2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) | 979-8-3503-2372-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/WASPAA58266.2023.10248168

TIME-DOMAIN AUDIO SOURCE SEPARATION BASED ON GAUSSIAN PROCESSES WITH DEEP KERNEL LEARNING

Aditya Arie Nugraha[1,2] *Diego Di Carlo*[1,2] *Yoshiaki Bando*[3,1] *Mathieu Fontaine*[4,1] *Kazuyoshi Yoshii*[2,1]

[1] Center for Advanced Intelligence Project (AIP), RIKEN, Japan
[2] Graduate School of Informatics, Kyoto University, Japan
[3] National Institute of Advanced Industrial Science and Technology (AIST), Japan
[4] LTCI, Télécom Paris, Institut Polytechnique de Paris, France

ABSTRACT

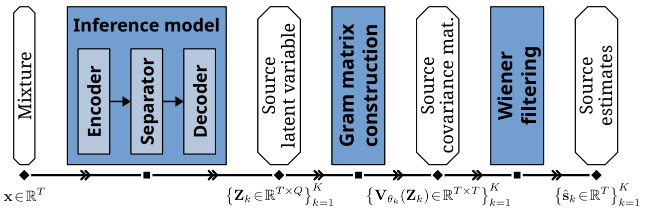
This paper revisits single-channel audio source separation based on a probabilistic generative model of a mixture signal defined in the continuous time domain. We assume that each source signal follows a non-stationary Gaussian process (GP), i.e., any finite set of sampled points follows a zero-mean multivariate Gaussian distribution whose covariance matrix is governed by a kernel function over time-varying latent variables. The mixture signal composed of such source signals thus follows a GP whose covariance matrix is given by the sum of the source covariance matrices. To estimate the latent variables from the mixture signal, we use a deep neural network with an encoder-separator-decoder architecture (e.g., Conv-TasNet) that separates the latent variables in a pseudo-time-frequency space. The key feature of our method is to feed the latent variables into the kernel function for estimating the source covariance matrices, instead of using the decoder for directly estimating the time-domain source signals. This enables the decomposition of a mixture signal into the source signals with a classical yet powerful Wiener filter that considers the full covariance structure over all samples. The kernel function and the network are trained jointly in the maximum likelihood framework. Comparative experiments using two-speech mixtures under clean, noisy, and noisy-reverberant conditions from the WSJ0-2mix, WHAM!, and WHAMR! benchmark datasets demonstrated that the proposed method performed well and outperformed the baseline method under noisy and noisy-reverberant conditions.

***Index Terms*—** Time-domain audio source separation, Gaussian processes, deep kernel learning

1. INTRODUCTION

Audio source separation extracts audio signals of interest or removes unwanted signals from recordings [1, 2]. It is invaluable in many applications, including speech enhancement [3], automatic speech recognition [4], music separation [5], and sound event detection [6].

One of the most modern approaches to single-channel audio source separation is to use a deep neural network (DNN) that works in the time domain, e.g., TasNet [7], Conv-TasNet [8], SuDoRM-RF [9], SuDoRM-RF++ [10], SepFormer [11, 12], MossFormer [13], and WaveFormer [14]. Most such time-domain separation methods are based on the encoder-separator-decoder architecture that performs mask-based source separation in a pseudo-time-frequency (pseudo-TF) space. More specifically, the *encoder* transforms a time-domain mixture signal into a spectrogram-like mixture representation, the *separator* estimates the spectrogram-like source representations, and the *decoder* recovers the time-domain source signals. All the networks are trained jointly in a fully data-driven manner, encouraging the learned pseudo-TF space to be optimal for source separation. To further improve the performance, various architectures for the separator [7–11, 13, 14] and various filterbanks for the encoder and decoder [15] have been investigated.


Mixture
Inference model
Encoder
Separator
Decoder
Source latent variable
Gram matrix construction
Source covariance mat.
Wiener filtering
Source estimates
$\mathbf{x} \in \mathbb{R}^T$
$\{\mathbf{Z}_k \in \mathbb{R}^{T\times Q}\}_{k=1}^K$
$\{\mathbf{V}_{\theta_k}(\mathbf{Z}_k) \in \mathbb{R}^{T\times T}\}_{k=1}^K$
$\{\hat{\mathbf{s}}_k \in \mathbb{R}^T\}_{k=1}^K$
Figure 1: The proposed Gaussian process-based single-channel audio source separation method based on time-domain Wiener filtering with deep kernel learning.

In this paper, we revisit the classical approach to audio source separation based on the Gaussian process (GP) [16]. If source signals follow independent GPs meaning that any finite set of sampled points is normally distributed, the mixture signal given as the sum of these signals follows a GP whose covariance function is given as the sum of the source covariance functions. Source separation is then recast as posterior inference of the latent sources from the observed mixture with Wiener filtering [17, 18]. Although this model is strictly formulated in the continuous time domain, most conventional methods perform Wiener filtering frame-wise, assuming the local stationarity and the inter-frame independence [19]. Specifically, the magnitude spectrogram of the mixture signal is decomposed while keeping the phase spectrogram untouched [1]. The performance of this strategy, however, is affected by the time and frequency resolutions. In addition, the possibly incompatible phase information causes unpleasant artifacts in reconstructed time-domain signals. To the best of our knowledge, a method based on variational sparse GPs [20] is the only method that operates entirely in the time domain.

Recent studies on GPs have focused on integrating deep learning methods into the probabilistic framework [21, 22]. One can use the covariance function of a GP for estimating the uncertainty to improve the robustness and interpretability, as in Bayesian neural networks [23]. Alternatively, one can parameterize the covariance function of a GP with a DNN in the framework of deep kernel learning (DKL) [24–26]. For example, a DNN was used for learning a low-dimensional representation of the input on which the kernel function works as a similarity measure [24]. Despite the theoretical support, these models tend to be hard to train in practice [22]. To mitigate this difficulty, heuristics (e.g., pre-training) need to be considered [24].

This work was supported by ANR Project SAROUMANE (ANR-22-CE23-0011), JST PRESTO no. JPMJPR20CB, and JSPS KAKENHI nos. 20H00602, 21H03572, 23K16912, and 23K16913.

In this paper, we propose a GP-based audio source separation method that uses a time-domain Wiener filter parameterized by a DNN for inferring source signals from a mixture signal at once (Figure 1). We assume that the non-stationarity of each source signal is governed by a time-domain sequence of latent variables. The Gram matrix of the source GP over all sampled points is computed with a kernel function over the latent variables, which are estimated from the mixture signal with an encoder-separator-decoder network, e.g., Conv-TasNet [8] and SepFormer [11]. The kernel function (GP-based generative model) and the network (DKL-based inference model) are trained jointly in the maximum likelihood framework. This is the first attempt to combine DKL with full-covariance Wiener filtering for GP-based source separation. We show that the proposed approach achieved good speech separation performances under clean, noisy, and noisy-reverberant conditions with the popular WSJ0-2mix [27], WHAM! [28], and WHAMR! [29] benchmark datasets, respectively.

2. RELATED WORK

Let $\mathbf{x} \triangleq [x_1, \ldots, x_T]^\mathsf{T} \in \mathbb{R}^T$ be a series of T sampled points from a mixture signal and $\mathbf{s}_k \triangleq [s_{k1}, \ldots, s_{kT}]^\mathsf{T} \in \mathbb{R}^T$ be that from the signal of source $k \in \{1, \ldots, K\}$, where K is the number of sources. The goal of source separation is to recover source signals $\{\mathbf{s}_k\}_{k=1}^K$ given the observed mixture signal $\mathbf{x}$.

2.1. Neural time-domain audio source separation

Time-domain audio source separation methods are capable of estimating the source signal $\mathbf{s}_k$ given the mixture signal $\mathbf{x}$. The main advantage of this approach is that it can circumvent the phase estimation, which has still been a challenging problem for separation methods that estimate masks for magnitude spectrograms or phase-aware masks for complex-valued spectrograms [1].

For this purpose, deep learning techniques have been used. In the encoder-separator-decoder approach [7–11, 13, 14], the encoder transforms $\mathbf{x}$ into a pseudo-TF representation from which source representations are extracted by the separator and transformed to $\mathbf{s}_k$ using the decoder. The separation can be generally expressed as

$$\{\hat{\mathbf{s}}_k\}_{k=1}^K \leftarrow \mathrm{Dec}\left(\mathrm{Sep}\left(\mathrm{Enc}\left(\mathbf{x}\right)\right)\right). \tag{1}$$

Strictly speaking, the separation is not done in the time domain but in a learned pseudo-TF space that encapsulates the signal features with phase information and is optimized for the training data. The optimizable filterbanks [15] used in the encoder and decoder are trained jointly with the separator that incorporates a powerful DNN, e.g., the long short-term memory network [7], the temporal convolutional network [8], the multi-resolution convolutional network [9, 10], and the transformer network [11, 13, 14].

2.2. Gaussian process regression with deep kernel learning

Let f be a continuous function over a space $\mathcal{Y}$. If f follows a Gaussian process (GP) [16], any finite set of T sampled points denoted by $\mathbf{f} \triangleq \{f(\mathbf{y}_t)\}_{t=1}^T$ given $\mathbf{Y} \triangleq \{\mathbf{y}_t \in \mathcal{Y}\}_{t=1}^T$ follows a multivariate Gaussian distribution. Its probability density function is given by $p_\Theta(\mathbf{f}|\mathbf{Y}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{V}_\Theta(\mathbf{Y}))$, where $\boldsymbol{\mu} \in \mathbb{R}^T$ is a mean vector and $\mathbf{V}_\Theta(\mathbf{Y}) \in \mathbb{R}^{T\times T}$ is a covariance matrix whose elements are computed with a kernel function $k_\Theta(\cdot,\cdot)$ parameterized by Θ as

$$\{\mathbf{V}_\Theta(\mathbf{Y})\}_{tt'} = k_\Theta(\mathbf{y}_t, \mathbf{y}_{t'}), \tag{2}$$

where, in this case, $t, t' \in \{1, \ldots, T\}$ serve as the row and column indices, respectively. Given another set of T^* sampled points denoted by $\mathbf{Y}^*$, GP regression [16] estimates the corresponding $\mathbf{f}^*$ via the posterior distribution of $\mathbf{f}^*$ given $\mathbf{f}$, $\mathbf{Y}$, and $\mathbf{Y}^*$.

Deep kernel learning (DKL) [24] exploits the expressive power of DNNs in constructing $\{\mathbf{V}_\Theta(\mathbf{Y})\}_{tt'}$ via a non-linear mapping $g_\Phi(\cdot)$, where Φ denotes DNN parameters, given a kernel $k_\Theta(\cdot,\cdot)$ as

$$\{\mathbf{V}_\Theta(\mathbf{Y})\}_{tt'} = k_\Theta(g_\Phi(\mathbf{y}_t), g_\Phi(\mathbf{y}_{t'})). \tag{3}$$

3. PROPOSED METHOD

This section describes the proposed method based on a latent variable model (Figure 1). It performs source separation in a pseudo-TF domain (latent variable estimation) with an inference model, computes the Gram matrices of source GPs, and finally estimates source signals with a Wiener filter. Let $\mathbf{Z}_k = [\mathbf{z}_{k1}, \ldots, \mathbf{z}_{kT}]^\mathsf{T} \in \mathbb{R}^{T\times Q}$ be a series of latent variables of source $k \in \{1, \ldots, K\}$, where $\mathbf{z}_{kt} \in \mathbb{R}^Q$.

3.1. Source separation based on Gaussian processes

We assume that each source signal is drawn from a GP in the continuous time domain and thus any finite set of T sampled points, $\mathbf{s}_k$, follows a real multivariate Gaussian distribution with a zero-mean vector and a covariance matrix governed by a source-specific kernel function $\mathbf{V}_{\Theta_k}$ with parameters Θ_k as follows:

$$p_{\Theta_k}\left(\mathbf{s}_k \middle| \mathbf{Z}_k\right) = \mathcal{N}\left(\mathbf{s}_k \middle| \mathbf{0}, \mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\right). \tag{4}$$

We also consider the presence of white noise. Let $\boldsymbol{\epsilon} \in \mathbb{R}^T$ be a series of T sampled points from white noise, which is expressed as follows:

$$p(\boldsymbol{\epsilon}) = \mathcal{N}\left(\boldsymbol{\epsilon} \middle| \mathbf{0}, \lambda\mathbf{I}\right), \tag{5}$$

where $\lambda \in \mathbb{R}_+$ is the noise variance (regularization parameter). Assuming the signal additivity, the mixture $\mathbf{x} = \sum_{k=1}^K \mathbf{s}_k + \boldsymbol{\epsilon}$ can be said to follow a multivariate Gaussian distribution as follows:

$$p_{\Theta,\lambda}\left(\mathbf{x} \middle| \mathbf{Z}\right) = \mathcal{N}\left(\mathbf{x} \middle| \mathbf{0}, \mathbf{V}_\Theta(\mathbf{Z}) + \lambda\mathbf{I}\right), \tag{6}$$

where $\mathbf{V}_\Theta(\mathbf{Z}) \triangleq \sum_{k=1}^K \mathbf{V}_{\Theta_k}(\mathbf{Z}_k)$ with $\Theta \triangleq \{\Theta_k\}_{k=1}^K$ and $\mathbf{Z} \triangleq \{\mathbf{Z}_k\}_{k=1}^K$. The posterior distribution of $\mathbf{s}_k$ given $\mathbf{Z}$ and $\mathbf{x}$ is given by

$$p_{\Theta,\lambda}\left(\mathbf{s}_k \middle| \mathbf{Z}, \mathbf{x}\right) = \mathcal{N}\left(\mathbf{s}_k \middle| \boldsymbol{\mu}_k^\mathbf{s}, \boldsymbol{\Sigma}_k^\mathbf{s}\right), \tag{7}$$

$$\boldsymbol{\mu}_k^\mathbf{s} = \mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\left(\mathbf{V}_\Theta(\mathbf{Z}) + \lambda\mathbf{I}\right)^{-1}\mathbf{x}, \tag{8}$$

$$\boldsymbol{\Sigma}_k^\mathbf{s} = \mathbf{V}_{\Theta_k}(\mathbf{Z}_k) - \mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\left(\mathbf{V}_\Theta(\mathbf{Z}) + \lambda\mathbf{I}\right)^{-1}\mathbf{V}_{\Theta_k}(\mathbf{Z}_k), \tag{9}$$

where the posterior mean is considered as the estimated source, i.e., $\hat{\mathbf{s}}_k \triangleq \boldsymbol{\mu}_k^\mathbf{s}$. Equation (8) is also known as Wiener filtering whose filter $\mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\left(\mathbf{V}_\Theta(\mathbf{Z}) + \lambda\mathbf{I}\right)^{-1}$ requires estimation of source latent variables $\mathbf{Z}$ to obtain the covariance matrices $\{\mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\}_{k=1}^K$.

3.2. Estimation of latent variables and covariance matrices

We estimate $\mathbf{Z}$ given $\mathbf{x}$ using a DNN-based inference model with an encoder-separator-decoder architecture. While the decoder was originally used for estimating time-domain source signals from the pseudo-TF source representations obtained by the separator (Section 2.1), our decoder estimates the time-domain latent variables $\mathbf{Z}$ from the frame-varying pseudo-TF representations as follows:

$$\mathbf{Z} \leftarrow \mathrm{Infer}_\Phi\left(\mathbf{x}\right) = \mathrm{Dec}_{\phi^\mathrm{D}}\left(\mathrm{Sep}_{\phi^\mathrm{S}}\left(\mathrm{Enc}_{\phi^\mathrm{E}}\left(\mathbf{x}\right)\right)\right), \tag{10}$$

where $\Phi \triangleq \{\phi^\mathrm{D}, \phi^\mathrm{S}, \phi^\mathrm{E}\}$ denotes the inference model parameters that gather the encoder parameters ϕ^E, the separator parameters ϕ^S, and the source-specific decoder parameters $\phi^\mathrm{D} \triangleq \{\phi_k^\mathrm{D}\}_{k=1}^K$. The decoder has a 1-dimensional convolutional layer as in most vanilla decoders except that it returns a matrix of $\mathbb{R}^{T\times Q}$, not a vector of $\mathbb{R}^T$.

We then compute a covariance matrix $\mathbf{V}_{\Theta_k}(\mathbf{Z}_k)$ using a non-stationary, non-degenerate composite kernel resulting from multiplying the linear kernel and the squared-exponential kernel [16]:

$$\left\{\mathbf{V}_{\Theta_k}(\mathbf{Z}_k)\right\}_{tt'} = \omega_{k0}\delta_{tt'} + \omega_{k1}\mathbf{z}_{kt}^\mathsf{T}\mathbf{z}_{kt'}e^{-\theta_k\left\|\mathbf{z}_{kt} - \mathbf{z}_{kt'}\right\|^2}, \tag{11}$$

where $\Theta_k \triangleq \{\omega_{k0}, \omega_{k1}, \theta_k\} \in \mathbb{R}_+^3$ gathers the kernel parameters.

3.3. Parameter optimization

The inference model Φ, the kernel parameters Θ, and the noise parameter λ are optimized, in principle, such that the log-likelihood of the sources given the mixture and latent variables, $\ln p_{\Theta,\lambda}(\mathbf{s}|\mathbf{Z}, \mathbf{x})$, is maximized. This is performed by gradient descent with backpropagation using permutation-invariant training (PIT) [30]. For better optimization, we opt for a two-stage training approach.

In the first stage, we train a vanilla encoder-separator-decoder model, e.g., Conv-TasNet [8], by minimizing the negative scale-invariant signal-to-distortion ratio (SI-SDR) [31] given by

$$\mathcal{L}^{\text{SI-SDR}} \triangleq -\sum_{k=1}^{K} 10\log_{10}\left(\frac{\alpha^2 \mathbf{s}_k^\mathsf{T}\mathbf{s}_k}{(\alpha\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})^\mathsf{T}(\alpha\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})}\right), \quad (12)$$

where, in this stage, the estimated sources $\{\boldsymbol{\mu}_k^\mathbf{s}\}_{k=1}^K$ are the output of the vanilla decoder and $\alpha = (\boldsymbol{\mu}_k^\mathbf{s})^\mathsf{T}\mathbf{s}_k(\mathbf{s}_k^\mathsf{T}\mathbf{s}_k)^{-1}$ is the optimal scaling factor that minimizes $(\alpha\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})^\mathsf{T}(\alpha\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})$.

In the second stage, we construct an inference model by combining the encoder and separator of the pre-trained vanilla model with our decoder. Although fine-tuning the parameters ϕ^E and ϕ^S may provide some improvement, it is not easy to effectively configure the fine-tuning procedure. As a preliminary attempt, we opt to freeze ϕ^E and ϕ^S in this paper and only optimize ϕ^D together with Θ and λ by minimizing the negative log-likelihood (NLL) given by

$$\begin{aligned}\mathcal{L}^{\text{NLL}} &\triangleq -\ln p_{\Theta,\lambda}(\mathbf{s}|\mathbf{Z},\mathbf{x}) = -\sum_{k=1}^{K}\ln p_{\Theta,\lambda}(\mathbf{s}_k|\mathbf{Z},\mathbf{x}) \\ &= \frac{1}{2}\sum_{k=1}^{K}\left(\left(\mathbf{L}_k^{\boldsymbol{\Sigma}}\right)^{-1}(\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})\right)^\mathsf{T}\left(\left(\mathbf{L}_k^{\boldsymbol{\Sigma}}\right)^{-1}(\mathbf{s}_k - \boldsymbol{\mu}_k^\mathbf{s})\right) \\ &\quad + \sum_{k=1}^{K}\sum_{t=1}^{T}\ln\left\{\mathbf{L}_k^{\boldsymbol{\Sigma}}\right\}_{tt} + \frac{KT}{2}\ln 2\pi, \end{aligned} \quad (13)$$

where $\mathbf{L}_k^{\boldsymbol{\Sigma}}$ is obtained by the Cholesky decomposition for regularized $\boldsymbol{\Sigma}_k^\mathbf{s}$, i.e., $\boldsymbol{\Sigma}_k^\mathbf{s} + \frac{\varepsilon}{L}\text{Tr}(\boldsymbol{\Sigma}_k^\mathbf{s})\mathbf{I} = \mathbf{L}_k^{\boldsymbol{\Sigma}}\left(\mathbf{L}_k^{\boldsymbol{\Sigma}}\right)^\mathsf{T}$ with ε working to ensure the positive definiteness of the left-hand side. In this paper, we set the default value to $\varepsilon = 10^{-6}$, but when it is needed, we let it be larger to allow a successful decomposition of a particular $\boldsymbol{\Sigma}_k^\mathbf{s}$. Instead of $\mathcal{L}^{\text{NLL}}$, $\mathcal{L}^{\text{SI-SDR}}$ can also be used in this stage.

3.4. Dimensionality reduction

Our model performs source separation based on the covariance matrix $\mathbf{V}_{\Theta_k}(\mathbf{Z}_k) \in \mathbb{R}^{T\times T}$, which is computationally prohibitive in practice. For example, a 1-s signal sampled at 8 kHz has a covariance matrix of size (8000×8000). To deal with it, we perform partitioning during both the training phase and the test phase.

In the training phase, the latent variables $\{\mathbf{Z}_k \in \mathbb{R}^{T\times Q}\}_{k=1}^K$ are estimated once given $\mathbf{x} \in \mathbb{R}^T$. Each source latent variable $\mathbf{Z}_k$ is partitioned into non-overlapping L segments $\{\mathbf{Z}_{kl} \in \mathbb{R}^{T'\times Q}\}_{l=1}^L$, where $T' \ll T$ and l is the segment index. In this paper, we set $T' = 1600$ (200 ms at 8 kHz). The covariance matrix construction and source separation are then performed given these segmented latent variables. $\mathcal{L}^{\text{NLL}}$ is computed segment-wise, whereas $\mathcal{L}^{\text{SI-SDR}}$ is calculated after concatenating the separated source segments.

In the test phase, the covariance matrix construction and source separation are performed given *possibly* overlapping latent variable segments. The final separated sources are obtained by simple concatenation of separated source segments (for non-overlapping segments) or by the overlap-add (OLA) technique [32] with a Hann weighting window (for overlapping segments). In this paper, we use segments with an overlap of size $T'/2 = 800$ (i.e., 100 ms at 8 kHz).

4. EVALUATION

We considered single-channel separations of two speech signals from mixtures under clean (ideal), noisy, and noisy-reverberant conditions. We used the 'min' variants of the WSJ0-2mix dataset [27] for the clean condition, the WHAM! dataset [28] for noisy conditions, and the WHAMR! dataset [29] for noisy-reverberant conditions. Each dataset has training, validation, and test sets of 20 000, 5000, and 3000 mixtures, respectively. All data are sampled at 8 kHz.

For clean speech mixtures, we set all separation models to output the two speech signals ($K = 2$). For noisy or noisy-reverberant mixtures, all models were set to additionally output the residual signal, corresponding to the noise component or the noise-and-reverberation component, respectively ($K = 3$). Nonetheless, the performance assessment took into account only the estimated speech signals.

Speech separation performance was assessed in terms of *improvements* of the SI-SDR (SI-SDRi), the perceptual evaluation of speech quality (PESQi), and the short-time objective intelligibility (STOIi) [31, 33, 34]. A permutation solver that maximizes SI-SDR (as in PIT [30]) was used to decide the best source ordering.

4.1. Configurations

We considered a non-causal *Conv-TasNet* [8] based on the Asteroid library [35] or a non-causal *SepFormer* [11, 12] based on the SpeechBrain library [36] as the baseline model.[1] We trained a baseline model with a batch of 16 (Conv-TasNet) or 4 (SepFormer) 4-s segments ($T = 32000$) for 300 epochs on the original training dataset (without data augmentation as in, e.g., [9, 10]). The learning rate of the Adam optimizer [37] was initially set to 10^{-3} (Conv-TasNet) or 10^{-4} (SepFormer) and halved when the validation loss did not improve after 5 consecutive epochs. A norm-based gradient clipping [38] with a threshold of 5 was applied.

Our GP-based model, *Conv-TasNet+GP* or *SepFormer+GP*, was built utilizing the parameters of encoder ϕ^E and separator ϕ^S of Conv-TasNet or SepFormer, respectively, trained for 200 epochs, at which the SI-SDRi scores have stopped improving, and were virtually the same after 300 epochs (cf. Table 1). We substituted the original decoder with our decoder that outputs 8-dimensional source latent variables ($Q = 8$). The parameters of our decoder ϕ^D were initialized using random semi-orthogonal matrices [39], while the other parameters were initialized as $\omega_{k0} = 10^{-2}$, $\omega_{k1} \leftarrow \mathcal{N}(1, 10^{-4})$, $\theta_k \leftarrow \mathcal{N}(1, 10^{-4})$, and $\lambda = 10^{-2}$. The training configuration of a GP-based model was the same as that of the vanilla model, except that it was trained with a batch of 16 4-s segments for 100 epochs. If we take into account the pre-training of the vanilla model, the GP-based model was trained for 300 epochs in total. Based on a grid-search-based hyperparameter tuning, we found that the generally-optimal initial learning rate for $\mathcal{L}^{\text{NLL}}$ was 2×10^{-5}, while that for $\mathcal{L}^{\text{SI-SDR}}$ was 5×10^{-5} (ConvTasNet+GP) or 5×10^{-6} (SepFormer+GP).

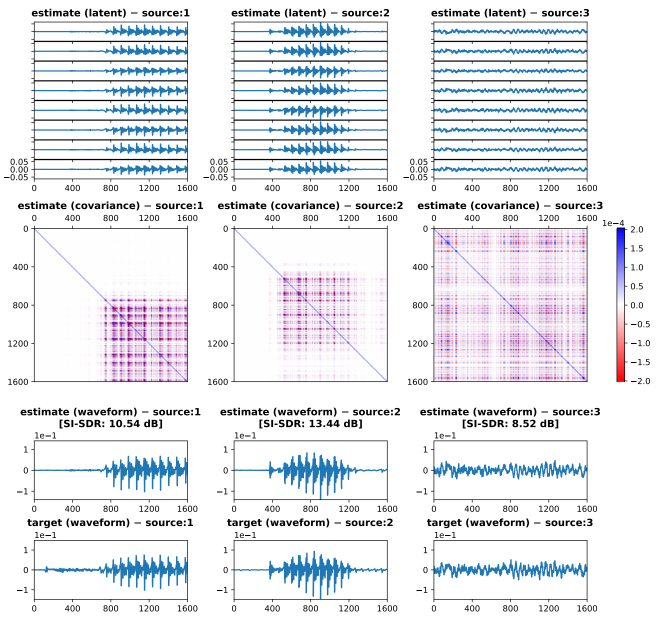
4.2. Results and discussion

Table 1 compares the performance of different models in terms of SI-SDRi, PESQi, and STOIi scores. These scores demonstrate that

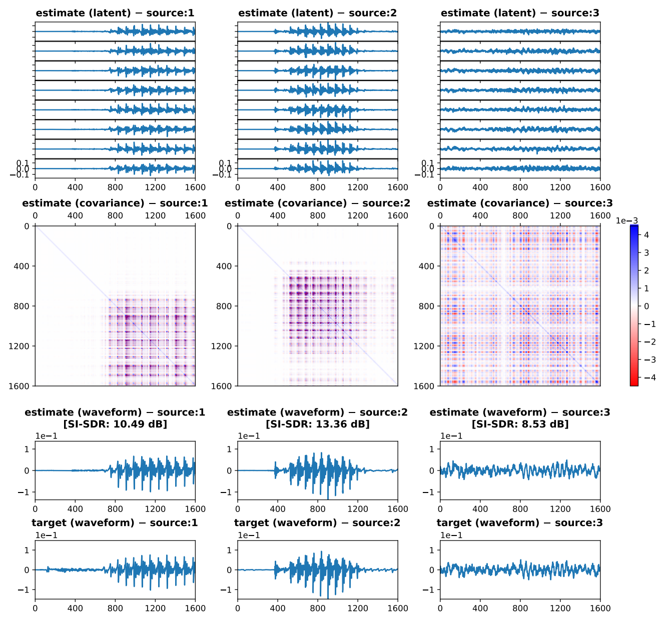
[1]The total number of parameters for WSJ0-2mix ($K = 2$) was 5.05 M (Conv-TasNet), 5.17 M (Conv-TasNet+GP), 25.68 M (SepFormer), or 25.81 M (SepFormer+GP), whereas the total number of parameters for WHAM! or WHAMR! ($K = 3$) was 5.12 M (Conv-TasNet), 5.31 M (Conv-TasNet+GP), 25.75 M (SepFormer), or 25.94 M (SepFormer+GP).

Table 1: Average speech separation performance scores of the different models on the test set. Higher is better for all metrics. OLA denotes the overlap-add operation required for separation with overlapping segments. Boldface numbers show the top performances taking into account the 95% confidence interval over the best performances (indicated by *) in each group separated based on the baseline model (shown in italics).

Model	Loss function	OLA	SI-SDRi (dB)			PESQi			STOIi		
			WSJ0-2mix	WHAM!	WHAMR!	WSJ0-2mix	WHAM!	WHAMR!	WSJ0-2mix	WHAM!	WHAMR!
Conv-TasNet [8]	$\mathcal{L}^{\text{SI-SDR}}$	n/a	**15.8**	13.3	10.6	**1.55**	0.71	0.45	0.219	0.240	0.214
Conv-TasNet+GP	$\mathcal{L}^{\text{NLL}}$	✗	**16.0**	13.6	10.7	1.52	0.76	0.51	**0.224**	0.248	0.221
Conv-TasNet+GP	$\mathcal{L}^{\text{NLL}}$	✓	**16.0***	**13.7**	**10.9**	1.54	**0.79**	**0.53***	**0.226***	0.250	**0.224**
Conv-TasNet+GP	$\mathcal{L}^{\text{SI-SDR}}$	✗	15.4	**13.7**	**10.9**	**1.56**	0.79	0.51	**0.224**	**0.251**	**0.224**
Conv-TasNet+GP	$\mathcal{L}^{\text{SI-SDR}}$	✓	15.5	**13.9***	**11.0***	**1.58***	**0.81***	**0.53**	**0.225**	**0.253***	**0.227***
SepFormer [11]	$\mathcal{L}^{\text{SI-SDR}}$	n/a	**19.8***	**15.5***	**13.0***	**2.03**	0.99	0.68	**0.241**	0.281	0.265
SepFormer+GP	$\mathcal{L}^{\text{NLL}}$	✓	19.2	**15.3**	**12.8**	**1.91**	**1.05**	**0.77***	**0.241**	0.282	**0.268**
SepFormer+GP	$\mathcal{L}^{\text{SI-SDR}}$	✓	19.6	**15.5**	**12.9**	**2.03***	**1.06***	**0.76**	**0.243***	**0.285***	**0.270***



(a) SepFormer+GP trained with $\mathcal{L}^{\text{NLL}}$



(b) SepFormer+GP trained with $\mathcal{L}^{\text{SI-SDR}}$

Figure 2: Separation examples of a mixture segment under a noisy-reverberant condition (of WHAMR! dataset) using SepFormer+GPs. For each subfigure, the columns from left to right show the first speech signal, the second speech signal, and the residual. The rows from top to bottom show latent variable estimates ($Q = 8$), covariance matrix estimates, time-domain waveform estimates, and time-domain waveform targets.

the proposed GP-based models outperformed the vanilla models, especially under the more challenging noisy (WHAM!) and noisy-reverberant (WHAMR!) conditions.[2] Although the SI-SDRi scores of SepFormer+GPs are not significantly different from those of SepFormer under these two conditions, the PESQi and STOIi scores imply that the pure time-domain modeling in the GP-based models, including SepFormer+GPs, improves perceptual quality, likely due to better phase consistency. These results also suggest that separation could benefit from the estimation of the residual covariance structure during the Wiener filter computation. Separation with overlapping segments (denoted by OLA) is shown to be useful probably by eliminating the boundary effect that causes the discontinued waveform in separation with non-overlapping segments. The GP-based models trained with $\mathcal{L}^{\text{SI-SDR}}$ generally performed better speech separation than those trained with $\mathcal{L}^{\text{NLL}}$. It may indicate that constraining the covariance matrices as in $\mathcal{L}^{\text{NLL}}$ makes optimization more challenging.

Figure 2 provides insight using examples of source latent variables, covariance matrices, and signals estimated using SepFormer+GPs. The estimated signals and the SI-SDR scores (shown within the figures) look similar. Although we may notice differences in the details, the estimated covariance matrices also show a similar pattern reflecting the temporal structure of the time domain signal.

5. CONCLUSION

This paper proposes a novel time-domain audio source separation based on Gaussian processes with deep kernel learning that effectively combines the expressive power of a DNN with the rigorous full-covariance Wiener filtering. With comparable numbers of parameters, the proposed GP-based models outperformed the corresponding vanilla models in speech separations under challenging noisy and noisy-reverberant conditions. Future work includes interpreting our approach within the variational framework and performing ablation studies to explore effective and efficient training procedures.

[2]Our baseline performance is generally higher than that in the literature. The reported SI-SDRi scores on WSJ0-2mix, WHAM!, and WHAMR! for Conv-TasNet are 15.3 dB, 12.7 dB, and 8.3 dB [8, 13], while those for SepFormer are 20.4 dB, 14.7 dB, and 11.4 dB [12]. This could be attributed to differences in the details of the network and training configurations.

6. REFERENCES

[1] E. Vincent, T. Virtanen, and S. Gannot, Eds., *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[2] S. Makino, Ed., *Audio Source Separation*. Springer, 2018.

[3] H. Dubey, *et al.*, "ICASSP 2022 Deep Noise Suppression Challenge," in *Proc. IEEE ICASSP*, 2022, pp. 9271–9275.

[4] S. Watanabe, *et al.*, "CHiME-6 Challenge: Tackling multi-speaker speech recognition for unsegmented recordings," in *Proc. INTERSPEECH*, 2020, pp. 1–7.

[5] Y. Mitsufuji, *et al.*, "Music Demixing Challenge 2021," *Front. Signal Process.*, vol. 1, no. 808395, pp. 1–14, 2022.

[6] N. Turpault, *et al.*, "Sound event detection and separation: A benchmark on DESED synthetic soundscapes," in *Proc. IEEE ICASSP*, 2021, pp. 840–844.

[7] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. IEEE ICASSP*, 2018, pp. 696–700.

[8] ——, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[9] E. Tzinis, Z. Wang, and P. Smaragdis, "SuDoRM-RF: Efficient networks for universal audio source separation," in *Proc. IEEE MLSP*, 2020, pp. 1–6.

[10] E. Tzinis, Z. Wang, X. Jiang, and P. Smaragdis, "Compute and memory efficient universal sound source separation," *J. Signal Process. Syst.*, vol. 94, no. 2, pp. 245–259, 2022.

[11] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE 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 Trans. Audio, Speech, Language Process.*, vol. 31, pp. 2169–2180, 2023.

[13] S. Zhao and B. Ma, "MossFormer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[14] B. Veluri, *et al.*, "Real-time target sound extraction," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[15] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," in *Proc. IEEE ICASSP*, 2020, pp. 6364–6368.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[17] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *IEEE Audio, Speech, Language Process.*, vol. 14, no. 1, pp. 191–199, 2005.

[18] A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3155–3167, 2011.

[19] E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, "Probabilistic modeling paradigms for audio source separation," in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. IGI Global, 2011.

[20] P. A. Alvarado, M. A. Alvarez, and D. Stowell, "Sparse Gaussian process audio source separation using spectrum priors in the time-domain," in *Proc. IEEE ICASSP*, 2019, pp. 995–999.

[21] T. Wang, L. Zhang, and W. Hu, "Bridging deep and multiple kernel learning: A review," *Inf. Fusion*, vol. 67, pp. 3–13, 2021.

[22] S. W. Ober, C. E. Rasmussen, and M. van der Wilk, "The promises and pitfalls of deep kernel learning," in *Proc. UAI*, 2021, pp. 1206–1216.

[23] E. Goan and C. Fookes, "Bayesian neural networks: An introduction and survey," *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pp. 45–87, 2020.

[24] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proc. AISTATS*, 2016, pp. 370–378.

[25] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold Gaussian processes for regression," in *Proc. IJCNN*, 2016, pp. 3338–3345.

[26] J. Bradshaw, A. G. de G. Matthews, and Z. Ghahramani, "Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks," pp. 1–12, 2017, arXiv:1707.02476v1.

[27] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE ICASSP*, 2016, pp. 31–35.

[28] G. Wichern, *et al.*, "WHAM!: Extending speech separation to noisy environments," in *Proc. INTERSPEECH*, 2019, pp. 1368–1372.

[29] M. Maciejewski, G. Wichern, E. McQuinn, and J. L. Roux, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. IEEE ICASSP*, 2020, pp. 696–700.

[30] 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. IEEE ICASSP*, 2017, pp. 241–245.

[31] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. IEEE ICASSP*, 2019, pp. 626–630.

[32] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, 1984.

[33] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ): A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE ICASSP*, vol. 2, 2001, pp. 749–752.

[34] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[35] M. Pariente, *et al.*, "Asteroid: The PyTorch-based audio source separation toolkit for researchers," in *Proc. INTERSPEECH*, 2020, pp. 2637–2641.

[36] M. Ravanelli, *et al.*, "SpeechBrain: A general-purpose speech toolkit," pp. 1–16, 2021, arXiv:2106.04624v1.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[38] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, 2013, pp. 1310–1318.

[39] A. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *Proc. ICLR*, 2014, pp. 1–15.