Audio source separation comes with the need to devise multichannel filters that can exploit priors about the target signals. In that context, experience shows that modeling magnitude spectra is effective. However, devising a probabilistic model on complex spectral data with a prior on magnitudes is non trivial, because it should both reflect the prior but also be tractable for easy inference. In this paper, we approximate the ideal donut-shaped distribution of a complex variable with approximately known magnitude as a Gaussian mixture model called BEADS (Bayesian Expansion Approximating the Donut Shape) and show that it permits straightforward inference and filtering while effectively constraining the magnitudes of the signals to comply with the prior. As a result, we demonstrate large improvements over the Gaussian baseline for multichannel audio coding when exploiting the BEADS model.

Index Terms— audio probabilistic model, magnitude, phase, source separation

I. INTRODUCTION

Audio source separation aims at processing an audio mixture $x$ composed of $I$ channels (e.g. stereo $I = 2$) so as to recover its $J$ constitutive sources $s_j$ [1]. It has many applications, including karaoke [2], upmixing [3], [4] and speech enhancement [5]. Usually, the sources are recovered by some time-varying filtering of the mixture, which is amenable to applying a $I \times I$ complex matrix $G(f,t)$ to each entry $x(f,t)$ of its Short-Term Fourier Transform (STFT) [6], [7], [3]. Devising good filters requires either dedicated models for sources power spectrograms [8], [9] or machine learning methods to directly predict effective filters [10].

The theoretical grounding for these linear filtering procedures boil down to considerations about second-order statistics for the sources STFT coefficients $s_j(f,t) \in \mathbb{C}$. When seen from a probabilistic perspective, this is often translated as picking a complex isotropic Gaussian distribution [11] for each $s_j(f,t) \in \mathbb{C}$, which is called the Local Gaussian Model (LGM) [12], [13] and is depicted on Figure 1a.

Although it enjoys tractability and easy inference, the main shortcoming of the LGM is that it gives highest probability to 0, which may appear as counterintuitive. Still, this comes as a consequence of the fact that stationarity makes all phases to be equally probable a priori, so that $\mathbb{E}[s_j(f,t)] = 0$. Combining the two first moments only, maximum entropy principles naturally lead us to pick the LGM [14].

However, this badly reflects the additional prior knowledge one often has on $s_j(f,t)$. The vast majority of methods use priors on its magnitude: $|s_j(f,t)|$ should be close to...
II. PROBABILISTIC MODEL

II-A. BEADS source model

The BEADS model is expressed as follows:

\[
\mathbb{P}[s_{j} (f, t) \mid b_{j} (f, t), \sigma_{j} (f, t)] = \sum_{c \in C} \pi_{j} (c \mid f, t) \mathcal{N} (b_{j} (f, t) \omega_{c}, \sigma_{j} (f, t)),
\]

where \( \mathcal{N} \) denotes here the complex isotropic Gaussian distribution\(^1\), \( \omega = \exp (i2\pi/C) \) is the \( C \)-th root of unity and \( \pi_{j} (c \mid f, t) \) is the prior for the phase of source \( j \) at time-frequency (TF) bin \( (f, t) \): it indicates the probability that \( s_{j} (f, t) \) is drawn from component \( c \) and hence that its phase is close to \( \omega_{c} \). While some phase unwrapping approach [23] may be used to set this prior, we take it as uniform here. The parameter \( \sigma_{j} (f, t) \) stands for the variance of each component. It may be understood as the expected squared error of our estimate \( b_{j} (f, t) \) for the magnitude. We can note that the LGM is equivalent to \( C = 1 \) and \( b_{j} (f, t) = 0 \), and that many beads tend to the donut shape, as shown in Figure 1d.

Now, we consider the joint prior distribution of the \( J \) independent sources. We need to consider all the \( C^{J} \) possible combinations for the components. Let \( \mathbb{N}_{C} \) be the set of the first \( C \) natural numbers. We write \( z (f, t) = [z_{j} (f, t) \in \mathbb{N}_{C}^{J} \) for the \( J \times 1 \) vector whose \( j \)-th entry \( z_{j} (f, t) \in \mathbb{N}_{C} \) is the actual component drawn for source \( j \). We define \( \pi (c \mid f, t) \) as the probability of each combination:

\[
\forall c \in \mathbb{N}_{C}^{J}, \pi (c \mid f, t) = \mathbb{P}[z (f, t) = c] = \prod_{j=1}^{J} \pi_{j} (c_{j} \mid f, t),
\]

where \( c_{j} \) is the \( j \)-th entry of \( c \in \mathbb{N}_{C}^{J} \). We have \( \sum_{c} \pi (c \mid f, t) = 1 \). The joint prior distribution of the sources is given by:

\[
\mathbb{P}[s (f, t) \mid \Theta] = \sum_{c \in \mathbb{N}_{C}^{J}} \pi (c \mid f, t) \mathcal{N} (\omega^{c} \bullet b (f, t), \sigma (f, t)),
\]

where \( \omega^{c} \) for \( c \in \mathbb{N}_{C}^{J} \) denotes the \( J \times 1 \) vector with entries \( \omega_{c_{j}} \), \( \omega \bullet b \) denotes element-wise multiplication of the vectors \( \omega \) and \( b \) and \( [v] \) denotes the diagonal matrix whose diagonal is the vector \( v \). In words, the prior distribution for the sources under the BEADS model is a Gaussian Mixture Model (GMM), with weights \( \pi (c \mid f, t) \), which is reminiscent but different from the pioneering work in [24] that was limited to one source signal.

II-B. Mixture likelihood and separation

We take the mix as a convolutive mixture with the narrowband assumption, so that \( x (f, t) \approx A (f) s (f, t) \), where \( A (f) \) is the \( I \times J \) mixing matrix at frequency bin \( f \). Further exploiting the BEADS model (1), we get the marginal model of the mixture given all parameters \( \Theta \) as:

\[
\mathbb{P}[x (f, t) \mid \Theta] = \sum_{c \in \mathbb{N}_{C}^{J}} \pi (c \mid f, t) \mathcal{N} (\pi_{c} (f, t), \Sigma_{x} (f, t)),
\]

with

\[
\begin{aligned}
\pi_{c} (f, t) &= A (f) \omega^{c} \bullet b (f, t) \\
\Sigma_{x} (f, t) &= A (f) [\sigma (f, t)] A (f)^{*}.
\end{aligned}
\]

Now, the real advantages of the BEADS model is that we can straightforwardly obtain the joint posterior distribution of the sources as:

\[
\mathbb{P}[s (f, t) \mid x, \Theta] = \sum_{c \in \mathbb{N}_{C}^{J}} \pi (c \mid f, t, x) \mathcal{N} (\mu_{c}, \Sigma_{c}),
\]

where the posterior statistics \( \mu_{c} \) and \( \Sigma_{c} \) for each combination \( c \) of the phases is:

\[
\begin{aligned}
\mu_{c} (f, t) &= G (f, t) (x (f, t) - \pi_{c} (f, t)) + \omega^{c} \bullet b (f, t) \\
\Sigma_{c} (f, t) &= \sum (f, t) = [\sigma (f, t)] - G (f, t) A (f) [\sigma (f, t)],
\end{aligned}
\]

\(^1\)The probability density function for the complex \( I \)-dimensional isotropic Gaussian is \( \mathcal{N} (x \mid \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|}} \exp \left( - (x - \mu)^{\ast} \Sigma^{-1} (x - \mu) \right) \), where \( |\Sigma| \) is the determinant of \( \Sigma \) [11].
Algorithm 1: BEADS decoder to update the phase configuration probabilities and perform separation.

Input: parameters $\Theta$, mixture $x(f,t)$, prior $\pi_j$, $C$.
Initialization: $\Sigma_{\pi} \leftarrow 0$, $\hat{s} \leftarrow 0$

For all $c_n \in \mathbb{N}^J$:
1) $\hat{\pi}(c \mid f,t,x) \leftarrow \pi(c \mid f,t)N(x(f,t) \mid \pi_c(f,t), \Sigma_x(f,t))$
2) $\hat{s}(f,t) \leftarrow \hat{\pi}(c \mid f,t,x)(\omega^c \cdot b(f,t) - \pi_c(f,t))$
3) $\Sigma_{\pi}(f,t) \leftarrow \Sigma_{\pi}(f,t) + \hat{\pi}(c \mid f,t,x)$

Finalization:
1) $\hat{s}(f,t) \leftarrow \sum_{c_n \in \mathbb{N}^J} \pi(c \mid f,t,x)\omega^c \cdot b(f,t) - \pi_c(f,t)$
2) $\hat{s}(f,t) \leftarrow G(f,t) x(f,t)$

with the $J \times I$ Wiener gain $G(f,t)$ defined as:

$$G(f,t) = [\pi(f,t) A(f)^\ast (A(f) [\pi(f,t)] A(f)^\ast)^{-1}].$$

The minimum mean squared error (MMSE) estimate for the sources thus becomes:

$$E[s(f,t) \mid x,\Theta] = G(f,t) x(f,t) + \sum_{c_n \in \mathbb{N}^J} \pi(c \mid f,t,x) (\omega^c \cdot b(f,t) - \pi_c(f,t)).$$

In the case we know the true phases configuration $z(f,t)$ or have estimated one at the decoder, this estimate (8) simplifies to $\hat{s}(f,t) = \mu z(f,t)$.

III. PARAMETERS ESTIMATION

In this section, we only consider the informed case, where the true source signals $s_j(f,t)$ and the mixing matrices $A(f)$ are known at the coder, while the mixture and the parameters $\Theta = \{b, \sigma, A\}$ are known at the decoder.

III-A. Decoder: posterior $\pi(c \mid f,t,x)$ and separation

We show how the probabilities $\pi(c \mid f,t)$ for the phase configurations may be updated to yield their posterior $\pi(c \mid f,t,x)$ that fully exploit the BEADS model for constraining the phases. Dropping the $(f,t)$ indices for readability, we have:

$$\forall c \in \mathbb{N}^J, \pi(c \mid x) = \frac{\pi(c)N(x \mid \pi_c, \Sigma_x)}{\mathbb{P}[x]}.$$ (9)

This posterior probability may hence be expressed up to a normalizing constant independent of $c$ as:

$$\pi(c \mid x) \propto \hat{\pi}(c \mid x) = \pi(c) \exp\left(- (x - \pi_c)^\ast \Sigma_x^{-1} (x - \pi_c)\right),$$ (10)

which can straightforwardly be computed at the decoder with known parameters $\Theta = \{b, \sigma\}$. $\pi(c \mid x)$ is obtained by normalization after computation for all $c \in \mathbb{N}^J$.

Algorithm 1 summarizes the computations done at the decoder to perform estimation of the posterior probabilities for the phase configurations and separation.

III-B. Coder: amplitudes $b$ and errors $\sigma$

The parameters to be learned at the coder are the amplitude priors $b_j(f,t)$ and the error models $\sigma_j(f,t)$. First, for saving bitrate and computing time, we only use BEADS for the $F_0$ frequency bands that have the highest energy in the mix, let them be $F_0$, and simply pick $b_j(f,t) = 0$ for others. Then, for $f \in F_0$ we compress them by picking a Nonnegative Tensor Factorization model [25]:

$$b_j(f,t) = \begin{cases} \sum_k W_{b,j}(k) H(t,k) Q_{b}(j,k) & \text{if } f \in F_0 \\ 0 & \text{otherwise} \end{cases}$$

in which case $\Theta_b = \{W_{b,j}, H, Q_b\}$ are small nonnegative $F_0 \times K, T \times K$ and $J \times K$, respectively. Same thing for $\sigma$, where we further reduce the number of parameters of the model by taking the same activations $H$ in both cases, but this time, we model all frequencies.

In this section and for convenience, we will assume that the component $z_j(f,t)$ drawn for each source at each TF bin is known and equal to the one closest to $s_j(f,t)$. This simplification has the advantage of strongly reducing the computational cost of the estimation algorithm. Indeed, the BEADS model (1) then reduces to:

$$s_j(f,t) \mid z_j \sim \mathcal{N}\left(b_j(f,t) \omega^{z_j(f,t)}, \sigma_j(f,t)\right).$$

We can define the relative source $\epsilon_j(f,t)$:

$$\epsilon_j(f,t) \overset{\text{def}}{=} s_j(f,t) \left(\omega^{z_j(f,t)}\right)^\ast \sim \mathcal{N}(b_j(f,t), \sigma_j(f,t)).$$ (12)

Provided $z_j(f,t)$ is correctly chosen as the component whose argument $2\pi z_j(f,t)$ is the closest to that of $s_j(f,t)$, we can furthermore safely assume that the real part $\Re(\epsilon_j(f,t))$ of $\epsilon_j(f,t)$ is nonnegative.

Now, we detail the learning procedure we propose for $b$ and $\sigma$. The strategy is to alternatively fix each of them and learn the other one.

Learning $b_j(f,t)$: Assume $\sigma_j(f,t)$ is kept fixed. The distribution (12) for the relative sources mean that we may estimate $\Theta_b$ using a weighted Euclidean method:

$$\Theta_b \leftarrow \arg\min_{\Theta_b} \sum_{f,t,j} \frac{|\Re(\epsilon_j(f,t)) - b_j(f,t)\theta_j(f,t)|^2}{\sigma_j(f,t)},$$

which can be done with a classical weighted NTF [26] scheme with the Euclidean cost function.

Learning $\sigma_j(f,t)$: If $b_j(f,t)$ is fixed we see from (12) that $\epsilon_j(f,t) - b_j(f,t)$ has an isotropic complex Gaussian distribution with variance $\sigma_j(f,t)$. This means $\Theta_\sigma$ can be estimated through:

$$\Theta_\sigma \leftarrow \arg\min_{\Theta_\sigma} \sum_{f,t,j} d_{IS}\left(|\epsilon_j(f,t) - b_j(f,t)|^2 \parallel \sigma_j(f,t)\theta_j(f,t)\right),$$

where $d_{IS}(a \parallel b)$ is the classical Itakura-Saito divergence for two nonnegative scalars $a$ and $b$. This optimization is
classical in the audio processing literature [27], [28], [29]. Considering that the activations $H$ we take for $\sigma$ are those for $b$, we only learn $W_\sigma$ and $Q_\sigma$ with it.

IV. EVALUATION

We evaluate the BEADS model through its performance for ISS, i.e. by displaying its average quality as a function of the bitrate required to transmit its parameters. To assess quality, we use BSSeval metrics [30]: SDR (Source to Distortion Ratio) and SIR (Source to Interference Ratio), both expressed in dB and higher for better separation. For normalization purpose, we compute $\delta$-metrics, defined as the difference between the score and performance of oracle Wiener filtering, i.e. using true sources spectrograms [31].

The data consists of 10 excerpts of 30 s, taken from DSD100 database\(^2\). Each consists of $J = 4$ sources (vocals, bass, drums and accompaniment), sampled at 44.1 kHz. We generated either mono ($I = 1$) or stereo ($I = 2$) mixtures from these sources, through simple summation or anechoic mixing (delays+gains), respectively. STFT was conducted with 50 % overlap and a window size of 93 ms.

We evaluated the following methods:

- **BEADS oracle**: $\hat{s}(f,t) = \mu_{\hat{z}(f,t)}$.
- **BEADS point** using only the phase configuration $\hat{z}$ that is most likely a posteriori.
- **BEADS** as given in Algorithm 1.
- **Itakura-Saito NTF** [32], with $K$ components.

Given all these methods and data, our extensive evaluation consisted in trying the methods with $F_0 = 150$ frequency bands for BEADS magnitudes, $C \in \{8, 16\}$ beads, and all methods were tried for $K \in [8, 128]$ NTF components. We picked 16 quantization levels for all parameters. Results were smoothed using LOESS [33] and are displayed on Figure 2.

An interesting fact we see on Figure 2 is that the oracle BEADS model significantly outperforms standard oracle Wiener filtering, even for very crude magnitude models $b_j(f,t)$. This can be seen by the fact that its $\delta$-metrics get positive even at very small bitrates.

Then, we may notice that the $\delta$-metrics appear as higher for mono than for stereo mixtures. In this respect, we should highlight that the absolute performance of oracle Wiener is of course higher for stereo (not shown on Figure 2), due to the knowledge of the mixing filters $A(f) \in C$ that alone bring good separation already and actually some information about the phase of the sources. Adding additional spectral knowledge in that case is then less important than in the mono case, where it is crucial.

Now, we see a very clear improvement of BEADS as described in Algorithm 1 over classical NTF-ISS, of approximately 2 dB SDR and 5 dB SIR, at most bitrates. This significant boost in performance shows that BEADS helps a lot in predicting the source signals by adequately handling priors on magnitudes, which is the main result for this study.

Finally, Figure 2 also shows that the procedure for computing the phase posterior probabilities is not sufficient for correctly identifying the true phase configuration. This can be seen by the strong discrepancies between the BEADS point estimate and its oracle performance. While marginalization over the configuration as described in Algorithm 1 helps a lot in this respect, there is much room for improvement for parameter estimation of this model.

V. CONCLUSION

In this paper, we introduced BEADS as a convenient probabilistic model for complex data whose magnitude is approximately known. BEADS is a Gaussian Mixture Model where all components share the same variance and are scattered along a circle. While simple conceptually, BEADS comes with several advantages. First, it translates the delicate problem of modeling the phase into setting probabilities over a discrete set of components. Second, it allows for easy inference and, finally, it straightforwardly leads to effective filtering procedures. Although we demonstrated its potential in an audio-coding application, we believe it may also be useful in the blind separation setting when embedded in an Expectation-Maximization estimation procedure.

\(^2\)http://sisec.inria.fr.
VI. REFERENCES


