IEEE Open Journal of Signal Processing

Received 6 September 2023; revised 26 January 2024; accepted 28 January 2024. Date of publication 8 February 2024; date of current version 20 February 2024. The review of this article was arranged by Associate Editor H. V. Zhao.

Digital Object Identifier 10.1109/OJSP.2024.3363649

Short Papers

Attention-Based End-to-End Differentiable Particle Filter for Audio Speaker Tracking

JINZHENG ZHAO [1], YONG XU [2] (Senior Member, IEEE), XINYUAN QIAN [3] (Senior Member, IEEE), HAOHE LIU [1] (Student Member, IEEE), MARK D. PLUMBLEY [1] (Fellow, IEEE), AND WENWU WANG [1] (Senior Member, IEEE)

[1]Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, U.K.
[2]Tencent, AI Lab, Bellevue, WA 98004 USA
[3]Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing 100083, China

CORRESPONDING AUTHOR: JINZHENG ZHAO (e-mail: j.zhao@surrey.ac.uk).

The work of Mark D. Plumbley was supported by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/T019751/1 "AI for Sound". The work of Wenwu Wang was supported by UDRC SIGNeTS under Project W911NF-20-2-0225. The work of Xinyuan Qian was supported by CCF-Tencent Rhino-Bird Open Research Fund and National Natural Science Foundation of China under Grant 62306029. This work was supported in part by Tencent AI Lab Rhino-Bird Gift Fund and in part by the University of Surrey.

ABSTRACT Particle filters (PFs) have been widely used in speaker tracking due to their capability in modeling a non-linear process or a non-Gaussian environment. However, particle filters are limited by several issues. For example, pre-defined handcrafted measurements are often used which can limit the model performance. In addition, the transition and update models are often preset which make PF less flexible to be adapted to different scenarios. To address these issues, we propose an end-to-end differentiable particle filter framework by employing the multi-head attention to model the long-range dependencies. The proposed model employs the self-attention as the learned transition model and the cross-attention as the learned update model. To our knowledge, this is the first proposal of combining particle filter and transformer for speaker tracking, where the measurement extraction, transition and update steps are integrated into an end-to-end architecture. Experimental results show that the proposed model achieves superior performance over the recurrent baseline models.

INDEX TERMS Particle filter, differentiable particle filter, transformer, end to end training, speaker tracking.

I. INTRODUCTION

Speaker tracking plays an important role in speech separation [1], speech enhancement [2] and speaker diarization [3]. The task of speaker tracking is to estimate the 2D position, 3D position or Direction of Arrival (DOA) of speakers at each time step. Generally, speaker tracking consists of two steps, measurement extraction and Bayesian filtering. Speaker localization can provide measurements for speaker tracking. For speaker localization, there are two types of methods: parametric-based methods [4] and learning-based methods [5]. One of the important parametric-based methods is global coherent field (GCF), which is widely used for obtaining measurements in speaker tracking [6]. GCF map accumulates the generalized cross-correlation phase transform [4] (GCC-PHAT) generated by signals from each microphone pair. Then a grid search method is employed to find the maximum over the acoustic map. The position producing the maximum on the acoustic map is regarded as the position of the sound source. Compared to parametric-based methods, learning-based methods are more robust against room reverberation and background noise [7] when trained on audio data recorded from different acoustic environments. It finds the relationships between the audio features such as GCC-PHAT [8] and the speakers' positions through neural networks.

Tracking considers the temporal variations of a speaker trajectory. The tracking algorithms often focus on temporal variations, smoothed trajectories, removing estimation outliers and compensating for missing observations. The family of Bayesian filtering algorithms is often used to address these problems, which aims to estimate target states recurrently given the previous states and the current measurements. There are two recursive steps in the Bayesian filtering, namely, prediction and updating. In the prediction step, the target states are transferred from the last time step to the current time step through a transition model. In the update step,

the states are updated from prior to posterior by the measurement model. Several methods have been developed in this family, including Kalman Filter (KF) and PF. KF assumes the transition process and the update process to be linear and the noise to be following Gaussian distribution. It uses Gaussian distribution to represent the target states and updates the mean and covariance at each time step. The performance is satisfactory in this linear Gaussian environment but this assumption limits the generalization of this model to complicated scenarios. Extended KF (EKF) and unscented KF (UKF) are proposed to mitigate the linear-Gaussian limitations. EKF approximates the non-linear transition and update process using a first-order Taylor series expansion. UKF employs deterministic sampling to generate a set of sigma points to calculate the mean and covariance of state distribution. Particle filter (PF) is a Sequential Monte Carlo (SMC) method and uses a group of particles instead of Gaussian distributions to represent the target states, which can handle the non-linear and non-Gaussian scenarios. PF contains four steps. At first, the particles are initialized with the same weights. The particle states are transitioned in the prediction step, and the particle weights are updated by the measurement likelihood in the update step. The target states are calculated as the weighted sum of the particle states. At last, the particles are resampled to avoid the weight degeneracy problem. The particles with high weights are maintained and duplicated, while the particles with low weights are discarded.

The Bayesian filter is a first-order Markov process. The estimation of current states depends on the state in the last time and the measurements in the current time. Similarly, transformer is an auto-regressive model when used in temporal prediction tasks such as machine translation, text summarization and tracking, whose output at current time step also depends on the input and output in the previous time steps. Based on this similarity, we propose a new method by combining the two models with the following novel aspects. First, the transition model, which is used to change the particle states in a particle filter, is learned with a multi-head self-attention model, instead of being pre-defined as a constant velocity model in a conventional particle filter. Second, in the observation model, which aims to update the particle weights according to the measurement likelihood, we use the multi-head cross-attention to model the interaction between different modalities, in order to capture the relationship between the encoded audio embedding and particle embedding. The designed model combine the advantages of a particle filter and transformer. 1) The particle filter is not an end-to-end architecture. The measurement needs to be obtained before the update step. The combination is an end-to-end architecture where the filter and the measurement model can be trained jointly. 2) The transition model and update model are often preset, which are hard to generalize to complex scenarios. The self-attention and cross-attention modules are employed as learnable transition and update models. 3) It is proved that the algorithm priors introduced by particle filter will improve the model performance [9]. In addition, the prediction step and update step introduced in the transformer model bring explainability to the model, as compared to training in a black-box neural network model.

The remaining part of this paper is presented as follows: In Section II, we summarize some related topics. In Section III, we formulate the tracking problem. In Section IV, we present our differentiable particle filter transformer for single-speaker tracking and discuss the extension of this model for multiple-speaker scenarios. In Section V, we show the experimental results of the baseline methods and the proposed method. We also discuss the model robustness against noise and sequence length. In Section VI, we conclude the paper and point out the limitation of our method and potential directions for future works.

II. RELATED WORK

A. SPEAKER TRACKING

In the past few years, Bayesian filter based methods have been developed for speaker tracking. Most methods adopt the paradigm of filtering with a measurement model. In [10], an adaptive particle filter is proposed for single speaker tracking. It uses GCF as audio measurement and uses face detection and color histogram as video measurement. An adaptive weight mechanism is designed to determine the importance of audio and visual modality dynamically. In general, training a model for extracting measurements requires abundant labeled data, which is not always available. Self-supervised learning and active learning proposed in [11] and [12] can be used to leverage unlabeled data to obtain measurements. In [13], an algorithm similar to [10] is explored under a reverberant and noisy environment with occluded speakers, speakers out of the field of view, and speakers not facing the cameras. In [6], a new dataset named CAV3D is proposed for audio-visual speaker tracking. Compared to the widely used AV16.3 dataset [14], CAV3D contains recordings with stronger reverberation and more complicated scenarios. In [15], particle filter is used for multiple speaker tracking with discriminative and generative measurement likelihood. In [16], a two-layer particle filter is proposed. Two groups of particles are passed through the audio and visual layers separately. The particle weights are determined by the likelihood of the two modalities.

The random finite set (RFS) based method is another branch of Bayesian filter, which can handle the varying number of speakers. RFS contains a varying number of elements. Both the target and measurement sets can be represented by the RFS. At each time step, the speaker RFS is the combination of surviving speakers, spawned speakers from last time step, and new speakers. Here, the spawned speakers refer to the speakers appeared in the last step and could be potentially existing in current time step without associated measurements, such as the occluded speakers. To lower the computational complexity of RFS, the probability hypothesis density (PHD) filter propagates the first-order moment of the multiple target state distribution. It has a linear Gaussian form, representing the target in Gaussian distribution and SMC form, representing the target with particles. In [17], the PHD filter is used for tracking unknown and varying number of moving audio sources. In [18], the SMC PHD filter is employed with the help of mean-shift to move particles to the local maximum. There are several works [19], [20] that combine particle flow with PHD filter while particle flow can help to transfer the particles from prior distribution to posterior distribution. Unlike the PHD filter, multi-target multi-Bernoulli (MeMBer) filter propagates posterior density function rather than the first-order moment. The target state is represented as Bernoulli RFS, which is empty or has a single element. In [1], generalized labeled Bernoulli filter (GLMB) is employed to solve the problem of multi-modal space-time permutation and deal with the problem of varying number of speakers. In [21], the Poisson multi-Bernoulli mixture (PMBM) filter is proposed for multi-target speaker tracking, which employs Poisson distributions to represent undetected targets and employs a multi-Bernoulli mixture to represent detected targets with different data association strategies.

B. DIFFERENTIABLE BAYESIAN FILTER

There have been some recent works that combine Bayesian filter and deep learning models for temporal prediction tasks. In [22], a backprop Kalman filter is proposed, which takes the raw image as the input and outputs the tracking results. In [23], a dynamic weight mechanism is jointly trained with backprop Kalman filter so that the importance of different modality can be determined by the quality of the measurements. There are also some works that combine neural networks with particle filter. In [24], a differentiable particle filter is designed with a semi-supervised learning strategy to reduce the requirement of labeled data. In [25], particle filter is combined with simultaneous localization and mapping (SLAM) for visual navigation. In [26], a particle filter network is proposed for visual localization, which encodes the measurement model and the particle filter in a single neural network. In [9], a similar architecture has been implemented, where the training strategy is formed in three steps, with two steps on training the transition and measurement models, and the final step on end-to-end learning of the whole model. In addition to the conventional neural network, a recurrent neural network, such as long short term memory (LSTM) and gated recurrent unit (GRU), can also be combined with a particle filter. In [27], PF-LSTM and PF-GRU are proposed, which replace the deterministic update with stochastic Bayesian update. In [28], particle transformer is proposed, which leverages weighted multi-head attention for differentiable resampling. Compared to [26] and [9], our model extends the localization (tracking) task to multiple objects for the first time. Compared to [27], our model integrates particle filter and transformer for object tracking for the first time, while in [27], particle filter is combined with recurrent neural network. Compared to [28], we combine particle filter and transformer for object tracking, while in [28], the two models are combined for differentiable resampling. In our proposed model, the particle states are also changed in the observation model [27], which is different from a vanilla particle filter where the particle states remain unchanged in the observation model. The design of the differentiable particle filter is used to provide a training strategy, such as using weighted particles for state representation and particle resampling.

III. PROBLEM FORMULATION

The whole algorithm design is based on particle filter framework. In a particle filter, N particles $\{w_{t,i}, \mathbf{x}_{t,i}\}_{i=1}^{N}$ are used to represent the target states, where $\mathbf{x}_{t,i}$ denotes the state of the i-th particle at time t, containing the DOA $d_{t,i}$ and distance $\delta_{t,i}$, with $w_{t,i}$ being its corresponding weight. A standard PF has four steps: initialization, prediction, update and resampling. In the first step, all particle weights are initialized at $t = 1$ to be the same:

$$\{w_{1,i}\}_{i=1}^{N} = 1/N \tag{1}$$

In the prediction step, the particle states $\{w_{t,i}, \mathbf{x}_{t,i}\}_{i=1}^{N}$ are transited from the last time step to the current time step $\{w_{t,i}, \mathbf{x}_{t+1,i}\}_{i=1}^{N}$ using the transition model:

$$\mathbf{x}_{t+1} \sim T(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{2}$$

where T is the transition model and is assumed to be a constant velocity model.

In the update step, the measurement likelihood l is first calculated:

$$l = M(\mathbf{Z}_{t+1}|\mathbf{x}_{t+1}) \tag{3}$$

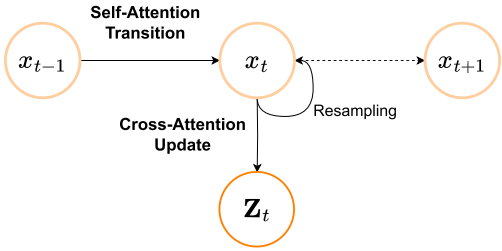

FIGURE 1. Overview of the model architecture, where a learned transition and update model is proposed. Here, s_t is the speaker state at time t and z_t is the measurement at time t.

where $\mathbf{Z}_{t+1}$ is the observation set, and M is the measurement model. The particle weights are updated by the measurement likelihood:

$$w_{t+1,i} = \frac{l_i \cdot w_{t,i}}{\sum_{k=0}^{N} l_k \cdot w_{t,k}} \tag{4}$$

The target state $\mathbb{X}$ is obtained as the weighted sum of the particle states:

$$\mathbb{X}_{t+1} = \sum_{i=1}^{k} w_{t+1,i} \cdot \mathbf{x}_{t+1,i} \tag{5}$$

After some iterations, particle filter may suffer from the weight degeneracy problem, where the target state is determined by a few high-weight particles. Therefore, the last step is particle resampling, where the particles with higher weights are duplicated while the particles with lower weights are discarded. The resampled particles are assigned with identical weights.

In this paper, we explore using audio signals captured by microphones for speaker tracking. Given the binaural audio waveform $\{\mathbf{a}_1, \mathbf{a}_2\}$ captured by two microphones, where $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^{|\mathbf{a}|}$ with $|\mathbf{a}|$ being the length of the waveform, the task of speaker localization aims to predict the direction of arrival (DOA) of the sound sources from the speakers with respect to the microphone array at each time step.

We use GCC-PHAT as the audio feature, which is commonly used in speaker tracking and calculated as:

$$\mathbf{G}_{i,j}(t,\tau) = \int_{-\infty}^{+\infty} \frac{STFT_{\mathbf{a}_i}(t,f) STFT^{*}_{\mathbf{a}_j}(t,f)}{\left|STFT_{\mathbf{a}_i}(t,f)\right| \left|STFT^{*}_{\mathbf{a}_j}(t,f)\right|} e^{j2\pi f\tau} df \tag{6}$$

where $\mathbf{G} \in \mathbb{R}^{T \times C}$, with T being the temporal dimension, C is the number of coefficients of delay lags, τ is the time delay lag, (i, j) denotes a microphone pair, $STFT$ represents Short Term Fourier Transform with (t, f) being time frame and frequency bin indexes, respectively, and $*$ denotes complex conjugate.

IV. PROPOSED METHODS

In this section, we show an end-to-end differentiable architecture which combines particle filter and transformer for single speaker tracking. Then, we discuss the extension of the proposed model to the problem of multi-speaker tracking.

A. ATTENTION-BASED DIFFERENTIABLE PARTICLE FILTER

The overview of the model can be seen in Fig. 1. The self-attention acts as an implicit learnable transition model with which the particle states are transferred to the next time step without applying an

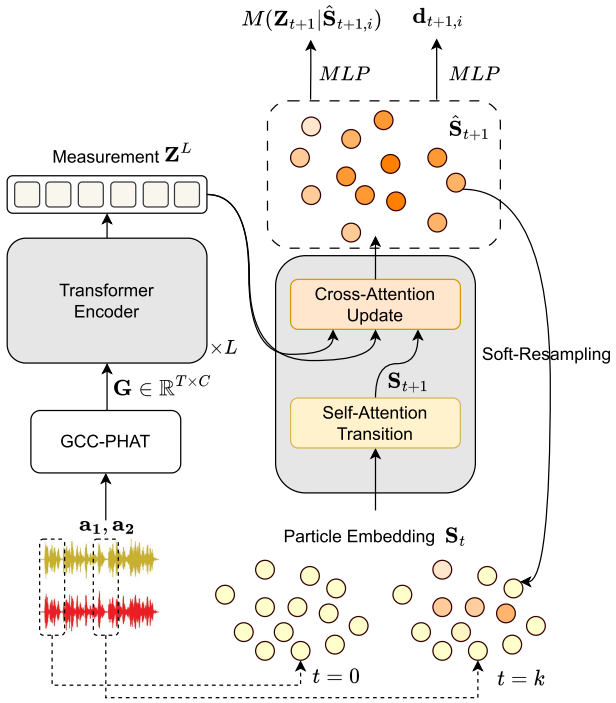

FIGURE. 2. Architecture of the proposed model architecture. GCC-PHAT of the two-channel audio is calculated and split as the input to the encoder. The decoder takes the particle embedding as input and performs self-attention and cross attention with the encoder output as the transition model and update model, respectively. Initially the colors of particles are the same, indicating the same weights. After the update, the particles are denoted by different colors with deeper color indicating higher weights. Finally, soft-resampling is used to select important particles.

explicit motion model to the particles. The cross-attention module is used to calculate the measurement likelihood.

The overall architecture of our model is shown in Fig. 2, which follows the paradigm of a vanilla transformer. The GCC-PHAT $\mathbf{G}$ is firstly added with the positional encoding $\mathbf{G}_{POS} \in \mathbb{R}^{T\times C}$ along the time dimension and input to the transformer encoder. In the transformer encoder, the input goes through the multi-head self-attention (MSA) layers and the fully connected layers with the residual connection. This can be described mathematically as follows,

$$\mathbf{Z}^0 = \mathbf{G} + \mathbf{G}_{POS} \tag{7}$$

$$\hat{\mathbf{Z}}^l = \mathrm{LN}\left(\mathrm{MSA}\left(\mathbf{Z}^{l-1}\right)\right) + \mathbf{Z}^{l-1}, \quad l = 1 \ldots L \tag{8}$$

$$\mathbf{Z}^l = \mathrm{LN}\left(\mathrm{MLP}\left(\hat{\mathbf{Z}}^l\right)\right) + \hat{\mathbf{Z}}^l, \quad l = 1 \ldots L \tag{9}$$

where $\hat{\mathbf{Z}}$ is the intermediate state after MSA layers and $\mathbf{Z}$ is the output of one transformer encoder module. $\hat{\mathbf{Z}}^l, \mathbf{Z}^l \in \mathbb{R}^{T\times F}$, with T being the length of the temporal feature and F being the feature dimension, l is the index of the transformer module, L denotes the number of repeated transformer encoder modules, MLP represents multi-layer perception, and LN is the layer normalization. The MSA is defined as,

$$\mathrm{MSA}\left(\mathbf{Z}\right) = \mathit{Concat}(\mathbf{H}_1, \ldots, \mathbf{H}_n)\mathbf{W}_o \tag{10}$$

where $\mathbf{H}_\omega$, $\omega = 1, \ldots, n$, are computed as

$$\mathbf{H}_\omega = \mathrm{softmax}\left(\frac{\mathbf{Z}\mathbf{W}_{Q_\omega}\mathbf{W}_{K_\omega}^T\mathbf{Z}^T}{\sqrt{d_K}}\right)\mathbf{Z}\mathbf{W}_{V_\omega} \tag{11}$$

where $\mathbf{W}_o \in \mathbb{R}^{T_o\times d_o}$, $\mathbf{W}_Q \in \mathbb{R}^{T_Q\times d_Q}$, $\mathbf{W}_K \in \mathbb{R}^{T_K\times d_K}$ and $\mathbf{W}_V \in \mathbb{R}^{T_V\times d_V}$ are trainable matrices. T_* is the sequence length and d_* is the feature dimension. In the transformer decoder, the particles are represented as the embedding matrices $\mathbf{S}_t \in \mathbb{R}^{N\times D}$, where N is the number of particles and D is the hidden dimension of the particle embedding. Each particle embedding implies the particle position. An advantage of the proposed model is that, as the feature extractor, the transformer encoder can be optimized over a sequence of audio frames instead of a single frame [22].

The particle embedding $\mathbf{S}_t$ is firstly added with the positional encoding $\mathbf{S}_{POS} \in \mathbb{R}^{N\times D}$ and then passed through the self-attention layer. The self-attention layer is applied on the first dimension of the particle embedding, which is regarded as the transition model in a particle filter. The transition of one particle state depends on the self-attention with other particle embedding,

$$\mathbf{S}_t = \mathbf{S}_t + \mathbf{S}^{POS} \tag{12}$$

$$\mathbf{S}_{t+1} = \mathrm{LN}\left(\mathrm{MSA}\left(\mathbf{S}_t\right)\right) + \mathbf{S}_t, \tag{13}$$

After the self-attention transition, particle states are transferred from $\mathbf{S}_t$ to $\mathbf{S}_{t+1}$. Then cross attention is used between the predicted particle states with the output of the encoder. The encoder output also contains corrupted information for DOA estimation, such as clutter, outliers and noise. The multi-head cross attention layer (MCA) is regarded as the measurement model and the output of the encoder is regarded as the measurement.

$$\hat{\mathbf{S}}_{t+1} = \mathrm{LN}\left(\mathrm{MCA}\left(\mathbf{S}_{t+1}, \mathbf{Z}^L\right)\right) + \mathbf{S}_{t+1}, \tag{14}$$

where the MCA operation is defined as

$$\mathrm{MCA}\left(\mathbf{S}, \mathbf{Z}\right) = \mathit{Concat}(\mathbf{H}_1, \ldots, \mathbf{H}_n)\mathbf{W}_o \tag{15}$$

where $\mathbf{H}_\omega$, $\omega = 1, \ldots, n$, are computed as

$$\mathbf{H}_\omega = \mathrm{softmax}\left(\frac{\mathbf{S}\mathbf{W}_{Q_\omega}\mathbf{W}_{K_\omega}^T\mathbf{Z}^T}{\sqrt{d_K}}\right)\mathbf{Z}\mathbf{W}_{V_\omega} \tag{16}$$

where $\mathbf{W}_o$, $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ are defined similarly as earlier. A fully connected layer is used to calculate the measurement likelihood in terms of the particle embedding.

$$M(\mathbf{Z}_{t+1}|\hat{\mathbf{S}}_{t+1,i}) = \mathrm{MLP}\left(\hat{\mathbf{S}}_{t+1,i}\right) \tag{17}$$

The particle weights are updated according to the likelihood:

$$w_{t+1,i} = M(\mathbf{Z}_{t+1,i}|\hat{\mathbf{S}}_{t+1,i}) \cdot w_{t,i} \tag{18}$$

The corresponding DOA posterior $\mathbf{d}_{t+1,i} \in \mathbb{R}^{360}$ are derived by an MLP layer in terms of the updated particle states:

$$\mathbf{d}_{t+1,i} = \mathrm{MLP}\left(\hat{\mathbf{S}}_{t+1,i}\right) \tag{19}$$

The final DOA posterior $\mathbf{d}_{t+1} \in \mathbb{R}^{360}$ is the weighted sum of the DOA posterior over all the particles:

$$\mathbf{d}_{t+1} = \sum_{i=1}^{N} w_{t+1,i}\mathbf{d}_{t+1,i} \tag{20}$$

Finally, the DOA is obtained as the peak index of the posterior:

$$\hat{d}_{t+1} = \arg\max_j \left(d_{j,t+1}\right) \tag{21}$$

where $d_{j,t+1}$ is the j-th element of the vector $\mathbf{d}_{t+1}$, and $j = 1, \ldots, 360/\epsilon$ with ϵ being the angle resolution, set to $\epsilon = 1^\circ$ in our experiments.

B. RESAMPLING

The resampling step selects and duplicates the important particles and discards the unimportant ones. However, the resampling step is not differentiable. To integrate the resampling step into the transformer, similar to [27] [26], we employ the soft-resampling. In soft-resampling, we resample the particles from a new distribution q instead of the original distribution p, where q is the combination of p and uniform distribution $u = 1/K$ with $0 < \alpha < 1$ being the hyperparameter to balance the two distributions, as follows

$$q(\cdot) = \alpha p(\cdot) + (1-\alpha)u(\cdot) \tag{22}$$

Instead of using equal weights for all the particles, the new particle weights are calculated as follows:

$$\hat{w}_t^i = \frac{p(i)}{q(i)} = \frac{w_t^i}{\alpha w_t^i + (1-\alpha)1/K} \tag{23}$$

C. EXTENSION TO MULTIPLE SPEAKERS

The extension of the proposed method to the problem of tracking multiple speakers follows the setting in a conventional particle filter. In the conventional particle filter, several groups of independent particles are used for different objects. In this paper, as an example, we consider the scenario of tracking two speakers. To this end, two independent groups of particles are employed as input to the decoder of the transformer. For the self-attention based transition step, the particles from the two groups are processed independently with an attention mask $\begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{pmatrix}$ to ensure the transition of different groups does not interfere with each other, where $\mathbf{I}$ is an identity matrix whose dimension is identical to the number of particles used for tracking each speaker.

The cross attention update is the same as that of (14)–(19). Compared to the single speaker tracking, multi-speaker tracking has a data association step, in which the measurements are matched with the speaker states. The data association is hidden in the cross attention update step and each particle embedding automatically finds the related measurements. When obtaining DOA, (20) and (21) are performed for each group of particles. In the multi-speaker tracking scenario, the number of the speakers may be varying with time. Thus, we add a binary classifier to estimate whether the particles correspond to an existing target,

$$\hat{\mathbf{E}}_{t+1} = \mathrm{MLP}\left(\hat{\mathbf{S}}_{t+1}\right) \tag{24}$$

where $\hat{\mathbf{E}} \in \mathbb{R}^{2\times 2}$ in the two-speaker scenario, which is rescaled using the *softmax* function to obtain the predicted speaker existence possibility. After obtaining the DOAs of multiple speakers and the existence probability by averaging the particle states according to their weights, the Hungarian algorithm [29] is used to match the estimated DOA with the ground truth. Similar to [30], the matching strategy is applied on the basis of the speaker existence probability and the DOA estimation, as follows

$$\begin{aligned} \sigma(\mathbf{\Pi}_{\sigma(\zeta)}, \mathbf{y}_\zeta) = & -\mathbf{1}_{\{c_\zeta \neq \varnothing\}} \hat{\mathbf{E}}\left(\sigma(\zeta), c_\zeta\right) \cdot \lambda_{cls} \\ & + \mathbf{1}_{\{c_\zeta \neq \varnothing\}} \mathcal{L}_{\mathrm{DOA}}\left(r_\zeta, \hat{d}_{\sigma(\zeta)}\right) \cdot \lambda_{DOA} \end{aligned} \tag{25}$$

where $\mathbf{1}_{\{c_\zeta \neq \varnothing\}}$ is an indicator function, taking the value of 1 if c_ζ is not empty. ζ is the index of the ground truth and $\sigma(\zeta)$ is the matching index. $\mathbf{\Pi}_{\sigma(\zeta)} = (\hat{\mathbf{E}}_{\sigma(\zeta)}, \hat{d}_{\sigma(\zeta)})$ and $\mathbf{y}_\zeta = (c_\zeta, r_\zeta)$ where $c_\zeta \in \{0, 1\}$ indicates the speaker existence and r_ζ is the ground truth DOA. $\mathcal{L}_{\mathrm{DOA}}$ is the absolute difference between $\hat{d}_{\sigma(\zeta)}$ and r_ζ. λ_{cls} and λ_{DOA} are the hyperparameters to balance the classification error and the DOA error.

At last, soft-resampling in terms of (23) is performed for each group of particles, independently. The overall process for multi-speaker tracking is shown in Algorithm 1.

D. LEARNING OBJECTIVE

For the DOA distance loss, we cannot use the distance between the predicted DOA and the ground truth DOA. This is because the arg max operation in (21) is not differentiable. Some works [31], [32] treat the DOA estimation task as the classification task and use the cross entropy loss. With a resolution ϵ, the DOA space is split into $360/\epsilon$ classes. However, the cross entropy loss cannot describe the relationship among different classes. For instance, the error between 0° and 180° should be larger than that between 0° and 90°. However, the cross entropy loss will treat them equally. To model the relationship between different classes, inspired by [5], we encode the ground truth r with a Gaussian distribution centered on ground truth DOA,

$$r_\psi \sim \mathbb{N}\left(r, \sigma^2\right) \tag{26}$$

where r_ψ are the ψ-th element of ground truth of the DOA $\mathbf{r}$. We generate the Gaussian distribution centered on the ground truth DOA with resolution $\epsilon = 1^\circ$ and covariance $\sigma^2 = 1^\circ$. Then we use the earth mover's distance (EMD) loss [33], used originally for speech quality evaluation [34], to measure the difference between the DOA posterior and the Gaussian distribution of the ground truth, as follows

$$\mathcal{L}_{\mathrm{EMD}}(\mathbf{d}) = \sum_{\psi=1}^{360/\epsilon} (d_\psi - r_\psi) \tag{27}$$

where d_ψ and r_ψ are the ψ-th element of DOA posterior $\mathbf{d}_{t+1}$ and ground truth $\mathbf{r}$, respectively.

Besides, we adopt the evidence lower bound (ELBO) loss [27] to maximize the particle likelihood:

$$\mathcal{L}_{\mathrm{ELBO}} = -\log \frac{1}{N} \sum_{i=1}^{N} p\left(\mathbf{r}_t \mid \mathbf{S}_{1:t,i}, \mathbf{Z}_{1:t}^L, \right) \tag{28}$$

where $\mathbf{r}_t$ is the ground truth DOA, $\mathbf{S}_{1:t}$ is the updated particle embedding. For the likelihood, we adopt $p(\mathbf{r}_t \mid \mathbf{S}_{1:t,i}, \mathbf{Z}_{1:t}^L,) = \mathcal{L}_{\mathrm{EMD}}(\mathbf{d}_i)$ to calculate the EMD loss between particle states and the ground truth Gaussian distribution. The EMD loss provides a macro optimizing strategy to improve the model performance while the ELBO gives a micro optimizing strategy to focus on the particle estimation. The learning objective is the combination of the ELBO loss and the DOA distance loss with the hyperparameter λ:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\mathrm{ELBO}} + (1-\lambda) \cdot \mathcal{L}_{\mathrm{EMD}} \tag{29}$$

For the multiple speaker scenario, the cross entropy loss $\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{E}}, c)$ is added for class prediction, which is defined as follows,

$$\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{E}}, c) = \sum_{k=0}^{M-1} -(c_k \log \hat{\mathbf{E}}(k, 1) + (1-c_k) \log \hat{\mathbf{E}}(k, 0)) \tag{30}$$

where $\hat{\mathbf{E}}(k, 0)$ is an element of the matrix $\hat{\mathbf{E}}$ at the position of the k-th row and first column, likewise, $\hat{\mathbf{E}}(k, 1)$ represents the element at the k-th row and second column.

V. EXPERIMENTS

In this section, we first evaluate the performance of the proposed model, as compared with the baseline models on single speaker tracking. We then compare the robustness of different models against sequence length and additive audio noise. At last, we show the model performance on multi-speaker tracking.

Algorithm 1: Attention-Based Differentiable Particle Filter for One Time Step.

Input: The maximum number of speakers M, GCC-PHAT $\mathbf{G}$

Output: The number of speakers S. The DOA of speakers $\hat{d}_1, \hat{d}_2, ..., \hat{d}_S$

Initializing particle embedding $\mathbf{S}_t^1, ..., \mathbf{S}_t^M$ $(t = 0)$ or inheriting from last time step $(t > 0)$;

Encoding $\mathbf{G}$ following Eq. (8) and (9);

for $k = 1, ..., M$ **do**

  Self-attention transition for particle embedding $\mathbf{S}_t^k$ following Eq. (12) and (13);

Cross-attention update following Eq. (14);

Weight updating following Eq. (17) and Eq. (18);

Calculating the possibility of speaker existence following Eq. (24) and estimating S;

According to S, calculate the corresponding DOA $\hat{d}_1, \hat{d}_2, ..., \hat{d}_S$ following Eq. (19), (20) and (21);

Soft-resampling following Eq. (23);

return $S, \hat{d}_1, \hat{d}_2, ..., \hat{d}_S$;

A. DATASET

In this paper, we focus on audio speaker tracking. Most speaker tracking datasets (2 hours for the AV16.3 dataset [14], 2 hours for the CAV3D dataset [6], and 0.5 hours for the AVDIAR dataset [35]) have limited size and do not support the training of deep learning models. Therefore, we resort to a simulated dataset. We use Two Ear auditory model[1] for simulating the binaural audio with moving sound source due to its easy implementation. Binaural audio has been used for localization, tracking and navigation [36], [37]. The dataset is simulated within a 2D room of 20×20 squared meters. The initial position of the speaker is chosen randomly within the room. The walking speed of the sound source is set to three meters per second to mimic the normal walking scenario. The direction of the velocity is randomly generated and fixed, so the sound source moves with a linear and constant speed. We have created almost 33 k trajectories with each trajectory containing 50 k sampling points. The speech corpus is from Librispeech [38]. The training set is from *train-clean-360*. The development set is from *dev-clean* and *dev-other*. The test set is from *test-clean* and *test-other*. We cut each speech clip to 10 seconds and input to the Two Ears auditory system with a trajectory to generate binaural audio with a sampling interval of 368 with a sampling rate of 44100. We collect around 33 k spatial speech clips of more than 100 hours in total. We use 27 k clips as the training set, 3 k clips as the development set, and another 3 k clips as the test set.

For the multi-speaker tracking scenario, we randomly choose two audio clips from the single-speaker dataset and add their waveforms to simulate the two-speaker scenario. In this way, we create a corpus of another 100 hours of speech data. Together with the single-speaker corpus, we use the blended corpus to train the multi-speaker tracking model. The number of audio clips in the training set, the development set and the test set is 60 k, 6 k and 6 k, respectively.

[1][Online].Available: https://github.com/TWOEARS/TwoEars

B. IMPLEMENTATION DETAILS

For GCC-PHAT calculation, we split each audio clip into chunks with a hop size of 368 to match the simulation interval. For each chunk, the GCC-PHAT is calculated by including six previous chunks and six latter chunks. The *n_fft* is set to 1024 and the hop size is set to 320. The number of coefficients of the delay lags is set to 96.

Both the transformer encoder and decoder use one transformer module to mimic the process in a particle filter. The dimension of the query, key and value is 128. The dimension of the latent representation from the fully-connected layer is 256. The number of heads for the multi-head attention is 4. For the particle filter, we use 30 particles with 128 dimension for the latent representation. α for soft-resampling is set to 0.2.

For model training, we run the Adam optimizer for 100 epochs in total, with the learning rate set to 5e-4 for the initial 50 epochs, and then decreased for the remaining 50 epochs with 0.1 learning rate decay. We adopt the early stop mechanism with 30 patience. For the Hungarian matching, $\lambda_{cls} = 3$ and $\lambda_{DOA} = 5/180$. For the learning objective, the hyper parameter λ is set to 0.5. For the multiple speaker scenario, we adopt the EMD loss and cross entropy loss. We discard the ELBO loss to reduce the computational cost.

C. EVALUATION METRICS

We use the mean absolute error (MAE) and Accuracy to evaluate the model performance. The MAE is calculated as:

$$MAE = \frac{1}{T \cdot M} \sum_{t=0}^{T} \sum_{m=0}^{M} (\pi - |\hat{d}_{t,m} - r_{t,m}|) \tag{31}$$

where T is the number of time steps and M is the number of speakers. MAE is the average error along the time dimension within one trajectory, and is smaller than 180 degrees. The Accuracy is calculated as the percentage of the trajectory whose MAE is smaller than three degrees. For the multi-speaker scenario, we also report the cardinality error, calculated as the absolute value of the estimated number of speakers and the ground truth.

D. COMPARISON WITH OTHER METHODS

We compare the proposed model with other temporal prediction models including the vanilla RNN, LSTM, and GRU, and other models which combine RNN and PF such as PF-LSTM and PF-GRU [27]. We choose the RNN-based models as baseline methods as they estimate the states in the current time step based on the current measurements and the previous states, in a similar spirit to the Bayesian filters. When re-implementing the baseline models, we choose a proper dimension of the embedding to ensure that different models have roughly the equivalent number of model parameters. The GCC-PHAT is directly input to the RNN-based method to obtain DOA.

The experimental results on the simulated dataset are shown in Table 1. It is observed that the proposed particle filter transformer outperforms the baseline methods by a large margin. The transformer encoder can provide extracted features and the transformer decoder combined with particle filter can estimate the speaker state. LSTM and GRU offer a better performance than the vanilla RNN as they have better ability in modelling longer sequences through different gates for remembering important information and discarding redundant information.

TABLE 1 Experimental Results on the Simulated Dataset

	MAE (°)	Accuracy (%)
RNN	44.31	49.94
LSTM	13.09	78.63
GRU	10.27	80.78
PF-LSTM [27]	23.24	69.42
PF-GRU [27]	22.86	70.75
Ours	**4.40**	**95.17**

The bold numbers denote the lowest MAE or the highest Accuracy.

TABLE 2 Ablation Study

	MAE (°)	Accuracy (%)
w/o End2End Training	6.33	89.85
w/o Temporal Prediction	4.44	**95.33**
Ours	**4.40**	95.17

The bold numbers denote the lowest MAE or the highest Accuracy.

E. ABLATION STUDY

We conduct the ablation study to show the effectiveness of the proposed end-to-end attention-based particle filter. The results of the ablation study are shown in Table 2. The first model uses transformer to obtain the measurements and uses the conventional particle filter for tracking. In each training iteration, only the transformer is optimized. For the transformer to obtain measurements, we use the transformer encoder and add an [*CLS*] token at the beginning of the GCC-PHAT. We use the first position of the output and pass it to a classification layer to get the DOA. We adjust the dimension of the hidden layers to match with the models in Table 1. It is observed that the performance of the two-stage model is not as good as the one-stage end-to-end model, which shows the effectiveness of the end-to-end model. The second model uses the same architecture as the proposed model but without temporal dependency. At each time step, new particles are generated as input to the decoder instead of using the resampled particles from the last step. The performance of the second model is better than that of the two-stage model and shows the competitive performance of our proposed model. The reason is that the feature contains the information from both previous and latter audio chunks. As explained in Section V.B, one chunk is calculated with six previous chunks and six latter chunks. While the input features already incorporate temporal information, the impact of temporal prediction effectiveness is not immediately apparent.

We also show how the change in the number of particles affects the model performance in Table 3. It can be seen that the model achieves the best Accuracy with 30 particles and gives the lowest MAE errors with 10 particles. A larger or smaller number of particles may lead to performance decline. In addition, we integrate differentiable resampling mechanism [28] with the proposed model and compare its performance with that of the model using soft-resampling. The experimental results are shown in Table 5. We can find that the performance of the model leveraging differentiable resampling is not as good as that using soft-resampling. Both the differentiable resampling method and our proposed model use transformer blocks. However, the cascaded Transformer architecture is hard to converge and achieve optimal performance. While the model [28] leverages fully connected layers as the transition and update models, which are more light-weighted and suitable for integrating with differentiable resampling. The soft-resampling mechanism is a model-free architecture and fits better with our transformer-based model.

TABLE 3 Impact of the Number of Particles on the Performance of the Model

No. Particles	MAE (°)	Accuracy (%)
5	4.86	94.63
10	**4.28**	94.57
20	5.43	94.35
30	4.40	**95.17**
40	5.56	92.73

The bold numbers denote the lowest MAE or the highest Accuracy.

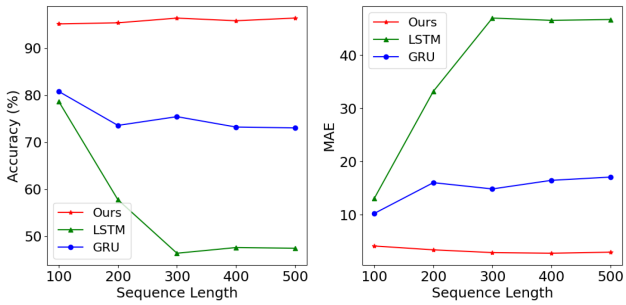

FIGURE 3. Impact of sequence length on the performance of the model.

F. THE IMPACT OF SEQUENCE LENGTH

In this section, we explore the impact of the sequence length on the model performance. We test three well-performing models in Table 1, LSTM, GRU and our proposed model under the sequence length of 100, 200, 300, 400 and 500, and the results of MAE and accuracy are reported in Fig. 3. It can be seen that the performance of LSTM and GRU drops significantly (Accuracy decreased from 78% to 50% for LSTM and accuracy decreased from 80% to 73% for GRU) with the increase of the sequence length. While the performance of our model is relatively stable against the long sequence with the Accuracy maintained in around 95%, which proves that our model has strong modeling and memory capacities over long sequence.

G. THE IMPACT OF NOISE

The simulated dataset we generate does not contain noise in the binaural audio. However, in real applications, audio signals captured are often contaminated by noise. Therefore, in this section, we explore the model robustness against noise. To this end, we add noise to the development set and the test set of the simulated dataset. The noise we use is from DEMAND [39], which provides noise from different scenarios including office, park, sports fields, and so on. The noise is added to the magnitude spectrum with Signal-to-Noise Ratio (SNR) of 20 db, 10 db, 0 db and −10 db, respectively. The experimental results are demonstrated in Table 4. We use the pre-trained models on the clean training set and evaluate it on the noisy version of the test set without finetuning the model. It is observed that our model performs better than the baseline models on all the SNR levels. The

TABLE 4 Experimental Results Against Noise Where * Denotes the Model That is Trained on Gaussian White Noise Data

	20db		10db		0db		-10db	
	MAE (°)	Accuracy (%)	MAE (°)	Accuracy (%)	MAE (°)	Accuracy (%)	MAE (°)	Accuracy (%)
RNN	62.02	35.96	69.89	29.17	78.33	20.88	83.51	13.63
LSTM	50.97	43.47	63.21	33.36	75.48	21.78	83.32	13.44
GRU	51.60	44.40	65.20	32.66	75.87	21.54	84.36	12.18
PF-LSTM [27]	57.95	39.68	67.31	31.27	77.41	21.49	83.44	13.38
PF-GRU [27]	60.89	38.34	70.37	29.78	80.00	20.22	86.44	12.06
Ours	**18.39**	**78.86**	**32.03**	**63.10**	**48.53**	**43.96**	**65.50**	**25.73**
Ours*	**11.92**	**85.86**	**17.71**	**78.88**	**27.08**	**68.39**	**38.05**	**55.00**

The bold numbers denote the lowest MAE or the highest Accuracy.

TABLE 5 Experimental Results of Different Resampling Methods

Resampling	MAE	Accuracy
Soft resampling	4.40	95.17
Differentiable resampling [28]	12.17	87.67

TABLE 6 Experimental Results on the Two-Speaker Scenario

	Card	MAE (°)
DETR [30] + PMBM [21]	0.90	31.48
Ours	0.35	31.91

Card Denotes the Cardinality Error.

performance of RNN is the worst due to its simplest architecture. Other models offer performance in a similar level. On the SNR level of 20 db, 10 db and 0 db, our model outperforms the baseline models to a large extent (accuracy increased almost 20% on 20 db and 10 db, and increased almost 10% on 0db), which shows that our model is more robust to additive noise. On the SNR level of -10 db, the power of the noise is greater than that of the speaker audio and this scenario is very challenging. The performance of all models is not satisfactory. However, in the more noisy environment our model still obtains 10% increased accuracy for the noise level at -10 db compared to the recurrent baseline models.

To increase the model robustness against noise, we also train the model on the noisy dataset. Specifically, we add Gaussian white noise to the training set, with the SNR level at 20 dB. We test the model on the noisy dataset contaminated by DEMAND [39]. It is observed that the model performance (**Ours***) improves as the Gaussian white noise is the most general form. The model trained with Gaussian white noise tends to be generalized to the specific noise.

H. VISUALIZATION

To present the tracking results more intuitively, we show the trajectories and particle states in Fig. 4. As the DOA classification resolution is 1°, the estimated DOA are not in decimal points, and the estimated trajectories appear to be saw-toothed. It can be seen that at the start period, the particles are scattered in different positions, which are similar in conventional particle filters. After some iterations, the particles converge to certain points. Although several outliers exist (the impulses in the third and fifth sub-figures), the estimated trajectories are close to the ground truth trajectories.

I. RESULTS ON MULTI-SPEAKER SCENARIO

We report the results of two-speaker tracking in Table 6. We compare the baseline method DEtection TRansformer (DETR) [30] combined with the Poisson multi-Bernoulli mixture (PMBM) filter [21], which is a two-stage process. DETR [30] is used to extract audio measurements. The vanilla DETR, originally proposed for the task of object detection, has a classification head and a localization head to determine the position of the bounding box. The Hungarian matching module is used for bipartite matching during the training stage, which is not differentiable. We modify the classification head to a binary classifier to determine the speaker existence, and modify the localization head for classification of 360 classes related to the DOA angles. DETR is trained on the simulated two-speaker dataset. PMBM is used for state estimation. Compared to PF, PMBM can estimate the varying number of speakers while PF needs to set the number of speakers as a prior. PMBM has been used for vehicle tracking [40] and multiple speaker tracking [41]. PMBM takes the audio measurements from DETR [30] and estimates the number of speakers and each speaker's DOA.

In the baseline of PMBM filter, the survival probability was set to 0.99 and the birth model is set to a Gaussian mixture. The detection probability is set to 0.9. Both the baseline method and our proposed model can be used for estimating the number of speakers. It is observed that the proposed method performs competitively with the baseline method in MAE and outperforms the baseline method in cardinality estimation. It can be seen that the multi-speaker tracking task is more challenging than single-speaker tracking. There are two reasons. On one hand, the GCC-PHAT feature is hard to be adapted in the multi-speaker scenario [7]. The performance of GCC-PHAT degrades significantly as the number of speakers increases [15]. On the other hand, data association is needed to match the measurements with the targets.

J. RESULTS ON REAL DATASET

We evaluate our methods on a real dataset, i.e. AVRI dataset [42], which is recorded with a four-microphone array and KINOVA robot. Compared to the simulated dataset, the real dataset is more complicated, covering varying reverberation, noise and speaker motions. The experimental results are shown in Table 7. It is shown that our proposed method offers competitive performance as compared with the state-of-the-art methods, despite having a smaller model size. Both A-CRNN [7] and AV-CRNN [42] employ GCC-PHAT and mel-spectrogram as audio features. In addition, both AV-CRNN [42]

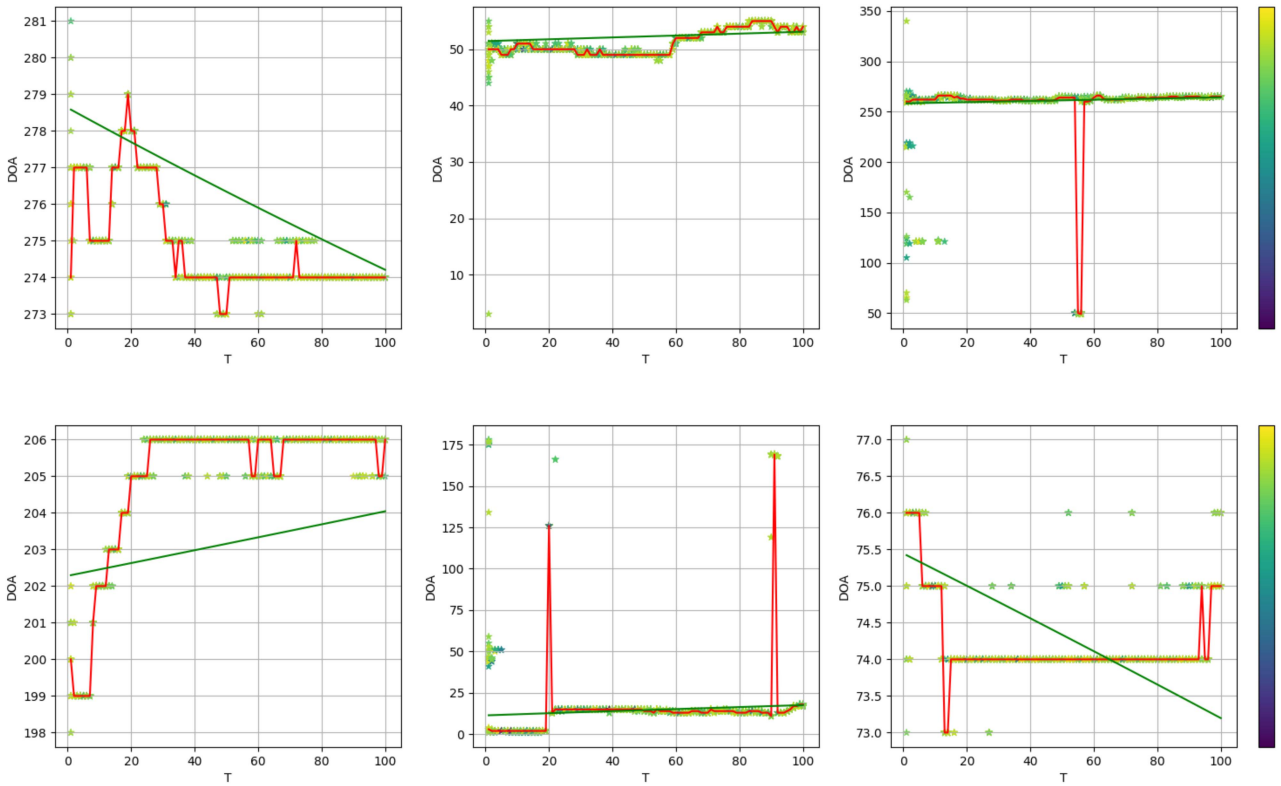

FIGURE 4. Visualization of tracking trajectories. The horizontal axis is the time and the vertical axis is DOA. The red and green lines represent the estimated trajectories and ground truth, respectively. The star represents the particles whose colors represent the particle weights. Darker colors mean higher particle weights.

TABLE 7 Experimental Results on the AVRI Dataset

Model	Modality	MAE (°)	Accuracy (%)	Model Size (M)
STFT-ResNet [43]	audio	17.21	67.63	6.315
GCC-MLP [5]	audio	19.03	66.00	2.604
AV-MLP [8]	audio-visual	17.55	68.78	2.647
A-CRNN [7]	audio	7.91	79.28	7.713
AV-CRNN [42]	audio-visual	7.58	79.72	7.715
CMAF [42]	audio-visual	7.26	80.86	3.825
Ours	audio	8.70	78.05	0.298

The Baseline Results are Imported from [42].

and CMAF [42] use additional facial features, and achieve marginal performance improvement. Our light-weighted model only takes GCCPHAT as input, which reduces the computational complexity and strikes a balance between performance and model complexity.

VI. CONCLUSION AND FUTURE WORK

We have presented a new end-to-end model for speaker tracking by leveraging the conventional particle filter and the transformer based learning architecture. The particle filter provides potential explainability for the transformer, while the transformer offers a strong measurement model for the particle filter. This combination abandons the traditional pattern of tracking, which first extracts measurements and then feeds the measurements to the Bayesian filter. Instead, it provides an end-to-end differentiable architecture. Experiments on the simulated and real datasets show that the proposed model offers improved modeling capacity and robustness to long sequence and noise. However, there are limitations with the proposed method. On the one hand, when the algorithm is used for multi-speaker tracking, the maximum number of speaker needs to be specified, which is unknown in some scenarios. On the other hand, the model performance degrades in multi-speaker scenarios due to the limitations of GCC-PHAT and the existence of data association steps. In the future, we will improve the performance of the proposed method for multi-speaker tracking. Motivated by insights in [44], the resampling step can be optimized further, such as leveraging the parallel processing to reduce the computational cost and adjust the sampling frequency according to the degree of weight degeneracy to mitigate the problem of sample impoverishment.

ACKNOWLEDGMENT

The authors thank the reviewers and associate editor for their helpful comments.

REFERENCES

[1] J. Ong, B. T. Vo, S. Nordholm, B.-N. Vo, D. Moratuwage, and C. Shim, "Audio-visual based online multi-source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1219–1234, 2022.

[2] D. Liu, D. Kiecza, A. Srivastava, and F. Kubala, "Online speaker adaptation and tracking for real-time speech recognition," in *Proc. 9th Eur. Conf. Speech Commun. Technol.*, 2005, pp. 281–284.

[3] V. P. Minotto, C. R. Jung, and B. Lee, "Multimodal multi-channel online speaker diarization using sensor fusion through SVM," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1694–1705, Oct. 2015.

[4] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[5] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 74–79.

[6] X. Qian, A. Brutti, O. Lanz, M. Omologo, and A. Cavallaro, "Multi-speaker tracking from an audio–visual sensing device," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2576–2588, Oct. 2019.

[7] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2019, pp. 30–34.

[8] X. Qian, M. Madhavi, Z. Pan, J. Wang, and H. Li, "Multi-target doa estimation with an audio-visual fusion mechanism," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 4280–4284.

[9] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," in *Proc. Robotics: Sci. Syst.*, 2018.

[10] X. Qian, A. Brutti, M. Omologo, and A. Cavallaro, "3D audio-visual speaker tracking with an adaptive particle filter," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2896–2900.

[11] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2020.

[12] D. Yuan et al., "Active learning for deep visual tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2023, doi: 10.1109/TNNLS.2023.3266837.

[13] X. Qian, A. Xompero, A. Cavallaro, A. Brutti, O. Lanz, and M. Omologo, "3D mouth tracking from a compact microphone array co-located with a camera," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 3071–3075.

[14] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez, "AV16.3: An audio-visual corpus for speaker localization and tracking," in *Proc. Mach. Learn. Multimodal Interaction: 1st Int. Workshop*, 2005, pp. 182–195.

[15] X. Qian, A. Brutti, O. Lanz, M. Omologo, and A. Cavallaro, "Audio-visual tracking of concurrent speakers," *IEEE Trans. Multimedia*, vol. 24, pp. 942–954, 2021.

[16] H. Liu, Y. Li, and B. Yang, "3D audio-visual speaker tracking with a two-layer particle filter," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1955–1959.

[17] M. A. M. Izhar, M. Volino, A. Hilton, and P. Jackson, "Tracking sound sources for object-based spatial audio in 3D audio-visual production," in *Proc. Forum Acusticum*, 2020, pp. 2051–2058.

[18] V. Kılıç, M. Barnard, W. Wang, A. Hilton, and J. Kittler, "Mean-shift and sparse sampling-based SMC-PHD filtering for audio informed visual speaker tracking," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2417–2431, Dec. 2016.

[19] Y. Liu, V. Kılıç, J. Guan, and W. Wang, "Audio–visual particle flow SMC-PHD filtering for multi-speaker tracking," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 934–948, Apr. 2020.

[20] Y. Liu, A. Hilton, J. Chambers, Y. Zhao, and W. Wang, "Non-zero diffusion particle flow SMC-PHD filter for audio-visual multi-speaker tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4304–4308 .

[21] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-bernoulli mixture filter: Direct derivation and implementation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1883–1901, Aug. 2018.

[22] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop KF: Learning discriminative deterministic state estimators," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4376–4384.

[23] C. Schymura et al., "A dynamic stream weight backprop Kalman filter for audiovisual speaker tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 581–585.

[24] H. Wen, X. Chen, G. Papagiannis, C. Hu, and Y. Li, "End-to-end semi-supervised learning for differentiable particle filters," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5825–5831.

[25] P. Karkus, S. Cai, and D. Hsu, "Differentiable SLAM-net: Learning particle SLAM for visual navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2815–2825.

[26] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," in *Proc. Conf. Robot Learn.*, 2018, pp. 169–178.

[27] X. Ma, P. Karkus, D. Hsu, and W. S. Lee, "Particle filter recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5101–5108.

[28] M. Zhu, K. Murphy, and R. Jonschkowski, "Towards differentiable resampling," 2020, *arXiv:2004.11938*.

[29] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[31] Z. Wang, X. Zhao, H. Rong, Y. Tong, and J. Shi, "Microphone array-based sound source localization using convolutional residual network," *J. New Media*, vol. 4, no. 3, 2022, Art. no. 145.

[32] J. Zhao and C. Ritz, "Adapting GCC-PHAT to co-prime circular microphone arrays for speech direction of arrival estimation using neural networks," in *Proc. IEEE Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2022, pp. 815–819.

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[34] M. Yu, C. Zhang, Y. Xu, S. Zhang, and D. Yu, "MetricNet: Towards improved modeling for non-intrusive speech quality assessment," in *Proc. Interspeech*, 2021, pp. 2142–2146.

[35] I. D. Gebru, S. Ba, X. Li, and R. Horaud, "Audio-visual speaker diarization based on spatiotemporal Bayesian fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1086–1099, May 2018.

[36] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba, "Self-supervised moving vehicle tracking with stereo sound," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7053–7062.

[37] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9701–9707.

[38] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[39] J. Thiemann, N. Ito, and E. Vincent, "Demand: A collection of multi-channel recordings of acoustic noise in diverse environments," in *Proc. Meetings Acoust.*, 2013, pp. 1–6.

[40] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 433–440.

[41] J. Zhao et al., "Audio-visual tracking of multiple speakers via a PMBM filter," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5068–5072.

[42] X. Qian, Z. Wang, J. Wang, G. Guan, and H. Li, "Audio-visual cross-attention network for robotic speaker tracking," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 550–562, 2022.

[43] W. He, P. Motlicek, and J.-M. Odobez, "Adaptation of multiple sound source localization neural networks with weak supervision and domain-adversarial training," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 770–774.

[44] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.