Multimodal Representation Loss Between Timed Text and Audio for Regularized Speech Separation

Tsun-An Hsieh[1], *Heeyoul Choi*[2], *Minje Kim*[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]School of Computer Science and Electrical Engineering, Handong Global University, Korea

tsunanh2@illinois.edu, hchoi@handong.edu, minje@illinois.edu

Abstract

Recent studies highlight the potential of textual modalities in conditioning the speech separation model's inference process. However, regularization-based methods remain underexplored despite their advantages of not requiring auxiliary text data during the test time. To address this gap, we introduce a timed text-based regularization (TTR) method that uses language model-derived semantics to improve speech separation models. Our approach involves two steps. We begin with two pretrained audio and language models, WavLM and BERT, respectively. Then, a Transformer-based audio summarizer is learned to align the audio and word embeddings and to minimize their gap. The summarizer Transformer, incorporated as a regularizer, promotes the separated sources' alignment with the semantics from the timed text. Experimental results show that the proposed TTR method consistently improves the various objective metrics of the separation results over the unregularized baselines.

Index Terms: Speech source separation, language model, multimodal learning

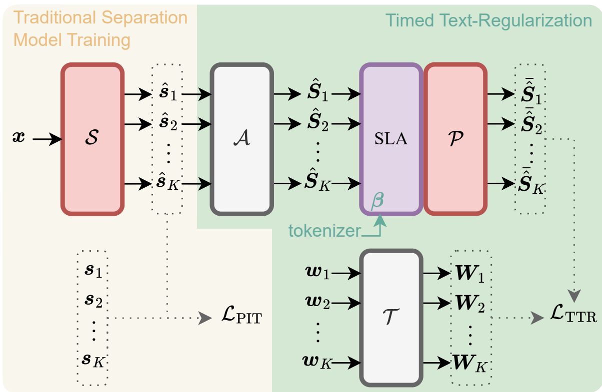

Figure 1: *The proposed TTR-SS training pipeline. The yellow and green areas represent the traditional training of a speech separation model and the proposed TTR, respectively. The test-time inference only uses the yellow area.*

1. Introduction

Recent studies have shown significant progress in deep learning-based audio source separation [1, 2, 3, 4, 5, 6], among which end-to-end approaches are popular approaches. For example, Conv-TasNet [7] established the encoder-separator-decoder structure, which removed traditional time-frequency feature extraction, such as magnitude spectrogram or mel-frequency cepstral coefficients. Dual-Path RNN [8] followed to capture both the temporal and spatial dependencies through modeling across both directions. In addition, introducing the Transformer [9] architectures to source separation also advanced the performance due to their self-attention mechanism, such as Dual-Path Transformer [10] and SepFormer [11, 12].

Another category of studies focuses on leveraging cross modality clues, e.g. visual and textual queries, as auxiliary information that conditions the separation system. Hence, this type of systems is suitable for extracting out a source of interest defined by the cue, which is a task often called *target source extraction* (TSE). Query-based approaches have been widely investigated in the fields of singing voice separation [13], speech separation [14], and sound separation [15, 16, 17, 18]. Among these works, the text modality has been one of the primary ways to convey information about the target source. For example, in [15], textual description or sample audio from the same speaker is used to designate the target speaker; LASS-Net [16] conditions the hidden vectors in the separation network with a Transformer-based query network to extract textually described sounds; CLIPSep [17] used contrastive language-image-audio pretraining to learn a joint embedding for trimodal representation and used it for TSE; in [18], a heterogeneous TSE task is defined to condition the separation network using various concepts such as gender, language, and loudness.

Although these successful TSE systems use auxiliary information for effective conditioning, using other modalities to *regularize* the separation task during training has not been studied in depth. The Voice ID loss [19] is an example, where the speech enhancement model is regularized to reduce both the typical reconstruction and the speaker verification loss. The prototypical speaker-interference (PSI) loss [20] also uses a speaker-level representation loss to regularize a TSE system. While these regularization methods are effective, they are limited to the audio modality, leaving room for investigating other modalities, such as text. For example, using an ASR loss jointly with the signal reconstruction loss is another promising multimodal approach while they are more focused on the ASR performance than reconstruction and do not consider inter-word relationships as in our method [21, 22, 23].

In this work, our main contribution is the multimodal representation loss defined between timed text and speech audio to regularize speech separation models. Compared to the other TSE methods, where the separation model is indirectly conditioned by auxiliary text [15, 16] or an unaligned script [13], our text modality is high-quality and strongly associated with the source audio. In particular, we use source-specific scripts of all the clean utterances for training, assuming their word-level synchronization with the audio frames. This assumption is rather strong to condition the model during testing as in the TSE methods because acquiring a time-aligned transcript of the test-time sources is extremely difficult. On the contrary, the proposed TTR method provides consistent performance improvement at no additional cost, i.e., without asking the user for a time-aligned transcript of the target source. Furthermore, since the proposed method is to regularize the training objectives, it does not add any computational cost to the test-time inference.

Figure 1 illustrates the proposed regularization method for speech separation. First, the two pretrained audio and text encoders, $\mathcal{A}$ and $\mathcal{T}$, convert K separated speech sources $\hat{\boldsymbol{s}}_{k\in\{1,\ldots,K\}}$ and their corresponding timed text $\boldsymbol{w}_k$ into frame-level audio embeddings $\hat{\boldsymbol{S}}_k$ and subword-level text embeddings $\boldsymbol{W}_k$, respectively. Second, the subword-level alignment (SLA) module associates consecutive audio embeddings with a subword embedding. Third, the summarizer Transformer $\mathcal{P}$ follows to convert each subword-specific audio embeddings into a summary vector, representing the subword in the audio modality. Finally, our TTR loss computes the similarity between the subword-level aggregation of the audio embeddings $\bar{\hat{\boldsymbol{S}}}$ and the corresponding subword embedding from the text encoder.

The pretrained summarizer, WavLM, and BERT, are frozen and combined as a regularization network, which provides an audio-text matching score to finetune the speech separation network. We call the finetuned separation network *Timed Text-Regularized Speech Separation* (TTR-SS). Experimental results demonstrate that the proposed TTR-SS improves the performance of two baseline separation systems on two and three-speaker speech separation tasks with additive noise, perhaps due to the sentence-level semantics introduced to the loss function. Moreover, we note that TTR enhances the more complex SepFormer to a greater extent than it does the simpler Conv-TasNet, indicating that the TTR loss introduces more information for the network to learn from, requiring a larger model capacity.

2. Timed Text-Regularized Source Separation

2.1. Problem Definition and the Loss Function

Given a time-domain mixture signal $\boldsymbol{x}$ that consists of K speech sources and a non-speech source $\boldsymbol{n}$, i.e., $\boldsymbol{x} = \sum_{k=1}^{K} \boldsymbol{s}_k + \boldsymbol{n}$, a speech separation model $\mathcal{S}$ is expected to estimate the sources back, $\{\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \ldots, \hat{\boldsymbol{s}}_K\} \leftarrow \mathcal{S}(\boldsymbol{x})$, with a potential permutation of the source order. To compare each estimate to the best-matching target source, during training, the permutation invariant training (PIT) scheme [24] is commonly used. In particular, given a set of source estimates and a reconstruction loss function, e.g., negative SI-SDR, PIT searches for the best permutation that minimizes the total loss out of $K!$ potential permutations, whose l-th permutation is defined by $\boldsymbol{c}^{(l)} = \{i_1^{(l)}, \ldots, i_K^{(l)}\}$:

$$\mathcal{L}_{\text{PIT}} \coloneqq \min_{\boldsymbol{c}^{(l)},\ l\in\{1,\ldots,K!\}} \sum_{k=1}^{K} -\text{SI-SDR}\left(\boldsymbol{s}_k || \hat{\boldsymbol{s}}_{i_k^{(l)}}\right). \qquad (1)$$

2.2. Subword-Level Alignment

Timed text-regularization (TTR) takes advantage of the sentence-level semantics to guide the training of the separation model. To achieve this, we use a pretrained audio encoder and a language model (LM) to extract audio and word embeddings and then compare them to compute a regularization loss. We denote the extraction processes for text and audio by $\boldsymbol{W} = \mathcal{T}(\boldsymbol{w})$ and $\boldsymbol{S} = \mathcal{A}(\boldsymbol{s})$, respectively. Their input $\boldsymbol{w}$ and $\boldsymbol{s}$ are a sequence of subwords and a waveform, while $\boldsymbol{W} \in \mathbb{R}^{D_{\boldsymbol{W}} \times M}$ and $\boldsymbol{S} \in \mathbb{R}^{D_{\boldsymbol{S}} \times T}$ represent the word and audio embeddings that are $D_{\boldsymbol{W}}$ and $D_{\boldsymbol{S}}$-dimensional vectors, respectively. Note that M is the number of subwords, which is usually much smaller than audio frames N.

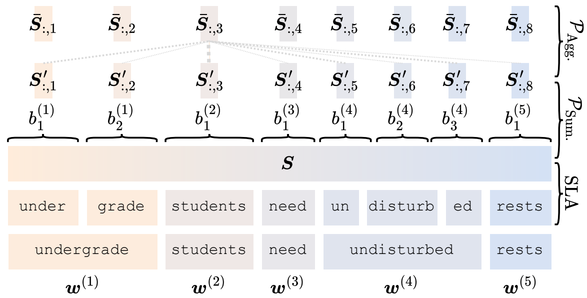

Figure 2: *Data flow of the SLA and summerizer Transformer. SLA divides audio embeddings into segments using subword boundaries. The $\mathcal{P}_{\text{Sum.}}$ summarizes each segment into a single vector. $\mathcal{P}_{\text{Agg.}}$ aggregates the summarized vectors to mimic the internal dependency of subword embeddings.*

We assume that reliable word boundaries of a ground-truth source utterance $\boldsymbol{s}$ are available, while their subword boundaries are not. Since a typical $\mathcal{T}$ model, e.g., BERT [25], uses tokenized subwords as its atomic input elements, the resulting embedding vectors $\boldsymbol{W}$ are at the subword level, which the audio embeddings $\boldsymbol{S}$ should be aligned to. As a remedy, we propose the *subword-level alignment* (SLA) algorithm (Figure 1), which infers the subword boundaries from the known word lengths. First, we assume the l-th word consists of m_l subwords, i.e., $\boldsymbol{w}^{(l)}{=}[w_1^{(l)}, \ldots, w_{m_l}^{(l)}]$. Hence, the entire subword sequence $\boldsymbol{w}{=}[\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}] = [w_1^{(1)}, \ldots, w_{m_1}^{(1)}, w_1^{(2)}, \ldots, w_{m_2}^{(2)}, \ldots w_1^{(L)}, \ldots, w_{m_L}^{(L)}]$, where L denotes the total number of words in $\boldsymbol{w}$. Therefore, $M{=}\sum_{l=1}^{L} m_l$. Then, for the known word lengths $\boldsymbol{b}{=}[b^{(1)}, \ldots, b^{(L)}]$, we simply postulate that each word length can be evenly divided, i.e., $b^{(l)}{=}b_1^{(l)}{+}\cdots{+}b_{m_l}^{(l)}$ and $b_m^{(l)}{=}b^{(l)}/m_l$, $\forall m$. The subdivision can redefine $\boldsymbol{b}{=}[b_1^{(1)}, \ldots, b_{m_1}^{(1)}, b_1^{(2)}, \ldots, b_{m_2}^{(2)}, \ldots, b_1^{(L)}, \ldots, b_{m_L}^{(L)}]$ as the list of subword lengths. In addition, we also define the subword boundaries $\boldsymbol{\beta}{=}[\beta_0, \beta_1, \ldots, \beta_{M-1}, \beta_M]$, where $\beta_m{=}\sum_{m'=1}^{m} \boldsymbol{b}_{m'}$, i.e., the sum of the first m subword lengths, while $\beta_0 = 0$. With this information, we can group the audio embeddings into subword-specific sub-sequences. For example, the t-th audio embedding $\boldsymbol{S}_{:,t}$ belongs to the m-th subword if $\beta_{m-1}{<}t/R{<}\beta_m$, where R denotes the frame rate of the audio embedding, i.e., the number of embeddings per second.

Specifically, BERT and WavLM [26] are adopted in this work for $\mathcal{T}$ and $\mathcal{A}$ for subword and audio embeddings extraction. The word-level boundaries are computed by Montreal Forced Aligner [27], which are further divided into the subword level using the abovementioned SLA algorithm.

2.3. Summarizer Transformer

A successful SLA results in multiple audio embeddings associated with a subword embedding due to the audio modality's higher frame rate. Hence a summarizing mechanism is required, for which we propose the *summarizer Transformer* function

$\mathcal{P}$. This Transformer consists of two parts: the subword-level summarizer $\mathcal{P}_{\text{Sum.}}$ and the sentence-level aggregator $\mathcal{P}_{\text{Agg.}}$ as shown in Figure 2. The subword summarizer, a two-layer Transformer encoder, summarizes each aligned audio embedding sequence within the subword boundary into a feature vector with the same dimensionality with a word embedding, i.e., $\boldsymbol{S}'_{:,m} \leftarrow \mathcal{P}_{\text{Sum.}}(\boldsymbol{S}_{:,\beta_{m-1}\le t<\beta_m})$. By repeating the process for all the subword-specific subsequences of $\boldsymbol{S}$, we get $\boldsymbol{S}' \in \mathbb{R}^{D_{\boldsymbol{W}}\times M}$. The sentence-level aggregator, which has an identical architecture to the summarizer, follows to transform the sequence of subword-level summarized audio embeddings into a final version, i.e., $\bar{\boldsymbol{S}} \leftarrow \mathcal{P}_{\text{Agg.}}(\boldsymbol{S}')$, without changing the dimension and length, resulting in the final audio representation $\bar{\boldsymbol{S}} \in \mathbb{R}^{D_{\boldsymbol{W}}\times M}$. Note that the sentence-level aggregation is over the M embeddings, ensuring that the final representation $\bar{\boldsymbol{S}}$ encodes long-term context across the entire sentence, which is otherwise missing during the subword-level summarizing process that operates within the subword boundaries.

Since the $\mathcal{P}$ function projects the audio embeddings $\boldsymbol{S}$ to the $D_{\boldsymbol{W}}$-dimensional space, where the audio embeddings are learned to be comparable to word embeddings. The $\mathcal{P}$ function learned to minimize the following timed text-regularization (TTR) loss function:

$$\mathcal{L}_{\text{TTR}}(\bar{\boldsymbol{S}}, \boldsymbol{W}) \coloneqq \frac{1}{M}\sum_{m=1}^{M}\left(1 - \frac{\bar{\boldsymbol{S}}_{:,m}\cdot \boldsymbol{W}_{:,m}}{\|\bar{\boldsymbol{S}}_{:,m}\|\|\boldsymbol{W}_{:,m}\|}\right), \quad (2)$$

i.e., the mean of the cosine distance between each pair of aligned subword-level embeddings from both modalities. Note that we use clean speech sources to pretrain $\mathcal{P}$, which is then frozen during the training of the separation model $\mathcal{S}$.

2.4. Timed Text-Regularized Source Separation

Figure 1 shows the finetuning pipeline for TTR-SS. The yellow area shows the traditional data flow of a speech separation model $\mathcal{S}$, which we pretrain using an ordinary speech separation pipeline: it takes a mixture $\boldsymbol{x}$ as input and predicts their constituent speech sources in the optimal order with the help from the PIT loss as shown in eq. (1). The timed-text regularizer (TTR) is also pre-trained as described in Sec. 2.2 and 2.3, using clean speech utterances and their corresponding timed texts.

The finetuning step further updates the separation module $\mathcal{S}$, while the summarizer Transformer, BERT, and WavLM modules are kept frozen. The difference is that, for finetuning, $\mathcal{P}$ takes the audio embeddings extracted from the source estimates as input rather than the clean utterances, i.e., $\hat{\bar{\boldsymbol{S}}}_k \leftarrow \mathcal{P}(\text{SLA}(\mathcal{A}(\hat{\boldsymbol{s}}_k)))$. Finally, we jointly minimize the PIT and TTR losses that are defined as the following total loss function:

$$\mathcal{L}_{\text{total}} \coloneqq \sum_{k=1}^{K}\mathcal{L}_{\text{PIT}}(\hat{\boldsymbol{s}}_k, \boldsymbol{s}_k) + \lambda\mathcal{L}_{\text{TTR}}(\hat{\bar{\boldsymbol{S}}}_k, \boldsymbol{W}_k), \quad (3)$$

with a blending weight λ chosen from $\{0.1, 0.5, 1.0\}$.

3. Experimental Setup

3.1. Dataset Description & Evaluation Metrics

We use the LibriMix dataset [28] to train and validate our proposed method. We use four subsets, Libri2Mix-Clean, Libri3Mix-Clean, Libri2Mix-Noisy, and Libri3Mix-Noisy, that consist of clean two- and three-speaker mixtures or their noisy versions. The speech sources are derived from LibriSpeech [29], while the noisy mixtures use ambient noises from the WHAM! dataset [30]. LibriMix follows the same structure as WHAM! and has two training sets, one validation set, and one test set. In this work, we use the *train-360*, *dev*, and *test* as the training, validation, and test sets, respectively, with an 8KHz sample rate. In addition, we test the systems on both the clean and noisy mixtures. Evaluation metrics used are the traditional BSS_Eval toolbox's decomposition of source-to-distortion ratio (SDR) into source-to-interference ratio (SIR) [31] as well as the scale-invariant SDR (SI-SDR) metric [32] and the short-time objective intelligibility (STOI) score [33].

3.2. Model Architecture and Training Setup

Baseline Model and Joint Finetuning We adopt Conv-TasNet [7] and SepFormer [11, 12] as our baseline models. Conv-TasNet consists of a 1-D convolutional encoder, a separator, and a decoder. The convolution encoder first encodes the raw waveforms into a 2-D feature map. The separator leverages the temporal convolution to estimate feature masks that separate the encoded 2-D feature map. Finally, the separated feature maps are transformed back to waveform predictions. Specifically, the 1-D convolutional encoder contains 24 convolutional blocks, where each block has 512 channels with a kernel size of 16, and each kernel strides by 8. With a similar encoder-separator-decoder structure, SepFormer mainly relies on the Transformer-based dual-path processing blocks as the separator. SepFormer repeats the separator twice, which contains eight layers of Transformers for inter- and intra-paths, and each Transformer layer has eight attention heads. The baseline model pretraining and TTR-SS finetuning share the same optimization configuration except that the loss functions are $\mathcal{L}_{\text{PIT}}$ and $\mathcal{L}_{\text{total}}$, respectively. We use the Adam optimizer [34] with a learning rate of 10^{-3} and 1.5×10^{-4} for Conv-TasNet and SepFormer, respectively. A learning rate scheduler halves the current learning rate if the validation loss is not reduced for five epochs. The training and finetuning are early-stopped if no improvement is seen after 30 epochs.

WavLM and BERT Two pre-trained models, WavLM and BERT, extract audio and word embeddings that contain phonetic and semantic information, respectively. Both use a Transformer encoder-based architecture, and we used their publicly available pretrained versions. To minimize computational overhead, we choose models with fewer parameters, i.e. *wavlm-base* and *bert-base-uncased*, and run the frozen models to extract the embeddings on-the-fly during training. The resulting dimensions of audio and word embeddings are both $D_{\boldsymbol{W}} = 768$.

Summarizer Transformer Both the subword-level summarizer and sentence-level aggregator have 768 input dimensions, which match WavLM's embedding, and are based on the same 4-layer Transformer encoder structure. As positional encodings are used in both WavLM and BERT, the summarizer Transformer opts not to use them. For the summarizer Transformer pretraining, Adam is again used with the learning rate, β_1, and β_2 set to be 10^{-4}, 0.9, and 0.98, respectively. No learning rate scheduler is involved. Early stopping engages when the validation loss is not improved for 30 epochs. $\mathcal{L}_{\text{TTR}}$ in eq. (2) is used for the pretraining, while finetuning is to reduce $\mathcal{L}_{\text{total}}$ by jointly updating both the separator $\mathcal{S}$ and the summarizer $\mathcal{P}$.

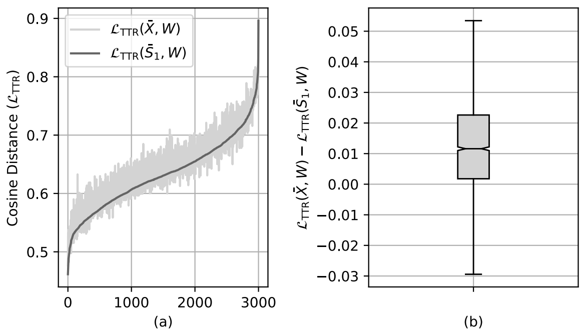

Figure 3: *Evaluation of the summarizer Transformer.*

4. Experimental Results and Discussion

4.1. Evaluation of the Summarizer Transformer

To verify the extent to which the textual data contributes, we evaluate the summarizer Transformer using the validation set. In Figure 3 (a), the horizontal axis represents the 3,000 utterances in the validation set, and the vertical axis stands for the cosine distance between the projected audio embeddings and subword embeddings. In this comparison, we show the embeddings from the matching source $\bar{\boldsymbol{S}}_1$ are more similar to the word embeddings $\boldsymbol{W}$ than the embeddings extracted from the mixtures $\bar{\boldsymbol{X}}$ are, where $\bar{\boldsymbol{X}} \leftarrow \mathcal{P}(\mathrm{SLA}(\mathcal{A}(\boldsymbol{x})))$. All the $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{S}}_1, \boldsymbol{W})$ values are sorted ascendingly, and the sorted indices are utilized to sort the values of $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{X}}, \boldsymbol{W})$. From Figure 3 (a), we see that the overall difference of $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{X}}, \boldsymbol{W})$ is higher than $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{S}}_1, \boldsymbol{W})$. Furthermore, Figure 3 (b) illustrates the distribution of the difference between $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{S}}_1, \boldsymbol{W})$ and $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{X}}, \boldsymbol{W})$. The notch indicates the 95% confidence interval, with 78.87% of $\mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{X}}, \boldsymbol{W}) \geq \mathcal{L}_{\mathrm{TTR}}(\bar{\boldsymbol{S}}_1, \boldsymbol{W})$, and the mean of the difference is 1.23×10^{-2}. These results indicate that the summarizer can distinguish between clean sources and mixtures based on the textual context.

4.2. Source Separation Performance

In this main experiment, we compare the performance of the Conv-TasNet and SepFormer baselines, and their TTR variants. Table 1 presents the evaluation improvement scores of the two separation models for each metric and task. Higher scores indicate better results. For a fair comparison, we use the pretrained checkpoints provided by the authors of both baseline models. Note that we use the SepFormer variation, which does not use dynamic mixing or pretraining.

When we apply the TTR loss to finetune the baseline models, the proposed regularization introduces performance improvements in all task variations. In Table 1, the boldface denotes the best results per subtask. Specifically, Conv-TasNet+TTR outperforms its baseline in SI-SDR by 0.31 dB in Libri2Mix-clean and 0.19 dB in Libri2Mix-noisy tasks. As for three-source mixtures, the proposed model surpasses the baseline by 0.43 dB for Libri3Mix-clean and 0.49 dB for Libri3Mix-noisy subsets, respectively. Investigating SepFormer's performance, TTR successfully boosts SI-SDR by 1.47 dB in Libri2Mix clean separation task and by 0.83 dB in Libri2Mix noisy task. A similar trend can also be found in Libri3Mix tasks where SI-SDR scores was improved by 1.15 dB in Libri3Mix-clean and 1.18 dB in Libri3Mix-noisy, respectively.

Table 1: *Source separation performance. SDR and SI-SDR improvements are reported in decibel (dB), while STOI values range between 0 and 1, where 1 is the upper bound.*

Task	Model (λ)	SDRi	SI-SDRi	STOI
Libri2Mix Clean	Conv-TasNet (N/A)	15.11	14.76	0.9311
	+ TTR (1.0)	15.42	15.07	0.9342
	+ TTR (0.5)	15.42	15.08	0.9341
	+ TTR (0.1)	**15.44**	**15.10**	**0.9344**
	SepFormer (N/A)	18.68	18.35	0.9574
	+ TTR (1.0)	20.12	19.82	0.9682
	+ TTR (0.5)	20.14	19.85	0.9681
	+ TTR (0.1)	**20.17**	**19.87**	**0.9685**
Libri2Mix Noisy	Conv-TasNet (N/A)	12.36	11.80	0.8490
	+ TTR (1.0)	**12.54**	**11.99**	**0.8540**
	+ TTR (0.5)	12.29	11.74	0.8482
	+ TTR (0.1)	12.47	11.90	0.8512
	SepFormer (N/A)	15.11	14.54	0.8949
	+ TTR (1.0)	15.99	15.37	**0.9103**
	+ TTR (0.5)	**16.00**	**15.39**	0.9100
	+ TTR (0.1)	15.98	15.36	0.9096
Libri3Mix Clean	Conv-TasNet (N/A)	12.40	11.98	0.8365
	+ TTR (1.0)	**12.82**	**12.41**	**0.8448**
	+ TTR (0.5)	12.76	12.35	0.8439
	+ TTR (0.1)	12.77	12.35	0.8438
	SepFormer (N/A)	17.26	16.91	0.9141
	+ TTR (1.0)	18.43	18.06	0.9289
	+ TTR (0.5)	**18.45**	**18.09**	**0.9292**
	+ TTR (0.1)	18.39	18.03	0.9291
Libri3Mix Noisy	Conv-TasNet (N/A)	10.93	10.39	0.7669
	+ TTR (1.0)	**11.41**	**10.88**	**0.7793**
	+ TTR (0.5)	11.37	10.84	0.7776
	+ TTR (0.1)	11.35	10.81	0.7769
	SepFormer (N/A)	14.73	14.24	0.8489
	+ TTR (1.0)	**15.94**	**15.42**	**0.8727**
	+ TTR (0.5)	15.88	15.36	0.8717
	+ TTR (0.1)	15.91	15.39	0.8720

Since all SDR and STOI scores follow a similar trend as SI-SDR, we draw the conclusion that the TTR-SS models are able to produce separated sources with higher quality and intelligibility. Additionally, it is evident that the performance enhancement derived from the SepFormer baselines surpasses that from Conv-TasNet, suggesting that the proposed regularization provides more information to learn for larger models.

5. Conclusion

We presented a novel timed text-based regularization method that leverages sentence-level semantics from a language model, enhancing speech separation performance across diverse speaker and noise environments. For this purpose, we introduced the SLA and summarizer Transformer to align and minimize the gap between different modalities, i.e. audio and text. In our experiment, we demonstrated that the SLA and summarizer Transformer effectively differentiate between mixtures and clean sources by comparing them with their respective textual representations, indicating a meaningful regularization. Consequently, TTR enhances all evaluation metrics, particularly for SepFormer, a more complex and sizable model, at higher SNR levels. This underscores the efficacy of the proposed TTR method, which works with the conventional PIT loss, thus improving source separation tasks. The benefit comes with no additional computational or data collection cost during the test time due to its regularization-based approach.

6. Acknowledgement

This material is based on work supported in part by the National Science Foundation under Grant No. 2046963. The authors appreciate Dr. Cem Sübakan for providing invaluable information regarding the development of the baseline models. Part of the work was done when the authors were with Indiana University.

7. References

[1] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 246–250.

[2] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[3] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[4] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 334–340.

[5] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," *arXiv preprint arXiv:1911.13254*, 2019.

[6] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm -rf: Efficient Networks for Universal Audio Source Separation," in *Proc. of the IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, 2020, pp. 1–6.

[7] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[8] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[9] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[10] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020.

[11] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 21–25.

[12] C. Subakan, M. Ravanelli, S. Cornell, F. Grondin, and M. Bronzi, "Exploring self-attention mechanisms for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2169–2180, 2023.

[13] K. Schulze-Forster, C. S. Doire, G. Richard, and R. Badeau, "Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2382–2395, 2021.

[14] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Joint Phoneme Alignment and Text-Informed Speech Separation on Highly Corrupted Speech," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 7274–7278.

[15] K. Kilgour, B. Gfeller, Q. Huang, A. Jansen, A. Wisdom, and M. Tagliasacchi, "Text-Driven Separation of Arbitrary Sounds," in *Proc. Interspeech*, 2022, pp. 5403–5407.

[16] X. Liu *et al.*, "Separate What You Describe: Language-Queried Audio Source Separation," in *Proc. Interspeech*, 2022.

[17] H. W. Dong, N. Takahashi, Y. Mitsufuji, J. McAuley, and T. Berg-Kirkpatrick, "CLIPSep: Learning Text-queried Sound Separation with Noisy Unlabeled Videos," *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

[18] E. Tzinis, G. Wichern, A. S. Subramanian, P. Smaragdis, and J. Le Roux, "Heterogeneous Target Speech Separation," in *Proc. Interspeech*, 2022, pp. 1796–1800.

[19] S. Shon, H. Tang, and J. Glass, "VoiceID Loss: Speech Enhancement for Speaker Verification," in *Proc. Interspeech*, 2019, pp. 2888–2892.

[20] S. Mun, D. Gowda, J. Lee, C. Han, D. Lee, and C. Kim, "Prototypical speaker-interference loss for target voice separation using non-parallel audio samples," in *Proc. Interspeech*, 2022, pp. 276–280.

[21] T. von Neumann, K. Kinoshita, L. Drude, C. Boeddeker, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, "End-to-end training of time domain audio separation and recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 7004–7008.

[22] J. Shi, X. Chang, S. Watanabe, and B. Xu, "Train from scratch: Single-stage joint training of speech separation and recognition," *Computer Speech & Language*, vol. 76, p. 101387, 2022.

[23] Y. Koizumi, S. Karita, A. Narayanan, S. Panchapagesan, and M. A. U. Bacchiani, "SNRi target training for joint speech enhancement and recognition," in *Proc. Interspeech*, 2022.

[24] D. Yu, M. Kolbæk, Z. H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[25] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2019, pp. 4171–4186.

[26] S. Chen *et al.*, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[27] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[28] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation," *arXiv preprint arXiv:2005.11262*, 2020.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[30] G. Wichern *et al.*, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. Interspeech*, 2019, pp. 1368–1372.

[31] E. Vincent, C. Fevotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[32] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[33] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

[34] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.