 2023. Learning robust self-attention features for speech emotion recognition with label-adaptive mixup. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1408–1412. ISCA.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Max WY Lam, Jun Wang, Dan Su, and Dong Yu. 2020. Mixup-breakdown: a consistency training method for improving generalization of speech separation models. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6374–6378. IEEE.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for ctc-based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 6224–6228. IEEE.

Wuyang Liu, Yanzhen Ren, and Jingru Wang. 2023. Attention mixup: An accurate mixup scheme based on interpretable attention mechanism for multi-label audio classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

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.

Ivan Medennikov, Yuri Y Khokhlov, Aleksei Romanenko, Dmitry Popov, Natalia A Tomashenko, Ivan Sorokin, and Alexander Zatvornitskiy. 2018. An investigation of mixup training strategies for acoustic models in asr. In *Interspeech*, pages 2903–2907.

Linghui Meng, Jin Xu, Xu Tan, Jindong Wang, Tao Qin, and Bo Xu. 2021. Mixspeech: Data augmentation for low-resource automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7008–7012. IEEE.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. 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, Demonstrations*, pages 48–53. Association for Computational Linguistics.

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.

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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu. 2019. Speech recognition with augmented synthesized speech. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 996–1002. IEEE.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. 2017. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, volume 2017, pages 999–1003.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020. Mixup-transformer: dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR.

Deng-Bao Wang, Lanqing Li, Peilin Zhao, Pheng-Ann Heng, and Min-Ling Zhang. 2023. On the pitfall of mixup for uncertainty calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7609–7618.

Xiangjin Xie, Li Yangning, Wang Chen, Kai Ouyang, Zuotong Xie, and Hai-Tao Zheng. 2023. Global mixup: Eliminating ambiguity with clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13798–13806.

Chen Xu, Xiaoqian Liu, Xiaowen Liu, Qingxuan Sun, Yuhao Zhang, Murun Yang, Qianqian Dong, Tom Ko, Mingxuan Wang, Tong Xiao, et al. 2023a. Ctc-based non-autoregressive speech translation. *arXiv preprint arXiv:2305.17358*.

Chen Xu, Rong Ye, Qianqian Dong, Chengqi Zhao, Tom Ko, Mingxuan Wang, Tong Xiao, and Jingbo Zhu. 2023b. Recent advances in direct speech-to-text translation. *arXiv preprint arXiv:2306.11646*.

Xue Ying. 2019. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Yu Zhang, Daniel S Park, Wei Han, James Qin, Anmol Gulati, Joel Shor, Aren Jansen, Yuanzhong Xu, Yanping Huang, Shibo Wang, et al. 2022. Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532.

Yan Zhou, Qingkai Fang, and Yang Feng. 2023. Cmot: Cross-modal mixup via optimal transport for speech translation. *arXiv preprint arXiv:2305.14635*.

A Experimental Settings

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). To assess the performance in both low-resource and high-resource environments, we conduct experiments on LibriSpeech 10h, 50h, 100h, and 960h. We report results on all four subsets, including dev-clean, dev-other, test-clean, and test-other. The average word error rate (WER) is calculated 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 report results WER on both the dev and test sets.
- **MuST-C** is a multilingual speech translation corpus extracted from the TED talks (Gangi et al., 2019). We test our method on the MuST-C English-German (En-De) speech translation dataset of 400 hours of speech. We select (and tune) the model on the dev set (Dev) and report the results on the tst-COMMON set (Test).

For pre-processing, we follow the standard recipes in fairseq toolkit (Ott et al., 2019), which eliminates the utterances of more than 3,000 frames or fewer than 5 frames. To explore the impact of integrating another augmentation method, we employ speed perturbation in our experiments conducted on the AiShell-1 dataset. The extraction of 80-channel Mel filter bank features is carried out using a 25ms window and a stride of 10ms. For segmentation, we employ SentencePiece (Kudo and Richardson, 2018) segmentation with a size of 10,000 for the LibriSpeech 100h and MuST-C datasets, 256 for the LibriSpeech 960h dataset. And the AiSHELL-1 dataset is segmented using 4231 characters. For the MuST-C AST dataset, we utilize a shared vocabulary for the source and target languages.

A.2 Model Settings

We train the ASR model using the Enc-CTC architecture and AST models with the Enc-Dec architecture. α and γ are set to 0.2 and 1, respectively. The weight w_{CTC} and w_{CE} for the training objective are set to 0.3 and 1.0 in the encoder-decoder model, while 1.0 and 0.0 in the Enc-CTC model. And the weight $w_{\text{CTC}}^{\text{COS}}$ and $w_{\text{CE}}^{\text{COS}}$ for the COS method are set to the half of them. SpecAugment (Park et al., 2019) is always applied for better results. Note that our pipeline first applies SpecAugment pre-processing, then performs interpolation augmentation (IPA) on the SpecAugmented samples. This order allows IPA to increase diversity on top of the distortions from SpecAugment

All methods are implemented using the fairseq toolkit. We employ the Adam optimizer and follow the default learning schedule in fairseq. We apply dropout with a rate of 0.1 and label smoothing

$\epsilon_{ls} = 0.1$ for regularization. Note that the feed-forward size is set to 1024 on the LibriSpeech 960h dataset for comparison with previous results.

We do not incorporate pre-training and knowledge distillation techniques during the training process. We train the model 300 epochs on LibriSpeech 100h and 960h for better convergence and 100 epochs for both AiShell-1 ASR and MuST-C AST datasets. We early stop training when there is no performance improvement on the development set for 20 consecutive checkpoints. We report WER/CER and case-sensitive SacreBLEU for ASR and AST tasks, respectively.

A.3 Augmentation Settings

In our methodology, SpecAugment is always applied first, followed by sample interpolation. This sequence is based on two key considerations:

- SpecAugment is a per-sample operation, whereas IPA can be batch-processed. Applying SpecAugment before IPA results in greater efficiency.
- Employing IPA after SpecAugment introduces additional perturbations, potentially enhancing regularization effects. In addition, employing IPA after SpecAugment is easier from the perspective of the code implementation.