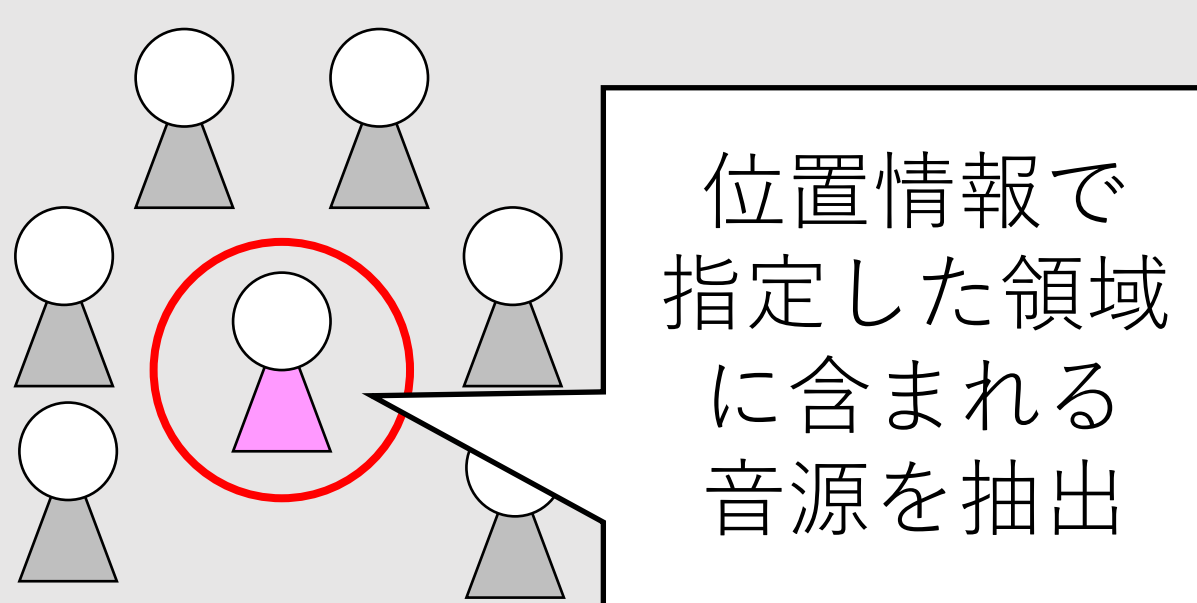


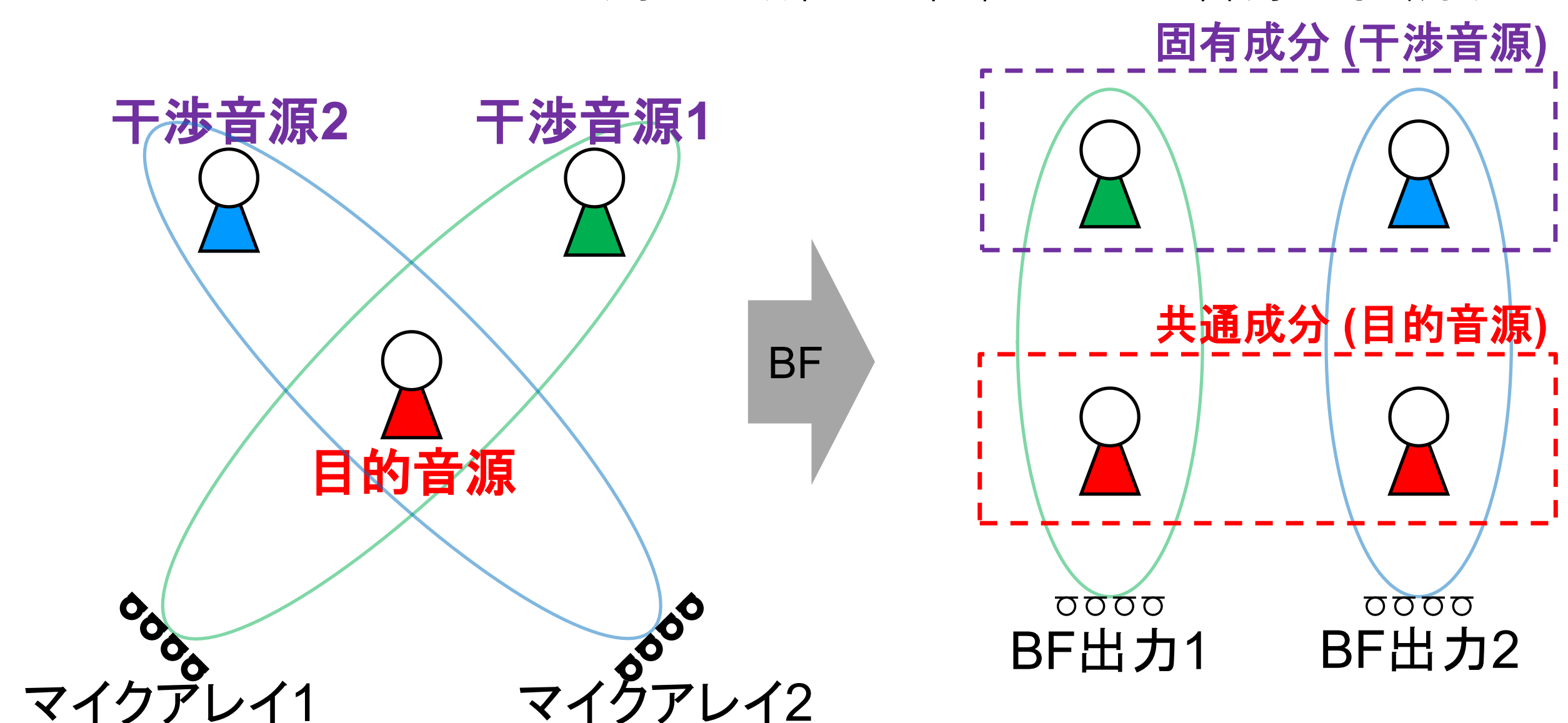
## 本発表の概要

- 空間上の特定の領域中にある目的音源を抽出するスポットフォーミングでは、分散配置したマイクロホンアレイと非負値テンソル因子分解を用いた手法を提案した[Ayano+, 2024]
- 提案手法において、制約付きの補助関数を設計し更新式を導出した
- ハイパーパラメータに特殊な値を用いることで更に高速に動作する更新式を導出した
- 実験では提案手法が高速に動作し、高いスポットフォーミング性能を得ることができることを確認



## 1. 研究背景

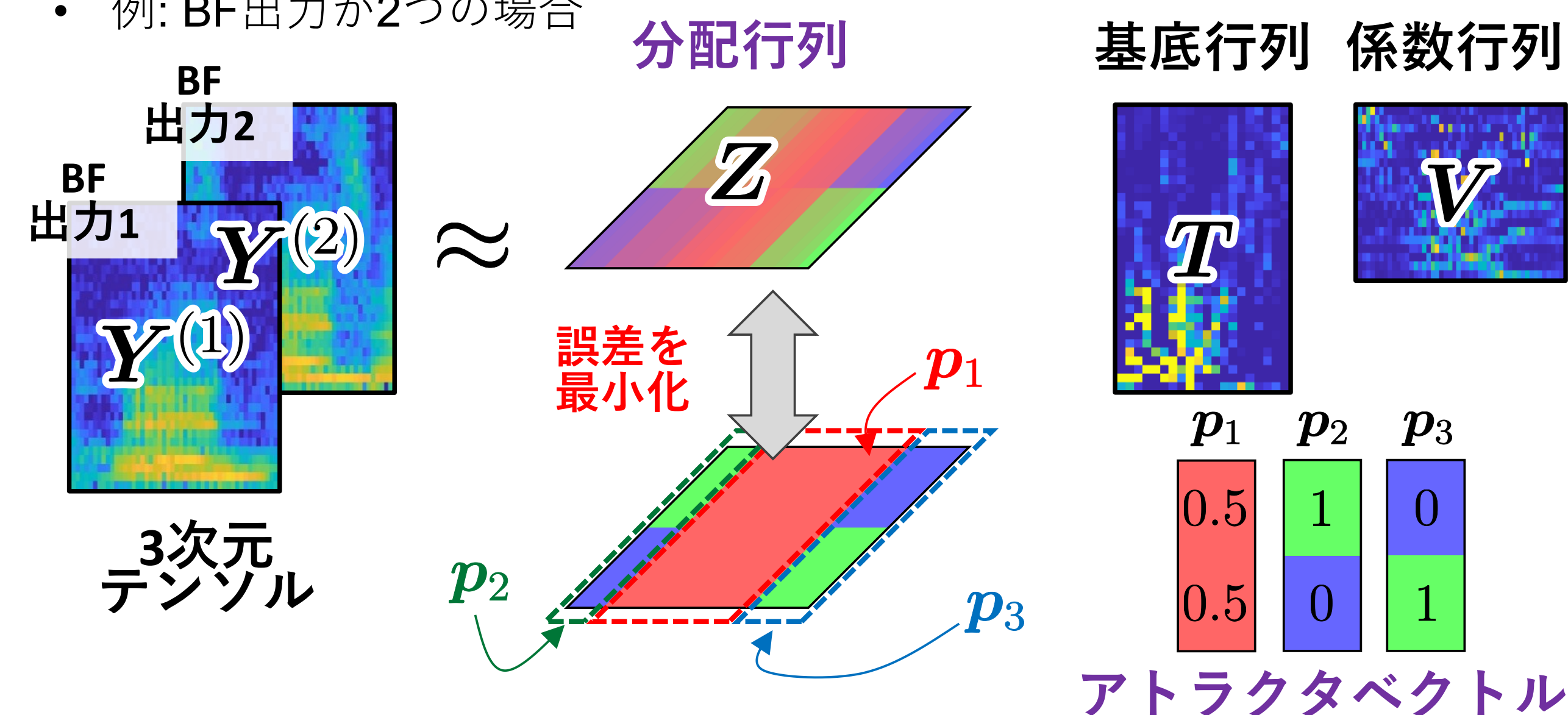
- ビームフォーミング (beamforming: BF)**
  - マイクロホンアレイを用いて所望の方向にある音源を強調する技術
- スポットフォーミング**
  - マイクロホンアレイを用いて所望の位置にある音源を強調する技術



- 分散してマイクロホンを配置し適切な方向にBFを行うことで、すべてのBF出力に共通の目的音源と固有の干渉音源が現れる
- 共通成分を抽出することで目的音源の抽出が可能
- 共通成分の抽出に非負値テンソル因子分解 (nonnegative tensor factorization: NTF) を用いた手法を提案している

### アトラクタ正則化付きNTFによる共通成分抽出

- BF出力 (3階テンソル) を基底行列, 係数行列, および分配行列の3つの非負行列に分解
- 分配行列の各列がアトラクタベクトル (ワンホットベクトルまたは中央ベクトル) に誘導する正規化 (アトラクタ正則化) を導入
  - 例: BF出力が2つの場合



- 中央ベクトルに引き寄せられた成分のみを抽出することでスポットフォーミングを実現

## 2. NTFの更新式の導出

- 以下の最適化問題で表されるNTFを用いる

$$\underset{Z, T, V}{\text{minimize}} \mathcal{J} \quad \text{s.t. } Z \in \{Z \in [0, 1]^{A \times K} \mid \|z_k\|_1 = 1\}, T \in \{T \in [0, 1]^{I \times K} \mid \|t_k\|_1 = 1\}, V \in \mathbb{R}_+^{J \times K}$$

$$\mathcal{J} = \sum_{a,i,j} \left( -c_{a,i,j} \log \sum_k z_{a,k} t_{i,k} v_{j,k} + \sum_k z_{a,k} t_{i,k} v_{j,k} \right) + \mu \sum_{a,k} (-p_{a,b_k} \log z_{a,k} + z_{a,k})$$

- 分配行列および基底行列の各列の和を1とする

- 補助関数と補助変数を設計し、補助関数を最小化

- 補助変数を用いて上記コスト関数の上界補助関数を設計し、最小化

$$\underset{Z, T, V, U}{\text{minimize}} \mathcal{J}^+ \quad \text{s.t. } Z \in \mathbb{R}_+^{A \times K}, T \in \mathbb{R}_+^{I \times K}, V \in \mathbb{R}_+^{J \times K}, U \ni u_{a,i,j,k} \forall a, i, j, k$$

$$\mathcal{J}^+ = \sum_{a,i,j} \left( -c_{a,i,j} \sum_k u_{a,i,j,k} \log \frac{z_{a,k} t_{i,k} v_{j,k}}{u_{a,i,j,k}} + \sum_k z_{a,k} t_{i,k} v_{j,k} \right) + \mu \sum_{a,k} (-p_{a,b_k} \log z_{a,k} + z_{a,k})$$

- $\mathcal{J}^+$ を各変数で微分する (例: 分配行列での微分)

$$\frac{\partial \mathcal{J}^+}{\partial z_{a,k}} = \sum_{i,j} \left( -\frac{c_{a,i,j} u_{a,i,j,k}}{z_{a,k}} + t_{i,k} v_{j,k} \right) + \mu \sum_{a,k} \left( -\frac{p_{a,b_k}}{z_{a,k}} + 1 \right)$$

- 微分が0となるように変数を決定

$$z_{a,k} = \frac{\sum_{i,j} c_{a,i,j} u_{a,i,j,k} + \mu p_{a,b_k}}{\sum_{i,j} t_{i,k} v_{j,k} + \mu}$$

- 補助変数に適切な値を代入し、正規化することで更新式を導出

- 反復更新式

- これらの更新を反復的に行い、もとの最適化問題を最小化

$$z_{a,k} \leftarrow \frac{z_{a,k} \sum_{i,j} c_{a,i,j} \frac{t_{i,k} v_{j,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}} + \mu p_{a,b_k}}{\sum_{i,j} t_{i,k} v_{j,k} + \mu}$$

$$v_{j,k} \leftarrow v_{j,k} \|z_k\|_1^{-1}$$

$$z_{a,k} \leftarrow z_{a,k} \|z_k\|_1^{-1}$$

$$t_{i,k} \leftarrow \frac{t_{i,k} \sum_{a,j} c_{a,i,j} \frac{z_{a,k} v_{j,k}}{\sum_{a,k'} z_{a,k'} t_{i,k'} v_{j,k'}}}{\sum_{a,j} z_{a,k} v_{j,k}}$$

$$v_{j,k} \leftarrow v_{j,k} \|t_k\|_1^{-1}$$

$$t_{i,k} \leftarrow t_{i,k} \|t_k\|_1^{-1}$$

$$v_{j,k} \leftarrow \frac{v_{j,k} \sum_{a,i} c_{a,i,j} \frac{z_{a,k} t_{i,k}}{\sum_{a,i} z_{a,k} t_{i,k}}}{\sum_{a,i} z_{a,k} t_{i,k}}$$

$$v_{j,k} \leftarrow \frac{v_{j,k} \sum_{a,i} c_{a,i,j} \frac{z_{a,k} t_{i,k}}{\sum_{a,i} z_{a,k} t_{i,k}}}{\sum_{a,i} z_{a,k} t_{i,k}}$$

分配行列および基底行列の更新後、係数行列を用いて正規化を行う

## 3. 高速化手法

- 設計する補助関数に対して各行列の制約を導入

$$\underset{Z, T, V, U}{\text{minimize}} \mathcal{J}^+ \quad \text{s.t. } Z \in \{Z \in [0, 1]^{A \times K} \mid \|z_k\|_1 = 1\}, T \in \{T \in [0, 1]^{I \times K} \mid \|t_k\|_1 = 1\}, V \in \mathbb{R}_+^{J \times K}, U \ni u_{a,i,j,k} \forall a, i, j, k$$

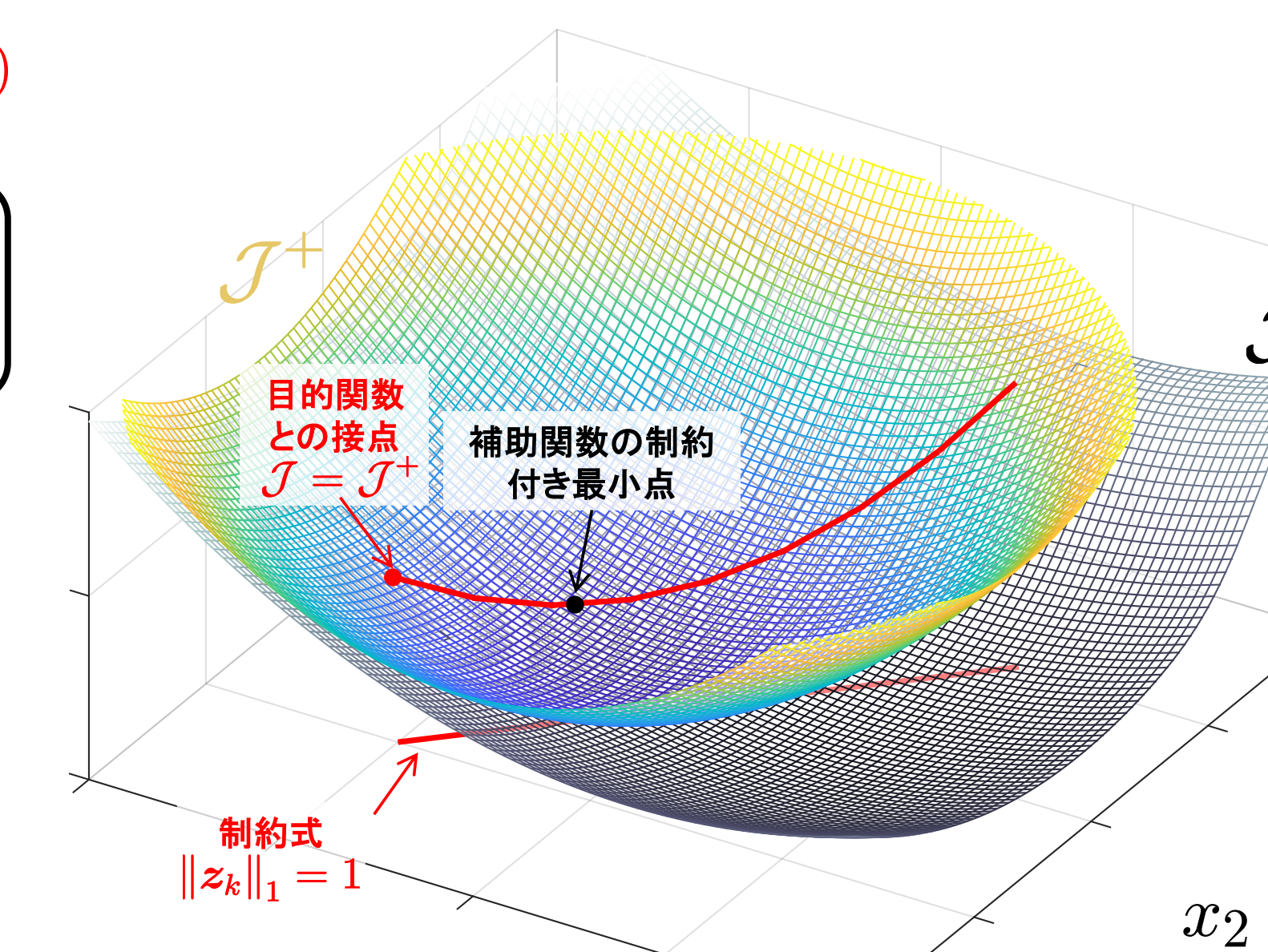
$$\mathcal{J}^+ = \sum_{a,i,j} \left( -c_{a,i,j} \sum_k u_{a,i,j,k} \log \frac{z_{a,k} t_{i,k} v_{j,k}}{u_{a,i,j,k}} + \sum_k z_{a,k} t_{i,k} v_{j,k} \right) + \mu \sum_{a,k} (-p_{a,b_k} \log z_{a,k} + z_{a,k})$$

- ラグランジュの未定定数法を用いて制約付きの解を導出

$$\mathcal{L}(Z, \lambda) = \mathcal{J}^+ + \sum_k \lambda_k (\|z_k\|_1 - 1)$$

$\mathcal{J}^+$ に対してL1制約を課した $\mathcal{L}$ を解く

- $\mathcal{L}$ を各変数で微分し、すべての式が0となるように変数を決定
- 補助関数に適切な値を代入し、更新式を導出



- 反復更新式

$$z_{a,k} \leftarrow z_{a,k} \sum_{i,j} c_{a,i,j} \frac{t_{i,k} v_{j,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}} + \mu p_{a,b_k}$$

$$z_{a,k} \leftarrow z_{a,k} \|z_k\|_1^{-1}$$

$$t_{i,k} \leftarrow t_{i,k} \sum_{a,j} c_{a,i,j} \frac{z_{a,k} v_{j,k}}{\sum_{a,k'} z_{a,k'} t_{i,k'} v_{j,k'}}$$

$$t_{i,k} \leftarrow t_{i,k} \|t_k\|_1^{-1}$$

$$v_{j,k} \leftarrow v_{j,k} \sum_{a,i} c_{a,i,j} \frac{z_{a,k} t_{i,k}}{\sum_{a,k'} z_{a,k'} t_{i,k'} v_{j,k'}}$$

係数行列に対する正規化が不要  
コスト関数の単調非増加性を保持

- スポットフォーミングにおいては大きい $\mu$ を用いることで高い性能を実現

- $\mu \rightarrow \infty$ とすることを考えると $z_{a,k}$ は $p_{a,b_k}$ と一致する

$$z_{a,k} \leftarrow p_{a,b_k}$$

$$t_{i,k} \leftarrow t_{i,k} \sum_{a,j} c_{a,i,j} \frac{z_{a,k} v_{j,k}}{\sum_{a,k'} z_{a,k'} t_{i,k'} v_{j,k'}}$$

$$t_{i,k} \leftarrow t_{i,k} \|t_k\|_1^{-1}$$

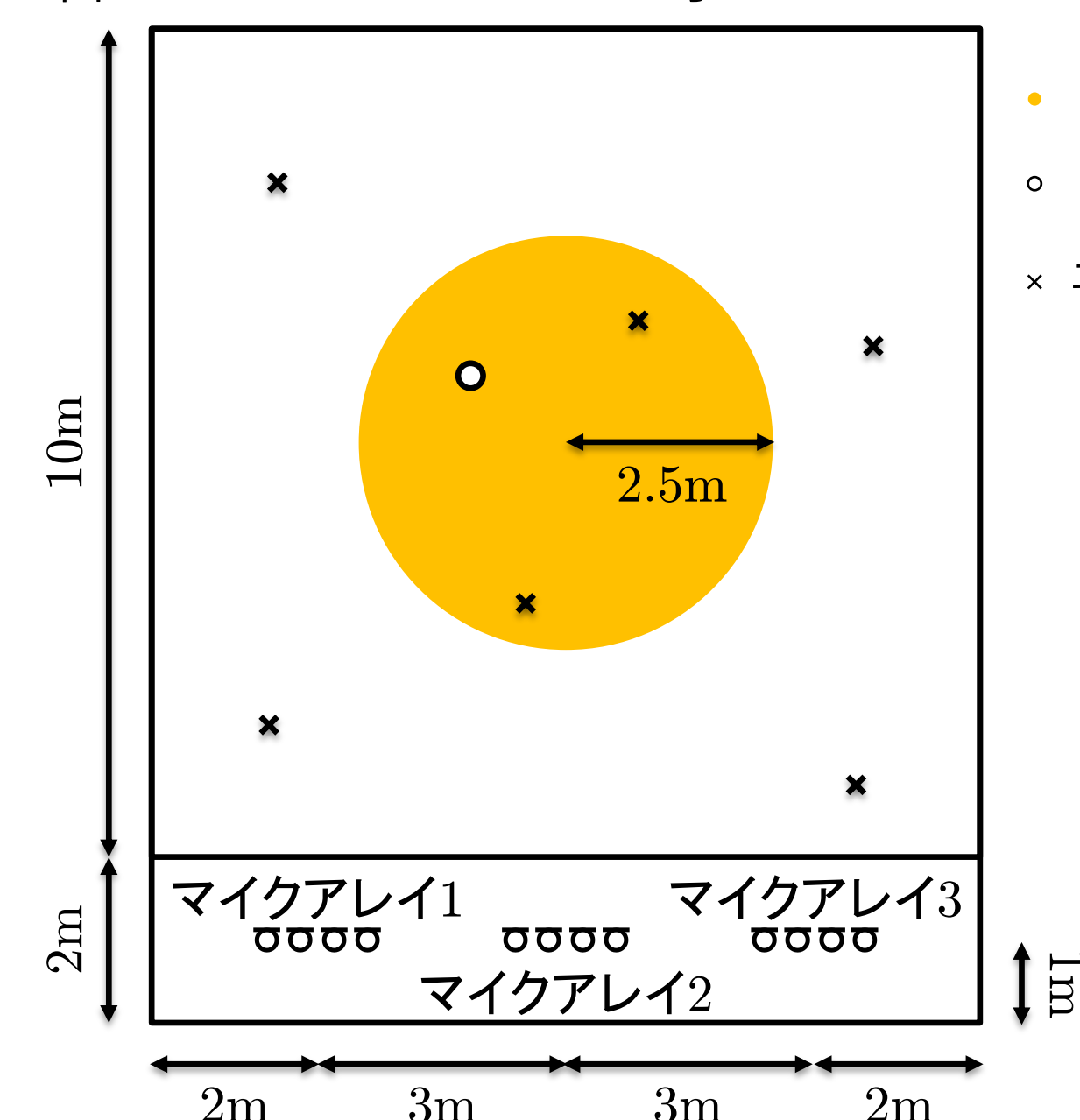
$$v_{j,k} \leftarrow v_{j,k} \sum_{a,i} c_{a,i,j} \frac{z_{a,k} t_{i,k}}{\sum_{a,k'} z_{a,k'} t_{i,k'} v_{j,k'}}$$

分配行列Zの更新式は代入操作となり、約4AIJK回の乗算を削減

## 4. 実験

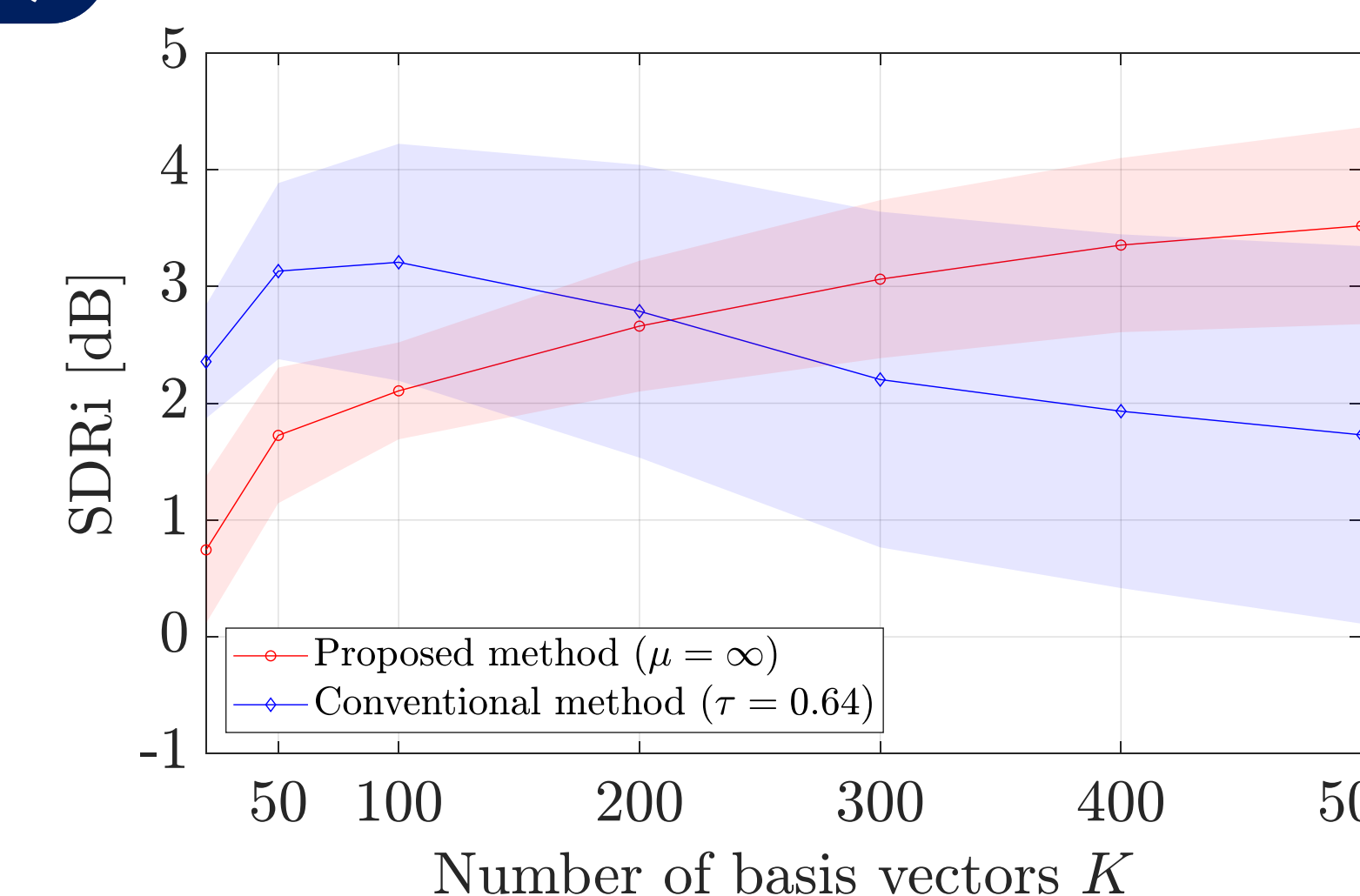
- シミュレーション実験

- 各マイクロホンに対して-40dBのホワイトノイズを付与



- 目的音源領域
- 目的音源
- 干渉音源

条件	値
窓長	256 ms
シフト長	128 ms
反復更新回数	100 回
	$\tau = 0.64$
ハイパーパラメータ	最初の50反復 $\mu = 0$
	最後の50反復 $\mu \rightarrow \infty$
評価スコア	SDR改善量 (SDRi)
	計算時間



$\mu \rightarrow \infty$ とした更新式を用いることで高速に動作し高い性能を維持

