Empowering Speech Processing with Deep Neural Networks: Theory and Applications

Kuan-Lin Chen Advisor: **Bhaskar D. Rao**
Department of Electrical and Computer Engineering, University of California, San Diego †kuc029@ucsd.edu

UC San Diego JACOBS SCHOOL OF ENGINEERING Electrical and Computer Engineering

ICASSP 2023 4 - 10 JUNE, RHODES ISLAND, GREECE

1 Complexity of deep ReLU networks

Theorem 1 (Theorem 1 of [1]). *Any continuous piecewise linear (CPWL) function $p\colon \mathbb{R}^n \to \mathbb{R}$ with q pieces can be represented by a ReLU network whose number of layers l, maximum width w, and number of hidden neurons h satisfy*

$$l \le 2\lceil \log_2 q \rceil + 1, \tag{1}$$

$$w \le \mathbb{I}[q > 1]\left\lceil \frac{3q}{2} \right\rceil q, \tag{2}$$

$$h \le \left(3 \cdot 2^{\lceil \log_2 q \rceil} + 2\lceil \log_2 q \rceil - 3\right) q + 3 \cdot 2^{\lceil \log_2 q \rceil} - 2\lceil \log_2 q \rceil - 3. \tag{3}$$

Furthermore, Algorithm 1 finds such a network in $\mathsf{poly}(n, q, L)$ *time where L is the number of bits required to represent every entry of the rational matrix $\mathbf{A}_i$ in the polyhedron representation $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}_i \mathbf{x} \le \mathbf{b}_i\}$ of the piece $\mathcal{X}_i$ for every $i \in [q]$.*

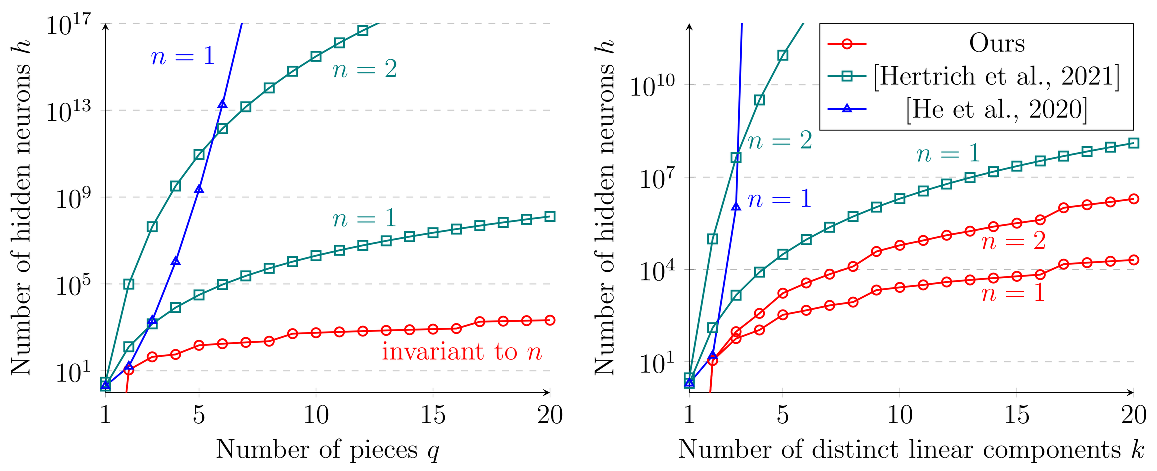


Figure 1: Any CPWL function $\mathbb{R}^n \to \mathbb{R}$ with q pieces or k distinct linear components can be exactly represented by a ReLU network with at most h hidden neurons. The upper bounds (red) in [1] are substantially tighter than existing bounds in the literature, showing that any CPWL function can be exactly realized by a ReLU network at a much lower cost.

Algorithm 1 Find a ReLU network that computes a given CPWL function

Input: A CPWL function p with pieces $\{\mathcal{X}_i\}_{i\in[q]}$ of $\mathbb{R}^n$.
Output: A ReLU network g computing $g(\mathbf{x}) = p(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$.

- $f_1, f_2, \cdots, f_k \leftarrow$ Find all distinct linear components of p
- **for** $i = 1, 2, \cdots, q$ **do**
  - $\mathcal{A}_i \leftarrow \emptyset$
  - **for** $j = 1, 2 \cdots, k$ **do**
    - **if** $f_j(\mathbf{x}) \ge p(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}_i$ **then**
      - $\mathcal{A}_i \leftarrow \mathcal{A}_i \bigcup \{j\}$
    - **end if**
  - **end for**
  - $v_i \leftarrow$ A network representing the min-affine function of $\{f_m\}_{m\in\mathcal{A}_i}$
- **end for**
- $v \leftarrow$ Combine ReLU networks $v_1, v_2, \cdots, v_q$ in parallel
- $u \leftarrow$ A ReLU network computing the maximum of q elements
- $g \leftarrow$ A ReLU network computing the composition $u \circ v$

2 Optimization of deep residual networks

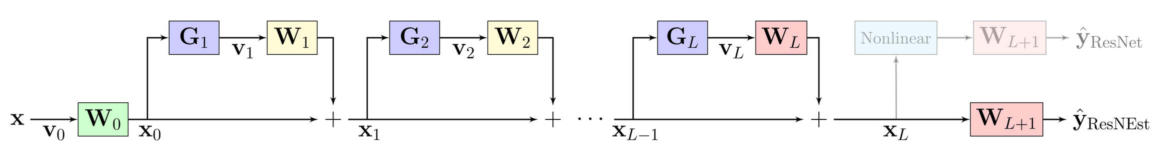


Figure 2: A generic vector-valued ResNEst that has a chain of L residual blocks. Different from the standard ResNet architecture, our ResNEst architecture drops nonlinearities at $\mathbf{x}_L$ so as to reveal a linear relationship between the output $\hat{\mathbf{y}}_{\text{ResNEst}}$ and the features $\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_L$.

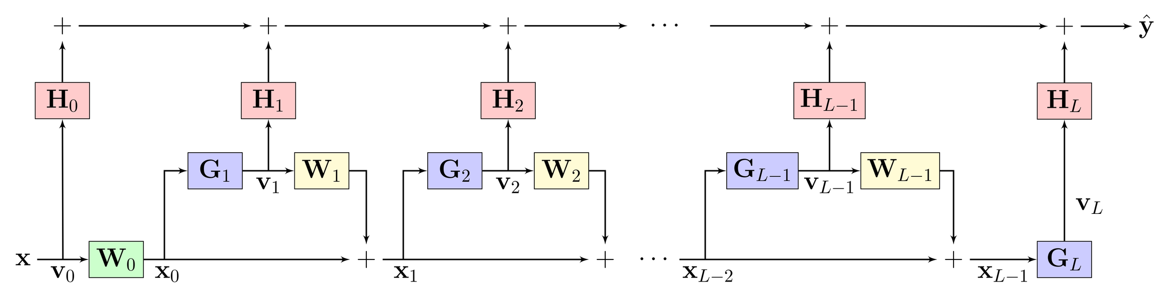


Figure 3: The proposed augmented ResNEst or A-ResNEst.

Because the ResNEst now reveals a linear relationship between the output and the features, we have:

$$\hat{\mathbf{y}}_{L\text{-ResNEst}}(\mathbf{x}) = \mathbf{W}_{L+1}\sum_{i=0}^{L}\mathbf{W}_i \mathbf{v}_i(\mathbf{x}), \tag{4}$$

$$\mathbf{v}_i(\mathbf{x}) = \mathbf{G}_i(\mathbf{x}_{i-1}; \boldsymbol{\theta}_i) = \mathbf{G}_i\left(\sum_{j=0}^{i-1}\mathbf{W}_j\mathbf{v}_j; \boldsymbol{\theta}_i\right). \tag{5}$$

We propose to utilize the basis function modeling point of view in the ResNEst and analyze the following ERM problem:

$$(\mathrm{P}_{\boldsymbol{\phi}}) \min_{\mathbf{W}_L, \mathbf{W}_{L+1}} \mathcal{R}(\mathbf{W}_L, \mathbf{W}_{L+1}; \boldsymbol{\phi}) \tag{6}$$

where $\mathcal{R}(\mathbf{W}_L, \mathbf{W}_{L+1}; \boldsymbol{\phi}) = \frac{1}{N}\sum_{n=1}^{N}\ell\left(\hat{\mathbf{y}}^{\boldsymbol{\phi}}_{L\text{-ResNEst}}(\mathbf{x}^n), \mathbf{y}^n\right)$ for any fixed feature finding weights $\boldsymbol{\phi}$. For A-ResNEst, we put

$$(\mathrm{PA}_{\boldsymbol{\phi}}) \min_{\mathbf{H}_0, \cdots, \mathbf{H}_L} \mathcal{A}(\mathbf{H}_0, \cdots, \mathbf{H}_L; \boldsymbol{\phi}) \tag{7}$$

where $\mathcal{A}(\mathbf{H}_0, \cdots, \mathbf{H}_L; \boldsymbol{\phi}) = \frac{1}{N}\sum_{n=1}^{N}\ell\left(\hat{\mathbf{y}}^{\boldsymbol{\phi}}_{L\text{-A-ResNEst}}(\mathbf{x}^n), \mathbf{y}^n\right)$.

- M is the output dimension of $\mathbf{W}_0$ (expansion factor).
- N_o is the output dimension of the network.

Theorem 2 (Theorem 1 of [2]). *If the loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is differentiable and convex in $\hat{\mathbf{y}}$ for any $\mathbf{y}$ and $M \ge N_o$, then the following two properties are true in ($P_{\boldsymbol{\phi}}$) under any $\boldsymbol{\phi}$ such that the linear inverse problem $\mathbf{x}_{L-1} = \sum_{i=0}^{L-1}\mathbf{W}_i\mathbf{v}_i$ has a unique solution: (a) every critical point with full rank $\mathbf{W}_{L+1}$ is a global minimizer, (b) $\mathcal{R}\left(\mathbf{W}^*_L, \mathbf{W}^*_{L+1}; \boldsymbol{\phi}\right) = \mathcal{A}\left(\mathbf{H}^*_0, \cdots, \mathbf{H}^*_L; \boldsymbol{\phi}\right)$ for every local minimizer $\left(\mathbf{W}^*_L, \mathbf{W}^*_{L+1}\right)$ of ($P_{\boldsymbol{\phi}}$).*

3 Uncertainty in supervised speech enhancement

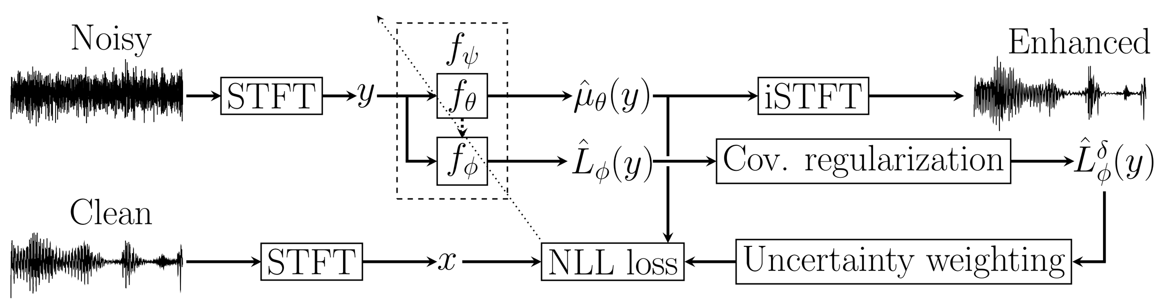


Figure 4: We augment a speech enhancement model f_θ with a temporary submodel f_ϕ to estimate heteroscedastic uncertainty during training [3].

The problem of maximum likelihood is equivalent to minimizing the empirical risk using the multivariate Gaussian NLL loss

$$\ell^{\text{Full}}_{x,y}(\psi) = \left[x - \hat{\mu}_\theta(y)\right]^{\mathsf{T}} \hat{\Sigma}^{-1}_\phi(y) \left[x - \hat{\mu}_\theta(y)\right] + \log\det\hat{\Sigma}_\phi(y). \tag{8}$$

The number of elements in $\hat{\Sigma}_\phi(y)$ is $4T^2F^2$, leading to exceedingly high training complexity. How can we reduce the complexity and make the maximum likelihood tractable?

$$\ell^{\text{Diagonal}}_{x,y}(\psi) = \sum_{t,f}\sum_{k\in\{r,i\}}\left[\frac{x^{t,f}_k - \hat{\mu}^{t,f}_{k;\theta}(y)}{\hat{\sigma}^{t,f}_{k;\phi}(y)}\right]^2 + 2\log\hat{\sigma}^{t,f}_{k;\phi}(y). \tag{9}$$

$$\ell^{\text{Block}}_{x,y}(\psi) = \sum_{t,f}\underbrace{d^{t,f}_{\theta,x}(y)^{\mathsf{T}}\left[\hat{\Sigma}^{t,f}_\phi(y)\right]^{-1} d^{t,f}_{\theta,x}(y) + \log t^{t,f}_\phi(y)}_{z^{t,f}_{x,y}(\psi)}. \tag{10}$$

Covariance regularization. Let $\delta > 0$ be the lower bound of the eigenvalues of the Cholesy factor of the covariance matrix.

$$\left[\hat{L}^\delta_\phi(y)\right]_{mm} = \max\left\{\left[\hat{L}_\phi(y)\right]_{mm}, \delta\right\}. \tag{11}$$

Uncertainty weighting. Let $\beta = 0.5$ and the loss function be a weighted average where the weight of a loss component depends on the *minimum eigenvalue* of the covariance matrix, i.e.,

$$\ell^{\beta\text{-Block}}_{x,y}(\psi) = \sum_{t,f}\lambda_{\min}\left[\hat{\Sigma}^{t,f}_\phi(y)\right]^\beta z^{t,f}_{x,y}(\psi). \tag{12}$$

	WB-PESQ			STOI (%)			SI-SDR (dB)			NORESQA-MOS		
SNR (dB)	-5	0	5	-5	0	5	-5	0	5	-5	0	5
Unprocessed	1.11	1.15	1.24	69.5	77.8	85.2	-5.00	0.01	5.01	2.32	2.36	2.45
MAE	1.50	1.76	2.09	84.4	90.4	93.9	9.83	12.63	15.02	2.77	3.27	3.65
MSE	1.63	1.94	2.29	85.1	90.6	94.0	10.24	13.21	15.97	2.86	3.52	4.02
SI-SDR	1.71	2.04	2.42	86.5	91.5	94.6	**10.96**	**13.92**	**16.80**	3.05	3.65	4.20
NLL ℓ^{Diagonal}	1.74	2.08	2.48	86.2	91.3	94.6	9.83	12.55	15.01	3.14	3.77	4.25
NLL ℓ^{Block}	**1.75**	**2.10**	**2.50**	**86.7**	**91.8**	**94.9**	10.22	13.15	15.99	**3.23**	**3.89**	**4.35**

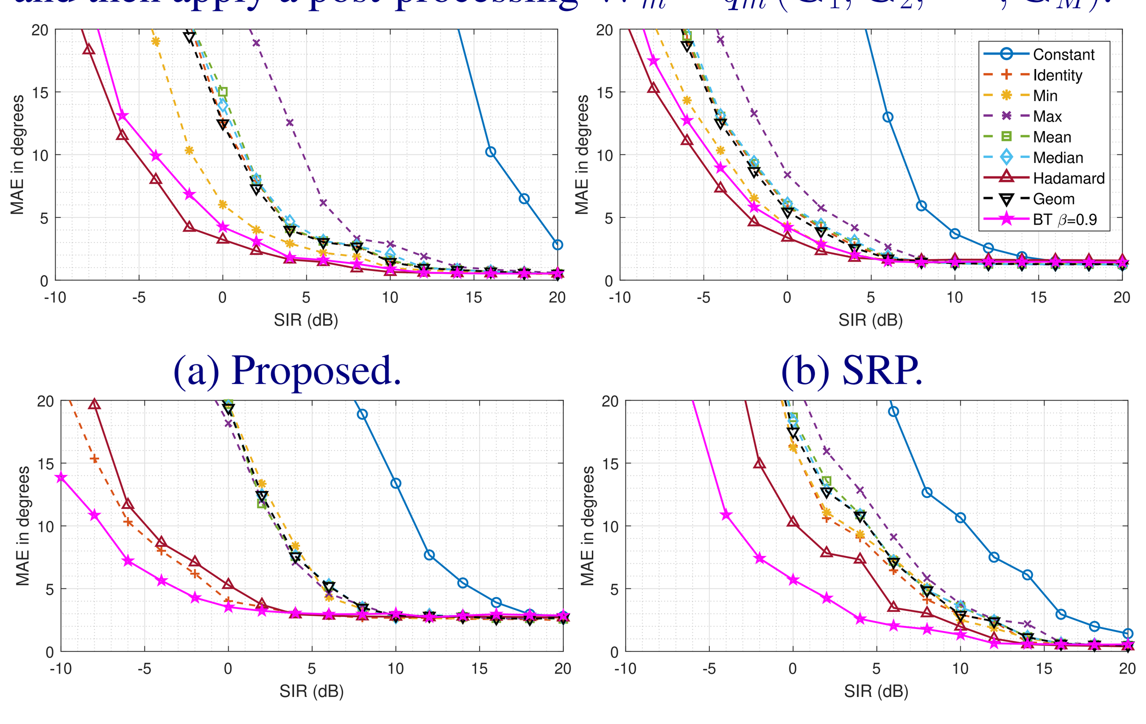
Figure 5: The NLL using a block diagonal covariance with suitable δ and β outperforms the MAE, MSE, and SI-SDR. The DNS dataset is used. We adopt the GCRN as f_θ for investigation. f_ϕ is an additional decoder that takes the output of the in-between LSTM of the GCRN as input.

4 DNN based direction of arrival estimation

Let $\tilde{\mathbf{y}}(t, f) = \mathbf{w}(t, f) \odot \mathbf{y}(t, f)$ be a filtered snapshot. We propose a criterion that normalizes the filtered snapshot [4].

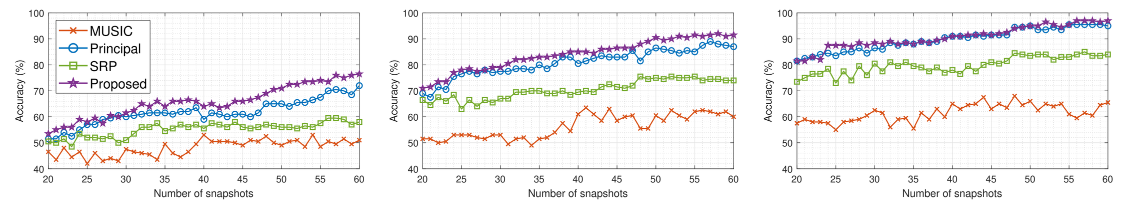
$$\max_\theta \sum_f \mathbf{v}^{\mathsf{H}}(\theta, f)\sum_t \frac{\tilde{\mathbf{y}}(t, f)\tilde{\mathbf{y}}^{\mathsf{H}}(t, f)}{\|\mathbf{y}(t, f)\|^2_2}\mathbf{v}(\theta, f). \tag{13}$$

To find the time-frequency weights $\mathbf{w}(t, f)$, we first use a U-Net (0.67M params) to predict the ideal ratio mask $\mathbf{G}_m$ on each sensor and then apply a post-processing $\mathbf{W}_m = q_m(\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_M)$.



(a) Proposed. (b) SRP. (c) MUSIC. (d) Principal vector.

Figure 6: MAE in degrees vs. SIR. $\mathrm{RT}_{60} = 0.3$s and $\mathrm{SNR} = 20$ dB.



(a) -6 dB SIR. (b) 0 dB SIR. (c) $+6$ dB SIR.

Figure 7: Accuracy vs. number of snapshots. $\mathrm{RT}_{60} = 0.3$s and $\mathrm{SNR} = 20$ dB.

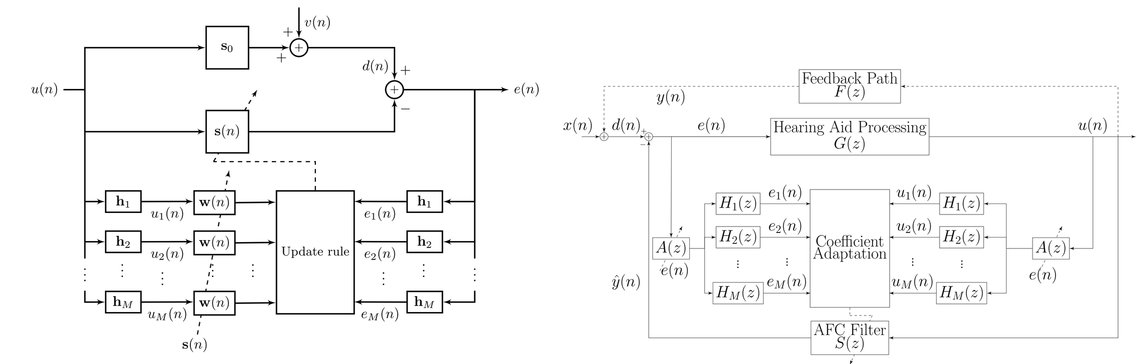
5 Adaptive filters and feedback cancellation

We minimize the sum of the squared error in each subband with a sparsity penalty term. We propose the following cost function:

$$J(\mathbf{s}) = \sum_{i=1}^{M}|e_i(n)|^2 + \tau\|\mathbf{s}\|^2_{\mathbf{W}^{-1}(n)} \tag{14}$$

where $e_i(n) = \mathbf{h}^T_i\mathbf{e}(n) = \mathbf{h}^T_i\left[\mathbf{d}(n) - \mathbf{U}^T(n)\mathbf{s}\right]$ is the i-th subband error and $\mathbf{s} \in \mathbb{R}^L$ is the coefficients of the adaptive filter, leading to the generalized proportionate-type normalized subband adaptive filter (GPtNSAF) [5]: $\mathbf{s}(n+1) = \mathbf{s}(n) + \mu\mathbf{g}(n)$ where

$$\mathbf{g}(n) = \mathbf{W}(n)\mathbf{U}_b(n)\left[\delta\mathbf{I}_M + \mathbf{U}^T_b(n)\mathbf{W}(n)\mathbf{U}_b(n)\right]^{-1}\mathbf{e}_b(n). \tag{15}$$



(a) System identification. (b) Feedback cancellation.

Figure 8: The GPtNSAF and the feedback cancellation framework [6].

We put $\mathbf{W}(n) = \mathrm{diag}\{w_1(n), \cdots, w_L(n)\}$ where

$$w_i(n) = \left(|s_i(n)| + c\right)^{2-p}, i = 1, 2, \cdots, L, \tag{16}$$

$p \in [1.0, 2.0], c > 0$ for promoting different degrees of sparsity.

	$M = 1$	$M > 1, \mathbf{H} \ne \mathbf{I}$	$M > 1, \mathbf{H} = \mathbf{I}$
$p = 2$	NLMS	NSAF	APA
$2 > p > 0$	PtNLMS	PtNSAF	PtAPA

Table 1: Special cases of GPtNSAF (or sparsity-promoting NSAF).

References

[1] K.-L. Chen, H. Garudadri, and B. D. Rao, "Improved bounds on neural complexity for representing piecewise linear functions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[2] K.-L. Chen, C.-H. Lee, H. Garudadri, and B. D. Rao, "ResNEsts and DenseNEsts: Block-based DNN models with improved representation guarantees," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[3] K.-L. Chen, D. D. E. Wong, K. Tan, B. Xu, A. Kumar, and V. K. Ithapu, "Leveraging heteroscedastic uncertainty in learning complex spectral mapping for single-channel speech enhancement," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[4] K.-L. Chen, C.-H. Lee, B. D. Rao, and H. Garudadri, "A DNN based normalized time-frequency weighted criterion for robust wideband DoA estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[5] K.-L. Chen, C.-H. Lee, B. D. Rao, and H. Garudadri, "A generalized proportionate-type normalized subband adaptive filter," in *Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019.

[6] K.-L. Chen, C.-H. Lee, B. D. Rao, and H. Garudadri, "Jointly leveraging decorrelation and sparsity for improved feedback cancellation in hearing aids," in *European Signal Processing Conference (EUSIPCO)*, 2020.