Speech Enhancement Overview
Methods & Ideas

Chen Chen

2021.7.5

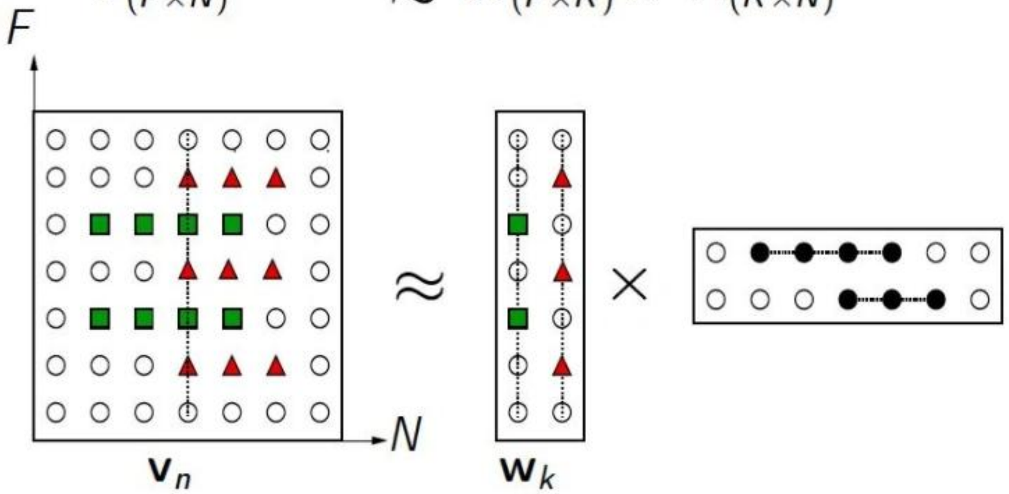
Matrix factorization： NMF & RPCA

$$M = L + S$$

L: Low Rank Matrix
S: Sparse Matrix

Robust Principal Component Analysis (RPCA)

$$\mathbf{V}_{(F\times N)} \approx \mathbf{W}_{(F\times K)} \times \mathbf{H}_{(K\times N)}$$

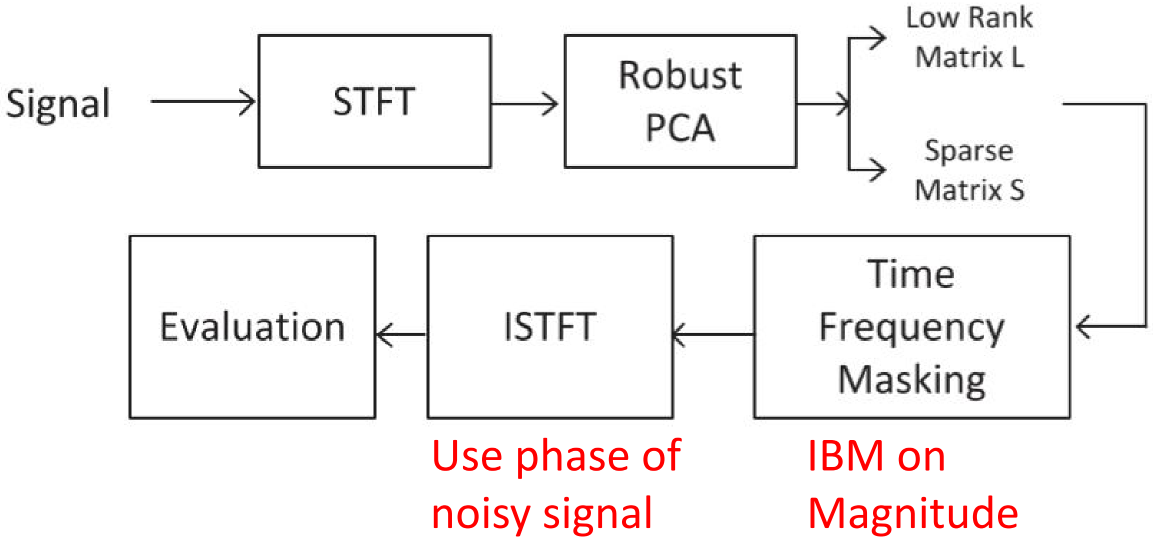


W: Basis Vectors Matrix
H : Encoding or Weights Matrix

Non-negative Matrix Factor-ization(NMF)

RPCA based Singing-Voice Separation

- Music is in a Low-Rank subspace
- Singing voices is relatively sparse
- Ignore phase



$$\text{minimize} \quad ||L||_* + \lambda ||S||_1$$

$$\text{subject to} \quad L + S = M$$

$$M \in \mathbb{R}^{n_1 \times n_2},\ L \in \mathbb{R}^{n_1 \times n_2},\ S \in \mathbb{R}^{n_1 \times n_2}$$

Nuclear norm (sum of singular values) 用于约束 Low Rank
L1-norm (sum of absolute values of matrix entries) 用于约束 sparse
λ> 0 is a trade-off parameter between the rank of L and the sparsity of S

P.-S. Huang, S. D. Chen, P. Smaragdis, and M. HasegawaJohnson, “Singing-voice separation from monaural recordings using robust principal component analysis,” in Proc. IEEE ICASSP, 2012, pp. 57–60.

NMF based Speech Music Separation

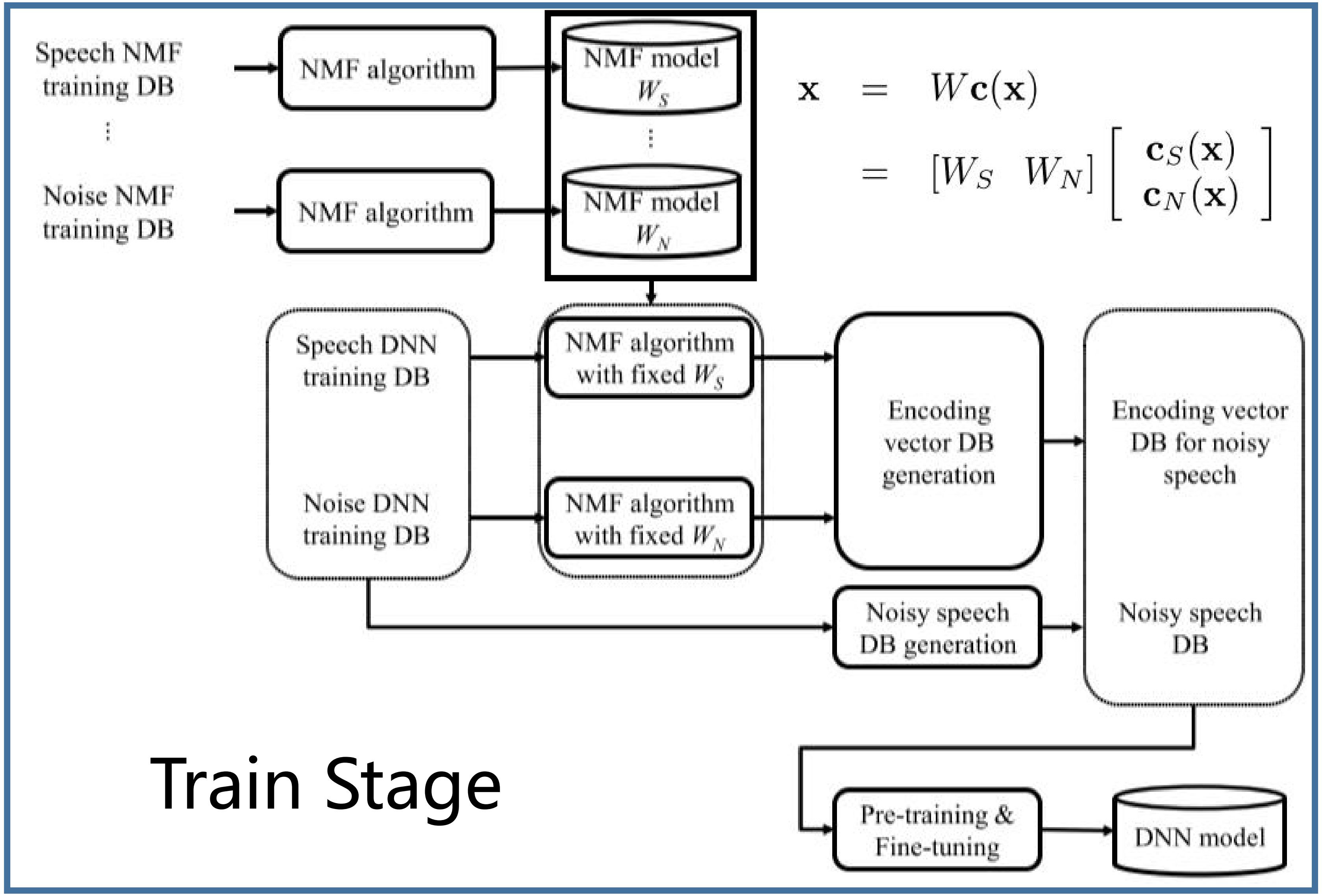
- Time domain $X(t,f) = S(t,f) + M(t,f)$
- After STFT $|X(t,f)|\, e^{j\phi_X(t,f)} = |S(t,f)|\, e^{j\phi_S(t,f)} + |M(t,f)|\, e^{j\phi_M(t,f)}$
- Ignore phase diff $\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{M}$
- Apply NMF $S = B_s W_s$, $M = B_M W_M$ $X = (B_S \; B_M)\begin{pmatrix} W_S \\ W_M \end{pmatrix}$

1. Pretrain Bs & Bm with clean speech & music data respectively
2. Apply NMF algorithm to real signal, solve Ws & Wm $\boldsymbol{X} \approx \left[\boldsymbol{B}_{\text{speech}} \;\; \boldsymbol{B}_{\text{music}}\right] \boldsymbol{W}$
3. Calculate S & M
4. Get T-F domain mask $\boldsymbol{H} = \dfrac{\tilde{\boldsymbol{S}}^p}{\tilde{\boldsymbol{S}}^p + \tilde{\boldsymbol{M}}^p}$
5. Apply the mask

$$\boldsymbol{H}_{\text{Wiener}} = \frac{\tilde{\boldsymbol{S}}^2}{\tilde{\boldsymbol{S}}^2 + \tilde{\boldsymbol{M}}^2}, \quad \boldsymbol{H}_{\text{hard}} = \text{round}\left(\frac{\tilde{\boldsymbol{S}}^2}{\tilde{\boldsymbol{S}}^2 + \tilde{\boldsymbol{M}}^2}\right)$$

E. M. Grais and H. Erdogan, “Single channel speech music separation using nonnegative matrix factorization and spectral masks," in Int. Conf. on Digital Signal Process., Corfu, 2011, pp. 1-6.

NMF-based SE Incorporating DNN



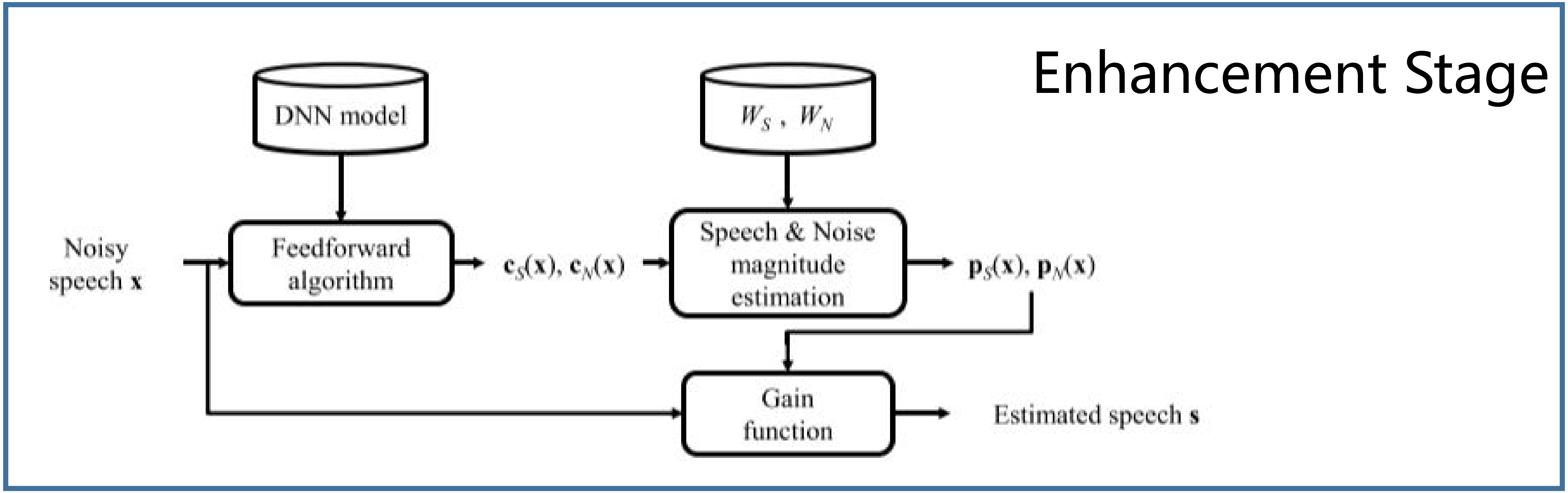
当S和N的basis之间有overlap时，无法保证通过传统NMF的损失函数最小化方法求解最优的encoding vector

难于计算的，用DNN拟合！

让DNN学习从输入信号到encoding vector的映射关系。

Kang, T. G. et al. “NMF-based speech enhancement incorporating deep neural network.” INTERSPEECH (2014).

NMF-based SE Incorporating DNN



$$\begin{aligned}
\mathbf{p}_S(\mathbf{x}) &= W_S \mathbf{c}_S(\mathbf{x}) \\
\mathbf{p}_N(\mathbf{x}) &= W_N \mathbf{c}_N(\mathbf{x}) \\
\hat{\mathbf{s}} &= \frac{(\mathbf{p}_S(\mathbf{x}))^m}{(\mathbf{p}_S(\mathbf{x}))^m + (\mathbf{p}_N(\mathbf{x}))^m} \otimes \mathbf{x}
\end{aligned}$$

Kang, T. G. et al. “NMF-based speech enhancement incorporating deep neural network.” INTERSPEECH (2014).

CLR: Convolutive sparse low-rank decomposition

It seems like RPCA+NMF

- Noise is in a Low-Rank subspace
- Speech is linear combination of basis
- Ignore Phase

$$\min_{H,L,E} \|E\|_F^2 + \lambda_H \|H\|_1 + \lambda_L \|L\|_* + \mathcal{I}_+(H)$$

$$\text{s.t.} \quad Y = \sum_{\tau=0}^{P-1} W(\tau) \overset{\tau\rightarrow}{H} + L + E.$$

- 考虑到语音信号的时间依赖性，引入卷积，且可以调节P来在灵活性和重建精度中权衡
- 相较于NMF，Noise 的秩随输入而变化，而非定值

Algorithm 1 Convolutive sparse low-rank decomposition

Input: noise+speech spectrogram Y, convolutive basis W,
Output: activations H, noise spectrogram L
Initialization: $H \leftarrow$ random positive values; $L \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots$ until convergence **do**
  update H:
    $R \leftarrow \left(Y - L^t\right)_+$
    $Z \leftarrow \sum_\tau W(\tau) \overset{\tau\rightarrow}{H^t}$
    $H_\tau \leftarrow H^t \circ \frac{\left(W(\tau)^\top \overset{\leftarrow\tau}{R} - \lambda_H \mathbf{1}^{K\times T}\right)_+}{W(\tau)^\top \overset{\leftarrow\tau}{Z}}$
    $H^{t+1} \leftarrow \frac{1}{T} \sum_\tau H_\tau$
  update L:
    $U, \Sigma, V^\top \leftarrow \text{svd}\left(Y - \sum_\tau W(\tau) \overset{\tau\rightarrow}{H^{t+1}}\right)$
    $L^{t+1} \leftarrow U \mathcal{S}_{\lambda_L}(\Sigma) V^\top$
end for

Chen, Zhuo, Brian McFee and D. Ellis. “Speech enhancement by low-rank and convolutive dictionary spectrogram decomposition.” INTERSPEECH (2014).

VAE-NMF

https://team.inria.fr/perception/research/ieee-mlsp-2018/

- Latent $\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I});$
- Speech $s_{fn} \mid \mathbf{z}_n; \boldsymbol{\theta}_s \sim \mathcal{N}_c(0, \sigma_f^2(\mathbf{z}_n)),$
- Noise $b_{fn}; \mathbf{w}_{b,f}, \mathbf{h}_{b,n} \sim \mathcal{N}_c\left(0, (\mathbf{W}_b \mathbf{H}_b)_{f,n}\right)$
- Mixed signal $x_{fn} = \sqrt{g_n} s_{fn} + b_{fn}$

be illustrated in Section 6). The speech and noise signals are further supposed to be mutually independent given the latent random vectors $\{\mathbf{z}_n\}_{n=0}^{N-1}$, such that for all $(f,n) \in \mathbb{B}$:

$$x_{fn} \mid \mathbf{z}_n; \boldsymbol{\theta}_s, \boldsymbol{\theta}_{u,fn} \sim \mathcal{N}_c\left(0, g_n \sigma_f^2(\mathbf{z}_n) + (\mathbf{W}_b \mathbf{H}_b)_{f,n}\right)$$

where $\boldsymbol{\theta}_{u,fn} = \{\mathbf{w}_{b,f}, \mathbf{h}_{b,n}, g_n\}$ is the set of unsupervised model parameters at time-frequency point (f,n).

Expectation-Maximization (EM) algorithm

From an initialization $\boldsymbol{\theta}_u^\star$ of the model parameters, it consists in iterating the two following steps until convergence:

▷ E-Step: Compute $Q(\boldsymbol{\theta}_u; \boldsymbol{\theta}_u^\star) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}_s, \boldsymbol{\theta}_u^\star)}[\ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_s, \boldsymbol{\theta}_u)];$

▷ M-Step: Update $\boldsymbol{\theta}_u^\star \leftarrow \arg\max_{\boldsymbol{\theta}_u} Q(\boldsymbol{\theta}_u; \boldsymbol{\theta}_u^\star).$

Our final goal is to estimate those coefficients according to their posterior mean:

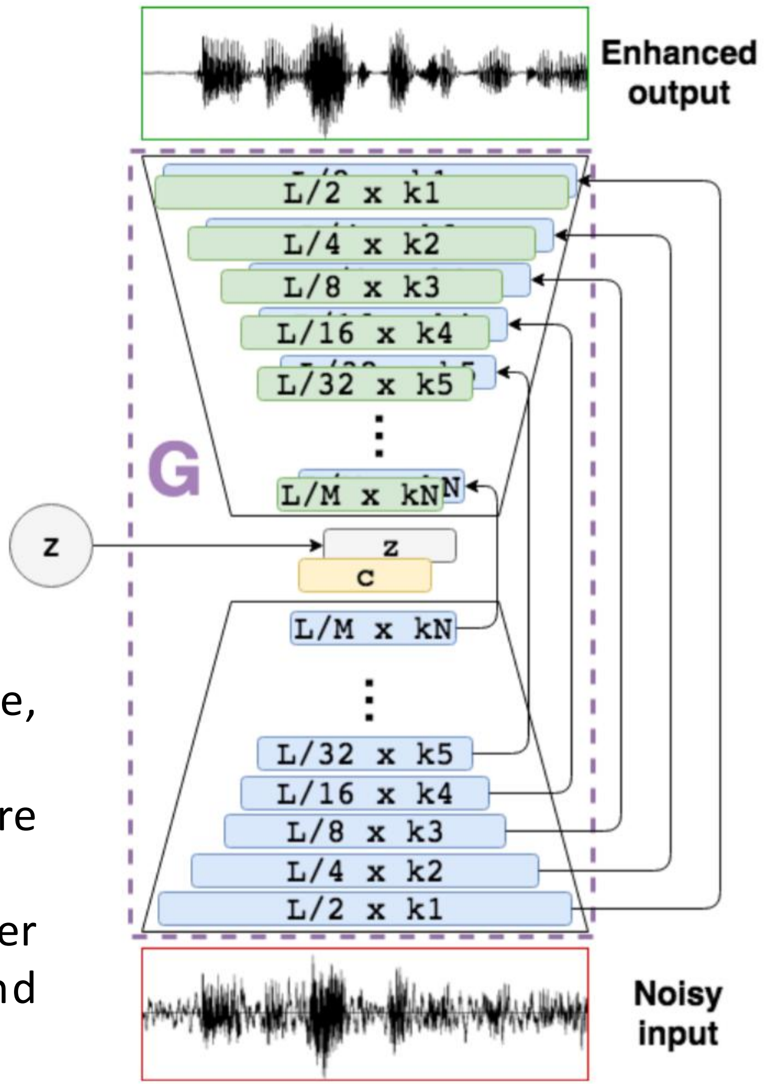
$$\begin{aligned}
\hat{\tilde{s}}_{fn} &= \mathbb{E}_{p(\tilde{s}_{fn}|x_{fn};\boldsymbol{\theta}_s,\boldsymbol{\theta}_u^\star)}[\tilde{s}_{fn}] \\
&= \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n;\boldsymbol{\theta}_s,\boldsymbol{\theta}_u^\star)}\left[\mathbb{E}_{p(\tilde{s}_{fn}|\mathbf{z}_n,\mathbf{x}_n;\boldsymbol{\theta}_s,\boldsymbol{\theta}_u^\star)}[\tilde{s}_{fn}]\right] \\
&= \mathbb{E}_{p(\mathbf{z}_n|\mathbf{x}_n;\boldsymbol{\theta}_s,\boldsymbol{\theta}_u^\star)}\left[\frac{g_n^\star \sigma_f^2(\mathbf{z}_n)}{g_n^\star \sigma_f^2(\mathbf{z}_n) + (\mathbf{W}_b^\star \mathbf{H}_b^\star)_{f,n}}\right] x_{fn}. \quad (18)
\end{aligned}$$

S. Leglaive, L. Girin, and R. Horaud, “A variance modeling framework based on variational autoencoders for speech enhancement,” in International Workshop on Machine Learning for Signal Processing (MLSP), 2018, pp. 1–6.

SEGAN

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}[(D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})-1)^2] + \lambda\,\|G(\mathbf{z},\tilde{\mathbf{x}})-\mathbf{x}\|_1.$$

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{x}_c\sim p_{\text{data}}(\mathbf{x},\mathbf{x}_c)}[(D(\mathbf{x},\mathbf{x}_c)-1)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\mathbf{x}_c\sim p_{\text{data}}(\mathbf{x}_c)}[D(G(\mathbf{z},\mathbf{x}_c),\mathbf{x}_c)^2]$$



- It provides a quick enhancement process. No causality is required and, hence, there is no recursive operation like in RNNs.
- It works end-to-end, with the raw audio. Therefore, no hand-crafted features are extracted and, with that, no explicit assumptions about the raw data are done.
- It learns from different speakers and noise types, and incorporates them together into the same shared parametrization. This makes the system simple and generalizable in those dimensions.

S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," in Proc. Interspeech, 2017.

SEGAN

http://veu.talp.cat/segan

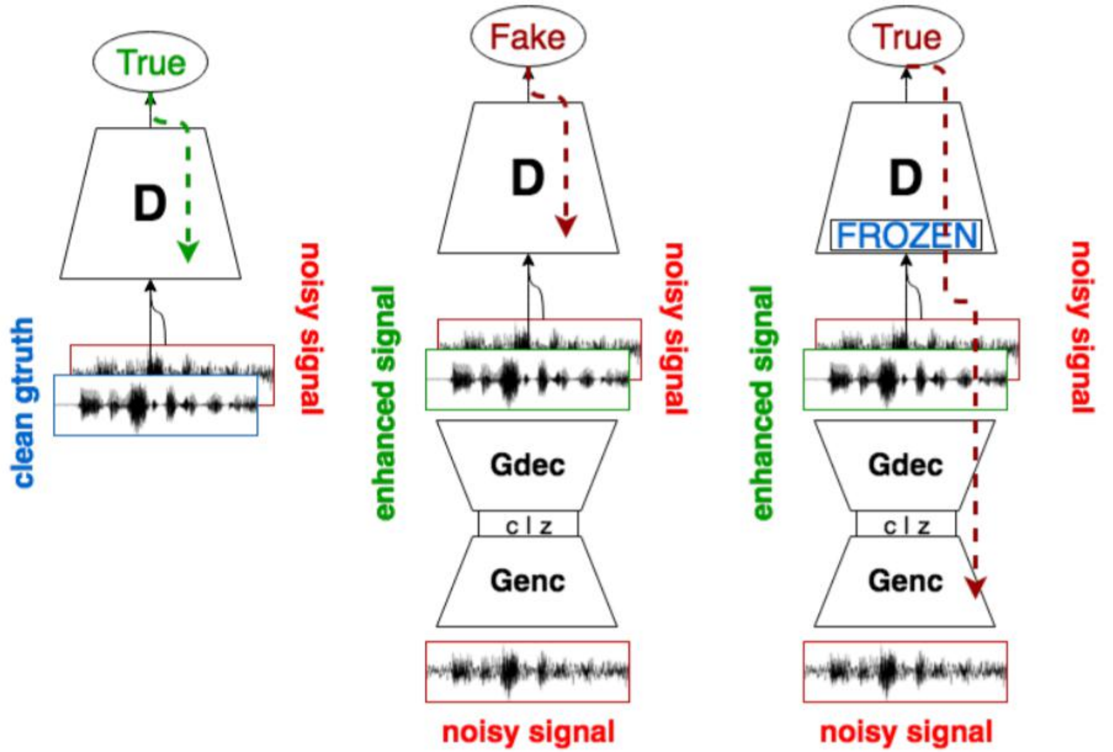


Figure 3: *Adversarial training for speech enhancement. Dashed lines represent gradient backprop.*

S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," in Proc. Interspeech, 2017.

HiFi-GAN

https://daps.cs.princeton.edu/projects/HiFi-GAN/index.php

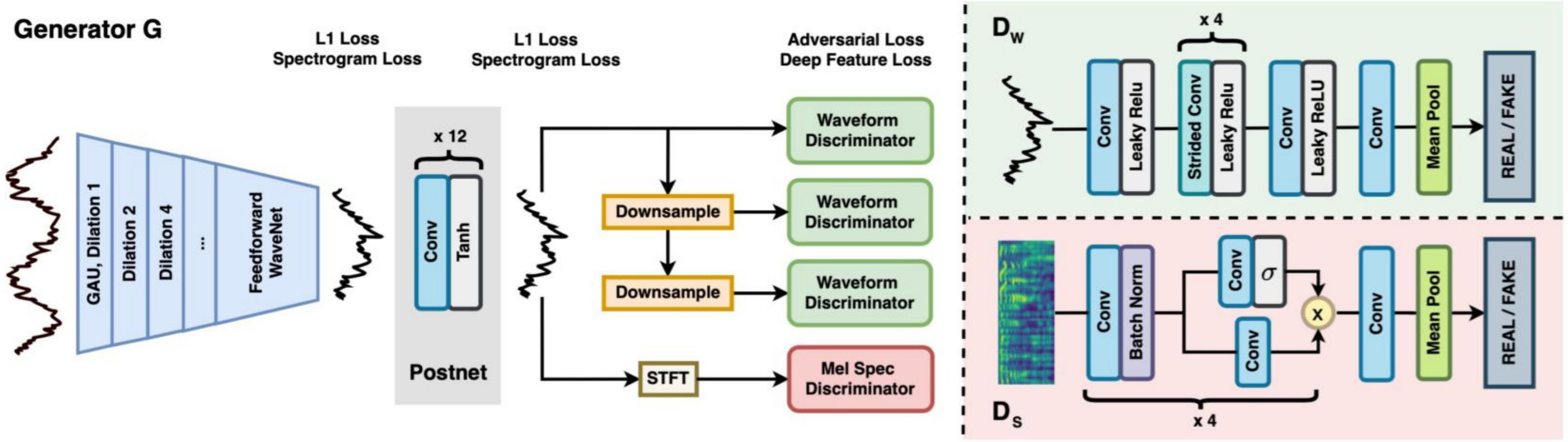


Figure 1: *GAN Architecture. Generator G includes both a feed-forward WaveNet for speech enhancement, followed by a convolutional Postnet for cleanup. Discriminators evaluate the resulting waveform (D_W, at multiple resolutions) and mel-spectrogram (D_S).*

$$L_G^{\text{Adv}}(x, x'; D_k) = \max[1 - D_k(G(x)), 0] \qquad (1)$$

$$L_{D_k}(x, x') = \max[1 + D_k(G(x)), 0] + \max[1 - D_k(x'), 0] \qquad (2)$$

where (x, x') is the pair of input audio x and target audio x'.

For a specific discriminator D_k, we formulate its deep feature matching loss on the generator as follows:

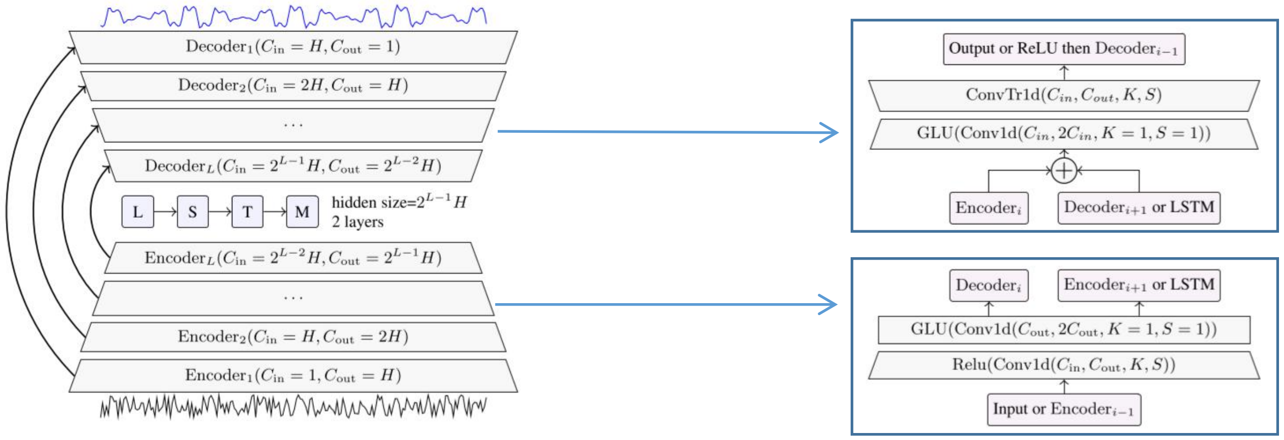
$$L_G^{\text{FM}}(x, x'; D_k) = \sum_{i=1}^{T_k} \frac{1}{N_i} ||D_k^{(i)}(G(x)) - D_k^{(i)}(x')||_1 \qquad (3)$$

where T_k is the numebr of layers in D_k excluding the output layer, and N_i is the number of units in the i-th layer $D_k^{(i)}$.

J. Su, Z. Jin, and A. Finkelstein, "HiFi-GAN: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks," in Proc. Interspeech, 2020.

DEMUCS

https://facebookresearch.github.io/denoiser/



runs faster than real-time on a single laptop CPU core

(a) *Causal* DEMUCS *with the noisy speech as input on the bottom and the clean speech as output on the top. Arrows represents U-Net skip connections. H controls the number of channels in the model and L its depth.*

(b) *View of each encoder (bottom) and decoder layer (top). Arrows are connections to other parts of the model. C_{in} (resp. C_{out}) is the number of input channels (resp. output), K the kernel size and S the stride.*

Overall we wish to minimize the following,

$$\frac{1}{T}[\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_1 + \sum_{i=1}^{M} \boxed{L_{\text{stft}}^{(i)}(\boldsymbol{y}, \hat{\boldsymbol{y}})]}$$

where M is the number of STFT losses, and each $L_{\text{stft}}^{(i)}$ applies the STFT loss at different resolution with number of FFT bins $\in \{512, 1024, 2048\}$, hop sizes $\in \{50, 120, 240\}$, and lastly window lengths $\in \{240, 600, 1200\}$.

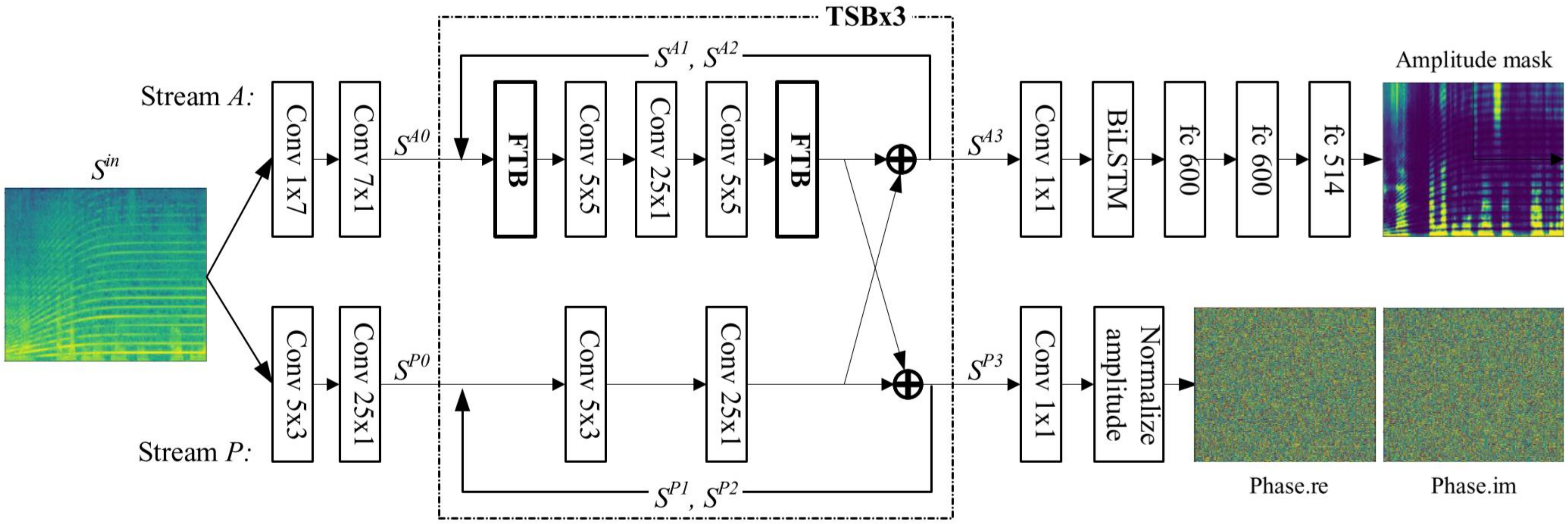
$$L_{\text{stft}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{sc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + L_{mag}(\boldsymbol{y}, \hat{\boldsymbol{y}})$$

$$L_{sc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\| |STFT(\boldsymbol{y})| - |STFT(\hat{\boldsymbol{y}})| \|_F}{\| |STFT(\boldsymbol{y})| \|_F}$$

$$L_{mag}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{T} \| \log |STFT(\boldsymbol{y})| - \log |STFT(\hat{\boldsymbol{y}})| \|_1$$

A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in Proc. Interspeech, 2020.

PHASEN



The basic idea behind PHASEN is to separate the predictions of amplitude and phase, as the two prediction tasks may need different features. With the information from the amplitude stream, the features for phase estimation is significantly improved.

D. Yin, C. Luo, Z. Xiong, and W. Zeng, "Phasen: A phase-and-harmonics-aware speech enhancement network," in Proc. AAAI 2020, pp. 9458-9465.