Speech Enhancement

Tal Rosenwein

Speech Enhancement- Background

Imagine recording a target speaker's voice in a living room using a microphone placed on a table.

- This scenario represents a typical use case of a voice-controlled smart device or a video-conferencing device in a remote-work situation.

Many sounds may co-occur while the speaker is speaking, e.g., a vacuum cleaner, music, children screaming, voices from another conversation, or from a TV.

The speech signal captured at a microphone thus consists of a mixture of the target speaker's speech and interference from the speech of other speakers and background noise (of course there are reverberations that will be excluded from this discussion for simplicity).

Speech Enhancement- Background

Imagine recordin a target speaker's voice in a living room using a microphone placed on a table

- This scenari smart device or a video-c

Many sounds ma uum cleaner, music, children s a TV.

The speech signa re of the target speaker's speech s and background nois cluded from this discussion fo

Speech Enhancement- Background

$$\mathbf{y}^m = \mathbf{x}_s^m + \sum_{k \neq s} \mathbf{x}_k^m + \mathbf{v}^m,$$

Where:

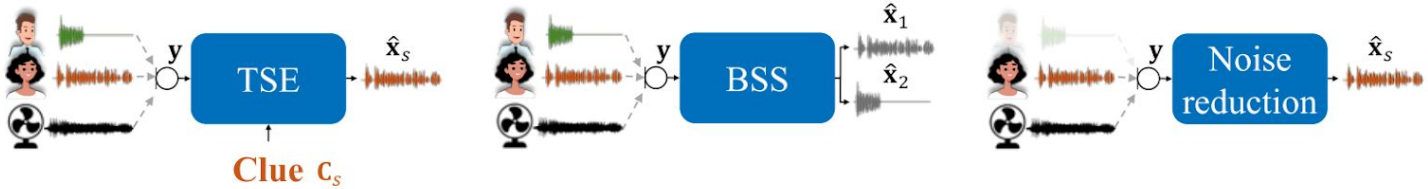
1. $y^m \in R^T$ is the time-domain signal of the mixture (acquired audio)
2. $x_s^m \in R^T$ is the target speech
3. $x_k^m \in R^T$ is the interference speech (non-target/s speech)
4. $v_m \in R^T$ is the noise signal
5. Variable *T* represents the duration (number of samples) of the signals, *m* is the index of the microphone in an array of microphones, *s* represents the index of the target speaker and *k* is the index for the other speech sources.

Speech Enhancement- Challenges

Why is speech enhancement so challenging?

- Signal to noise ratio
- Multiple interferences
- Interference can overlap (in time and in spectral properties) with the desired signal - energetic vs informative masking.
- Acoustic environment (reverb / echo)
- Near field / far field interference / desired signal.
- Interference is non stationary and thus hard to predict.
- Transient noises
- etc.

Speech Enhancement- Tasks



(a) TSE problem (b) BSS problem (c) Noise reduction problem

Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- Blind source separation.
- Target speech enhancement.

Image Source

Speech Enhancement- Tasks

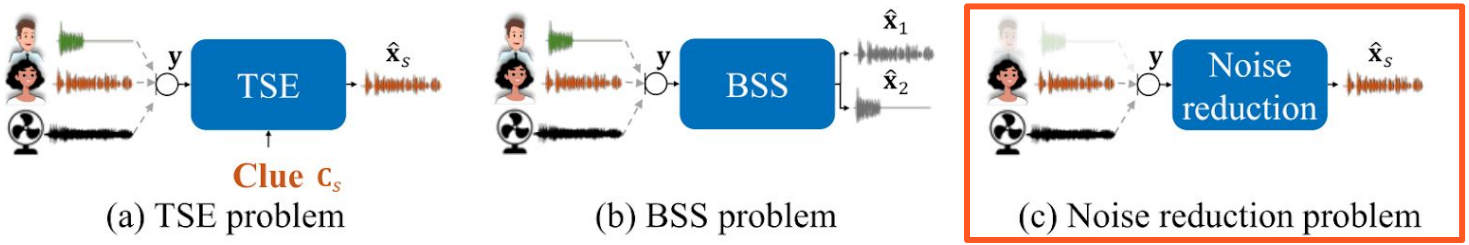


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- **Noise reduction.**
- Blind source separation.
- Target speech enhancement.

Image Source

Speech Enhancement- Tasks

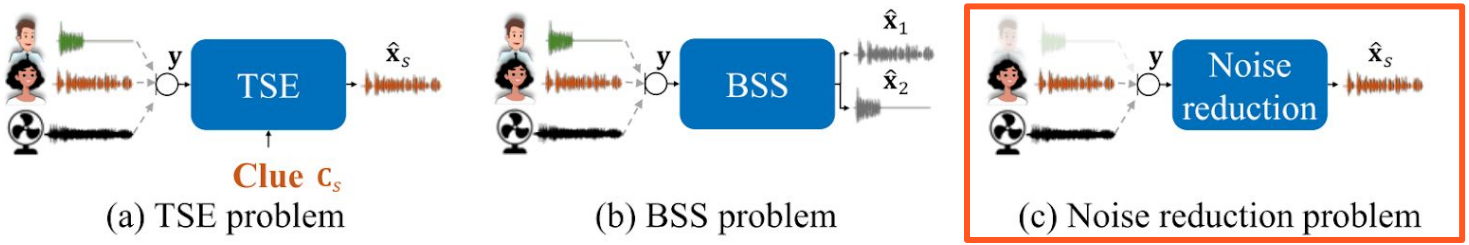


Fig. 2. Comparison of TSE with BSS and noise reduction

Noise reduction assumes that the interference consists of background noise only, i.e., $i = v$, and can thus enhance the target speech without requiring clues:

$$\hat{x}_s = Denoise(y; \theta_{Denoise})$$

$$\mathbf{y}^m = \mathbf{x}_s^m + \sum_{k \neq s} \mathbf{x}_k^m + \mathbf{v}^m,$$

Image Source

Speech Enhancement- Tasks

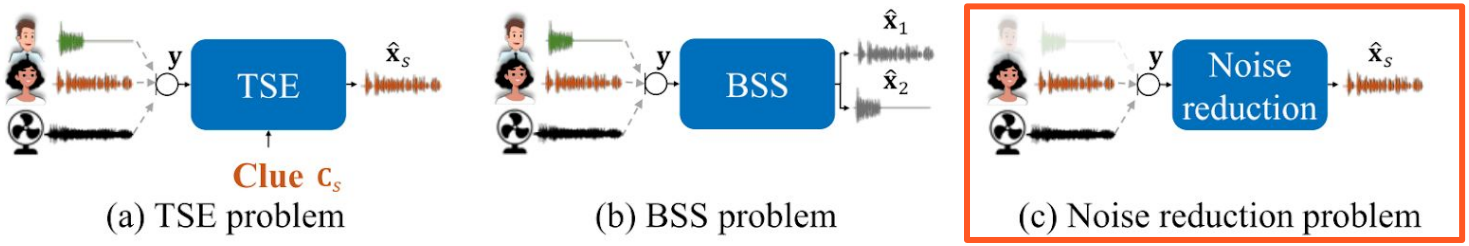


Fig. 2. Comparison of TSE with BSS and noise reduction

- noise reduction cannot suppress interfering speakers because it cannot discriminate among different speakers in a mixture without clues.
- Noise reduction is often used, e.g., in video-conferencing systems or hearing aids.

Image Source

Speech Enhancement- Tasks

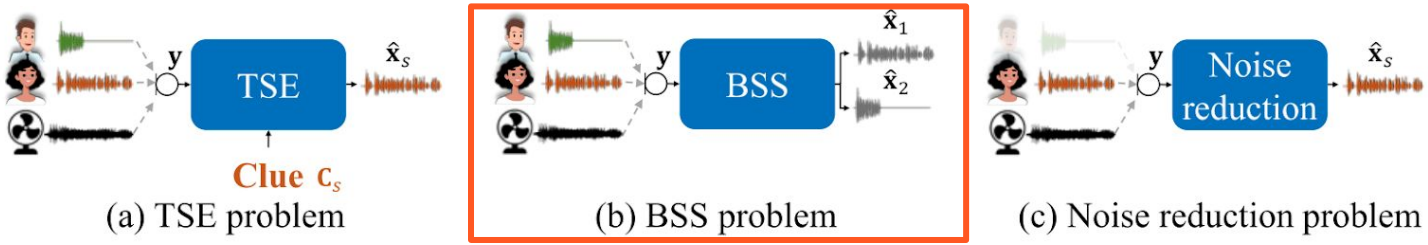


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- **Blind source separation.**
- Target speech enhancement.

Image Source

Speech Enhancement- Tasks

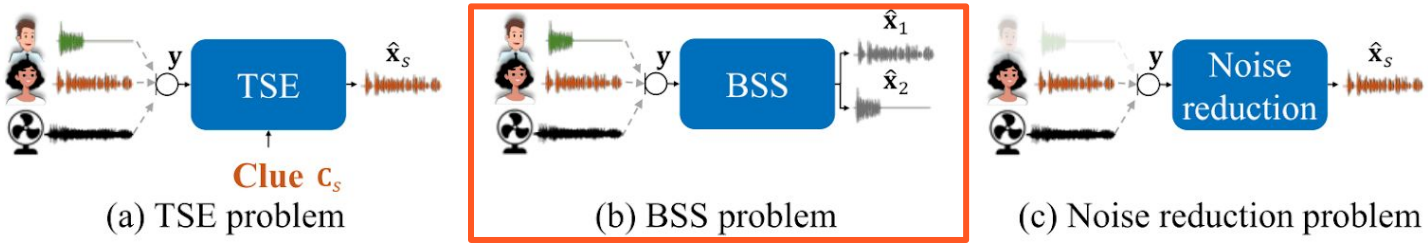


Fig. 2. Comparison of TSE with BSS and noise reduction

Blind Source Separation (BSS) estimates all the source signals in a mixture without requiring clues:

$$\{\hat{x}_1, \ldots, \hat{x}_K\} = BSS(y; \theta_{BSS}),$$

where $BSS(\cdot; \theta_{BSS})$ represents a separation system with parameters θ_{BSS}, $\hat{x}_k$ are the estimates of the speech sources, and K is the number of sources in the mixture.

Image Source

Speech Enhancement- Tasks

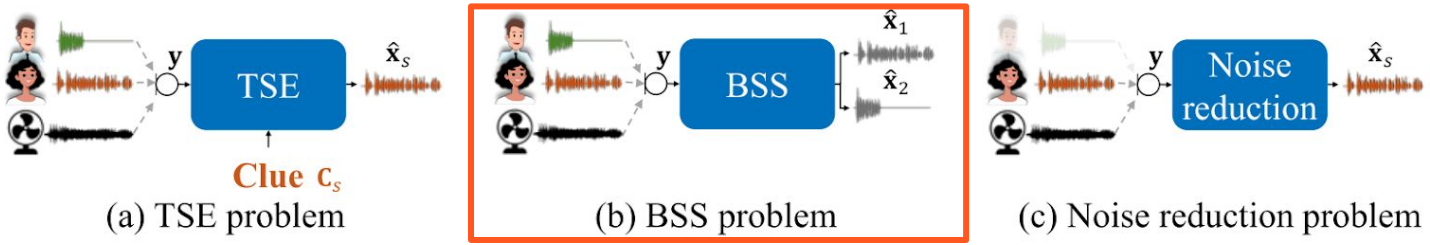


Fig. 2. Comparison of TSE with BSS and noise reduction

- BSS does not and cannot differentiate the target speech from other speech sources.
- There is a global permutation ambiguity problem between the outputs and the speakers.
- Since the number of outputs is given by the number of sources, the number of sources K must be known or estimated.
- Typical use cases for BSS include applications that require estimating speech signals of every speaker, such as automatic meeting transcription systems.

Image Source

Speech Enhancement- Tasks

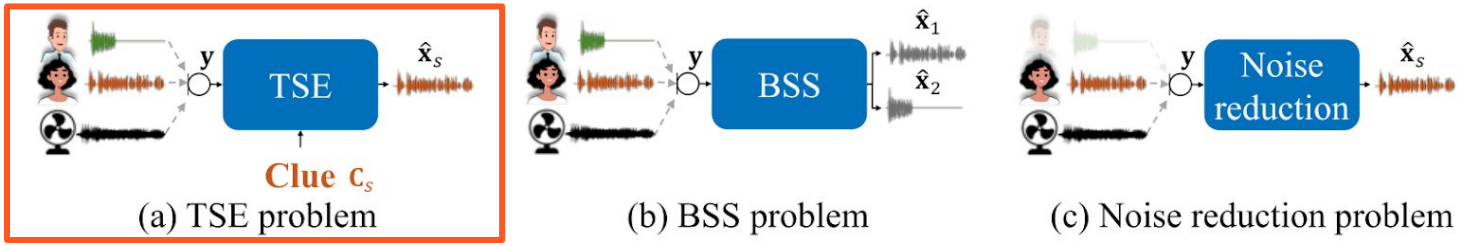


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- Blind source separation.
- **Target speech enhancement.**

Image Source

Speech Enhancement- Tasks

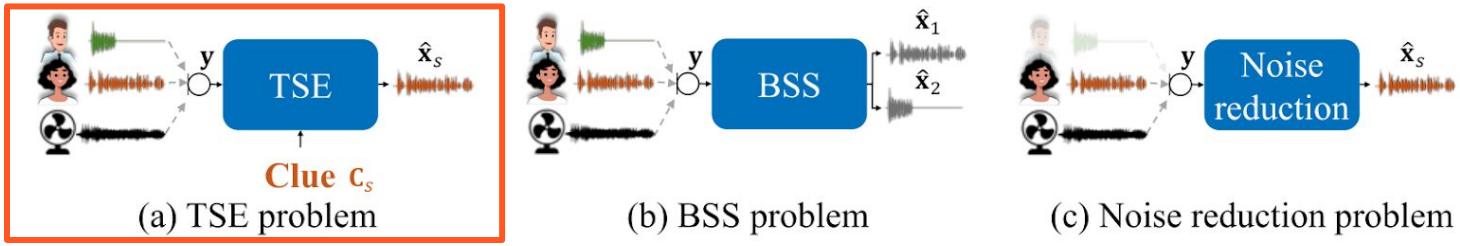


Fig. 2. Comparison of TSE with BSS and noise reduction

Target Speech Enhancement (TSE) estimates the target speech, given a clue, C_s, as

$$\hat{x}_s = TSE(y, C_s; \theta_{TSE}),$$

where $\hat{x}_s$ is the estimate of the target speech, $TSE(\cdot; \theta_{TSE})$ represents a TSE system with parameters θ_{TSE}. The clue, C_s, allows identifying the target speaker in the mixture. It can be of various types, such as a pre-recorded enrollment utterance, $C_s^{(a)}$, a video signal capturing the face or lips movements of the target speaker, $C_s^{(v)}$, or such spatial information as the direction of arrival (DOA) of the speech of the target speaker, $C_s^{(d)}$. TSE **estimates only the target speech** signal.

Image Source

Speech Enhancement- Tasks

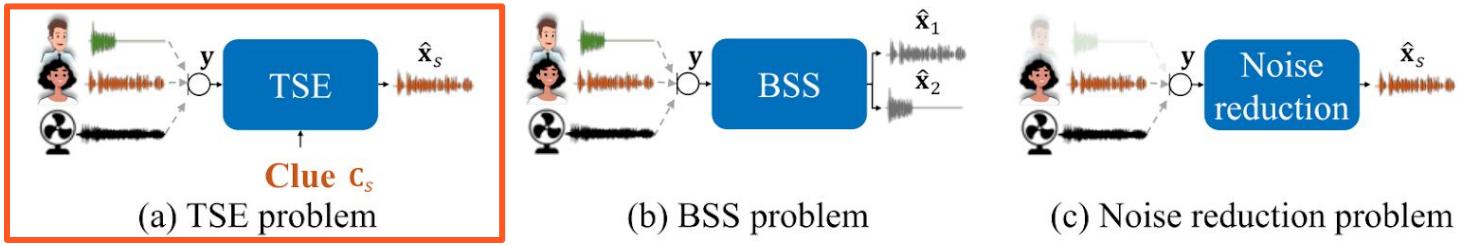


Fig. 2. Comparison of TSE with BSS and noise reduction

- TSE is typically used when one wants to isolate a specific voice from the mixture.

Image Source

Speech Enhancement- Categorization

Speech enhancement can be categorized as follows:

- Latency
  - Online
  - Offline
  - #ms latency?
- # of microphones:
  - Single mic
  - Multiple mics (mic-array)
- Filtering property:
  - Spatial filtering
  - Spectral filtering

Speech Enhancement- Categorization

Speech enhancement can be categorized as follows:

- Filtering domain:
  - Domain:
    - Spectral
    - Time
  - Feature extraction (Encoder):
    - STFT
    - Learnable features from audio domain
- Filtering method:
  - Supervised / unsupervised
  - Masked based / regression based / generative
  - Causal / non-causal
  - Realtime / offline
  - Learnable / DSP based methods

Speech Enhancement- Categorization

Speech enhancement can be categorized as follows:

- Sources:
  - Known / unknown - use or not enrollment-
  - Number of sources are known / unknown
  - Near / far field
  - Background noise consists of interference of speakers or not?
  - Read vs spontaneous speech
- Context length

Speech Enhancement- Categorization

Speech enhancement can be categorized as follows:

- Deployment constraints (HW):
  - Inactive target speaker
  - Mismatch between training and evaluation criteria
  - Unknown behavior
  - Robustness to recording conditions:
    - Training data should match the application scenario relatively well.
    - Real recordings can be used to adapt a TSE system to a new environment.
    - Speaker close / Speaker open
  - Lightweight and low-latency systems.
  - Spatial (stereo) rendering

Source

Speech Enhancement- Datasets

Two types of datasets:

- Static mixing- pre determined mixing across all training epochs
- Dynamic mixing - on-the-fly mixing

Speech Enhancement- Datasets

Two types of datasets:

- **Static mixing**- pre determined mixing across all training epochs
  - WSJ0-2mix and WSJ0-3mix datasets- Link
  - LibriMix- Link

Table 2: *Statistics of derived speech separation datasets.*

Dataset	Split	# Utterances	Hours
wsj0-{2,3}mix	train	20,000	30
	dev	5,000	8
	test	3,000	5
Libri2Mix	train-360	50,800	212
	train-100	13,900	58
	dev	3,000	11
	test	3,000	11
Libri3Mix	train-360	33,900	146
	train-100	9,300	40
	dev	3,000	11
	test	3,000	11
SparseLibri2Mix	test	3,000	6
SparseLibri3Mix	test	3,000	6
VCTK-2mix	test	3,000	9

Source

Speech Enhancement- Datasets

Two types of datasets:

- Static mixing- pre determined mixing across all training epochs
  - WSJ0-2mix and WSJ0-3mix datasets- Link
  - LibriMix- Link
- **Dynamic mixing** - on-the-fly mixing
  - VoxCeleb 1 & 2 - Link
  - AV Speech - Link

Dataset	VoxCeleb1	VoxCeleb2
# of POIs	1,251	6,112
# of male POIs	690	3,761
# of videos	22,496	150,480
# of hours	352	2,442
# of utterances	153,516	1,128,246
Avg # of videos per POI	18	25
Avg # of utterances per POI	116	185
Avg length of utterances (s)	8.2	7.8

Table 1: *Dataset statistics for both* `VoxCeleb1` *and* `VoxCeleb2`. *Note* `VoxCeleb2` *is more than 5 times larger than* `VoxCeleb1`. *POI: Person of Interest.*

Source

Speech Enhancement- Datasets

Two types of datasets:

- Static mixing- pre determined mixing across all training epochs
  - WSJ0-2mix and WSJ0-3mix datasets- Link
  - LibriMix- Link
- Dynamic mixing - on-the-fly mixing
  - VoxCeleb 1 & 2 - Link
  - AV Speech - Link
- **Background Noise Datasets:**
  - For noise, researchers often use either samples from AudioSet and its variants (i.e., DNS challenge), and VGG sound dataset

Speech Enhancement- Evaluation Metrics

Two types of metrics:

- With reference (ground truth)
- Without reference

Given the predicted signal, we can/cannot compare it to the reference and evaluate the performances.

Speech Enhancement- Evaluation Metrics

The metrics should evaluate **perceptual performances**:

- Intelligibility
- Quality
- Noise removal

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference	**WER**	MOS		

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR		
W/O Reference				

Signal to Distortion Ratio (SDR / SNR)

Let us consider a mixture $x = s + n \in R^L$ of a target signal *s* and an interference signal *n*. Let $\hat{s}$ denote an estimate of the target obtained by some algorithm. The classical SDR (or SNR) considers $\hat{s}$ as the estimate and *s* as the target:

$$SDR = 10\, log_{10}(\frac{||s||^2}{||s-\hat{s}||^2})\ [dB]$$

what is considered as the noise in such a context is the residual (distortion/noise) $s - \hat{s}$.
This term is combined of 3 components:

- Noise- should filter but didn't filter
- Signal - should not filter but filtered.
- Other- artifacts that were added that weren't either in *s* or *n*

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR, **SI-SDR**		
W/O Reference				

Scale Invariant Signal to Noise/Distortion Ratio (SI-SNR / SI-SDR)

SI-SNR evaluation metric is defined as:

$$\mathbf{s}_{target} = \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{\|\mathbf{s}\|^2}$$

$$\mathbf{e}_{noise} = \hat{\mathbf{s}} - \mathbf{s}_{target}$$

$$\text{SI-SNR} := 10\, log_{10} \frac{\|\mathbf{s}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2}$$

where $\hat{s} \in R^{1xT}$ and $s \in R^{1xT}$ are the estimated and target clean sources respectively, T denotes the length of the signals, and $\hat{s}$ and s are both normalized to have zero-mean to ensure scale-invariance.

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR, **SI-SDR**		
W/O Reference				

- Higher is better- when perfect, the SI-SNR is equal to infinity as the distortion is 0.
- It can be used as a loss function if we minimize the negative SI-SNR
- Scale invariance

$$l(\hat{s}_i, s_i) == l(\alpha\hat{s}_i, s_i)$$

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR, **SI-SDR**		
W/O Reference				

- Higher is better- when perfect, the SI-SNR is equal to infinity as the distortion is 0.
- It can be used as a loss function if we minimize the negative SI-SNR
- Scale invariance- find a scaled version of the GT, and then calculate the signal to distortion ratio (SDR) accordingly

$$l(\hat{s}_i, s_i) == l(\alpha \hat{s}_i, s_i)$$

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR, SI-SDR, **SI-SNRi**		
W/O Reference				

Scale Invariant Signal to Noise/Distortion Ratio Improvement (SI-SNRi)

Measures the improvement (compared to input).

Defined as:

$$SI - SNRi = SISNR(target) - SISNR(input)\ [dB]$$

Where we assign x instead of s in the SI-SNR formula in SI-SNR(input)

Since it is in log scale (dB) the improvement is relative to the input

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference		SDR, SI-SDR, SI-SNRi	**PESQ**	
W/O Reference				

Perceptual Evaluation of Speech Quality (PESQ) is a family of standards comprising a test methodology for automated assessment of the speech quality as experienced by a user of a telephony system

- PESQ compares the (original) reference signal with the predicted one.
- It assesses the voice quality perceived by human beings.
- The score is in the range of [-0.5, 4.5] where higher is better.

Source

Link for examples

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
V				

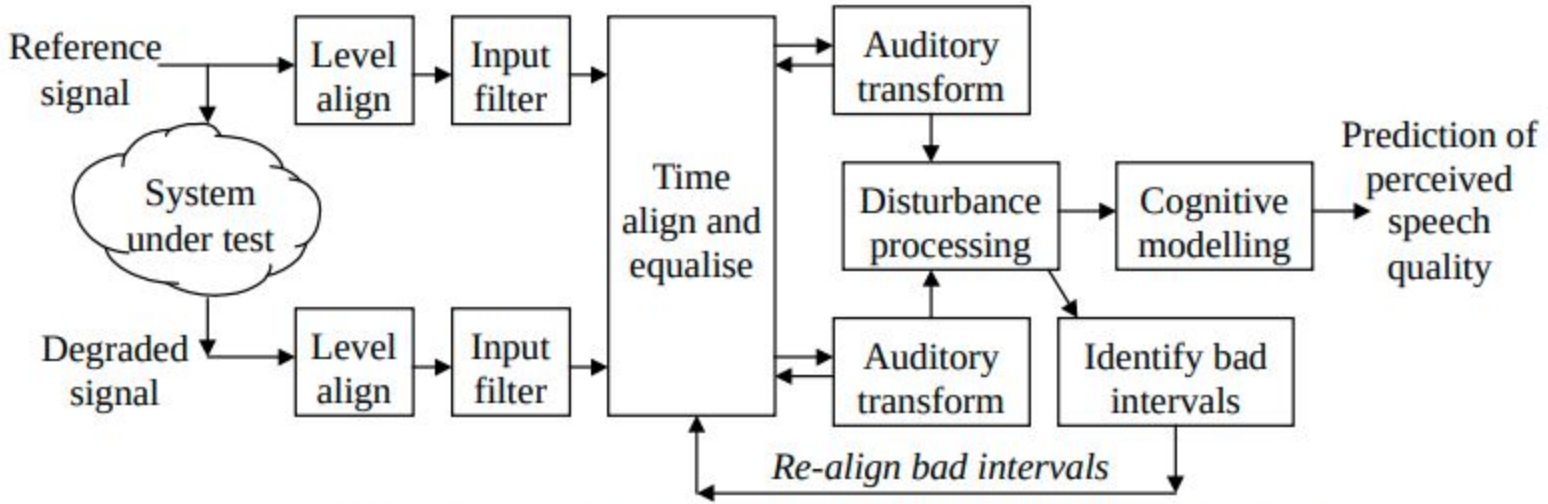


Figure 1: Structure of perceptual evaluation of speech quality (PESQ) model.

- The score is in the range of [-0.5, 4.5] where higher is better.

Source

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	**STOI**	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference				

Short-Time Objective Intelligibility measure (STOI) is correlated with the intelligibility of the signal.

- In the range of [0, 1], where higher is better.
- Shows high correlation with the intelligibility of noisy and time-frequency weighted noisy speech
- Based on shorter time segments (386 ms).

Source

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference				

But what happens when there is no access to the reference signals?

Source

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference		**MOS**		

Mean opinion score (MOS) is a commonly used but **subjective method** for evaluating speech **quality**.
It has a straightforward approach:

- Gather a diverse group of people (test subjects). Ask them to:
  - Listen to recordings
  - Rate the quality on a scale of 1 to 5, where higher is better.
- Average their scores.

Source

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference		**MOS**		

Mean opinion score (MOS) is a commonly used but **subjective method** for evaluating speech **quality**.

- ITU offers some recommendations for designing, running, and reporting the experiments to minimize subjectivity.
- It is still a subjective assessment of sound quality (or impairment).
- It's critical to understand the experiment conditions before the results, as they're easy to manipulate.

Source

Speech Enhancement- Evaluation Metrics

	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference	**WER**	MOS		

Using (manual / automatic) ASR engines to measure intelligibility.

- It is confusing as it requires to have a **reference transcript** (but not the signal itself)

Source

Speech Enhancement- Evaluation Metrics

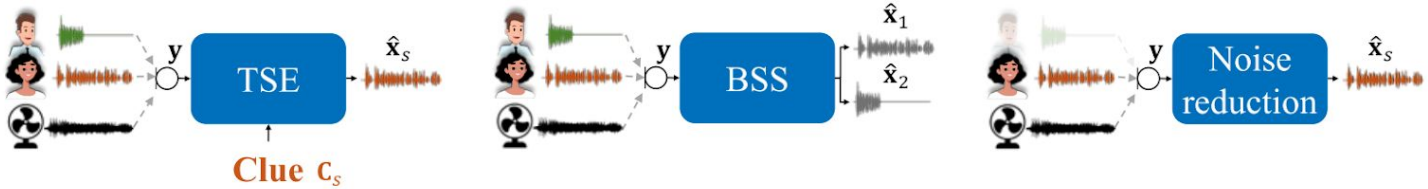
	Intelligibility	Quality - Distortion	Quality- Perception	Noise Removal
W Reference	STOI	SDR, SI-SDR, SI-SNRi	PESQ	
W/O Reference	**WER**	MOS		

Need to measure the **noise removal**

- Sometimes it is considered implicitly in other measures, but not always measured explicitly (directly)

Source

Speech Enhancement- Tasks - Deep Dive



(a) TSE problem (b) BSS problem (c) Noise reduction problem

Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- Blind source separation.
- Target speech enhancement.

Image Source

Speech Enhancement- Noise Reduction

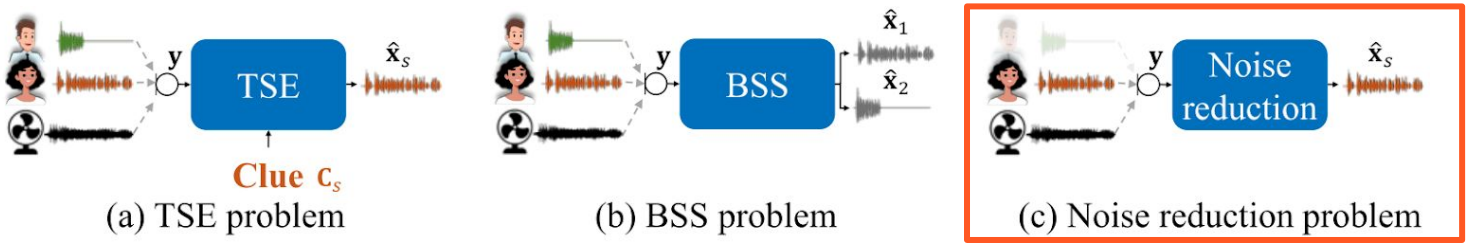


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- **Noise reduction.**
- Blind source separation.
- Target speech enhancement.

Image Source

Speech Enhancement- Noise Reduction

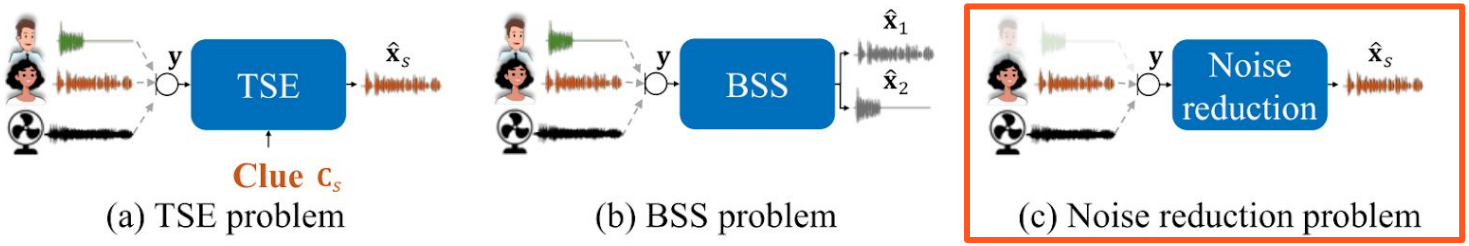


Fig. 2. Comparison of TSE with BSS and noise reduction

- A desired random variable *S* is corrupted by additive noise *V* producing the (received) measured signal *X*

$$X = S + V$$

- We would like to estimate S out of the mixture.

Image Source

Speech Enhancement- Noise Reduction

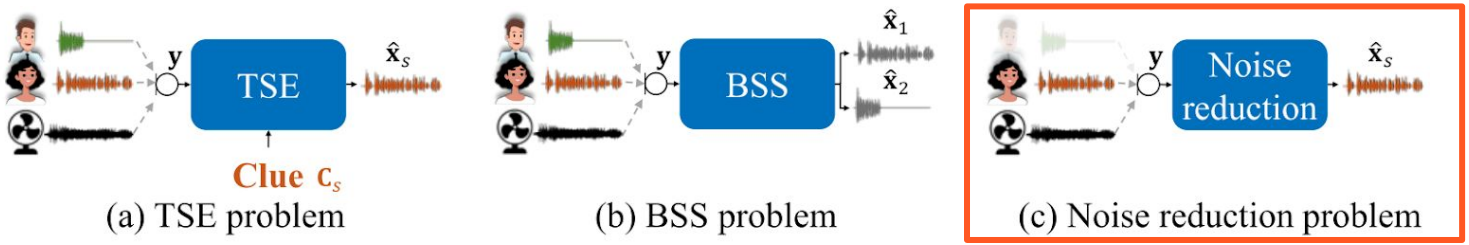


Fig. 2. Comparison of TSE with BSS and noise reduction

- Spectral subtraction
- Wiener filter
- Beamforming
- NN-based methods

Image Source

Speech Enhancement- Noise Reduction

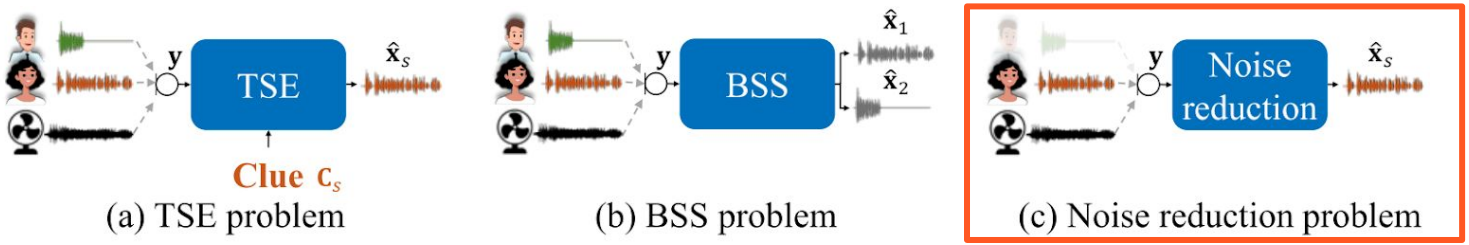


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Spectral subtraction**
- Wiener filter
- Beamforming
- NN-based methods

Image Source

Speech Enhancement- Noise Reduction

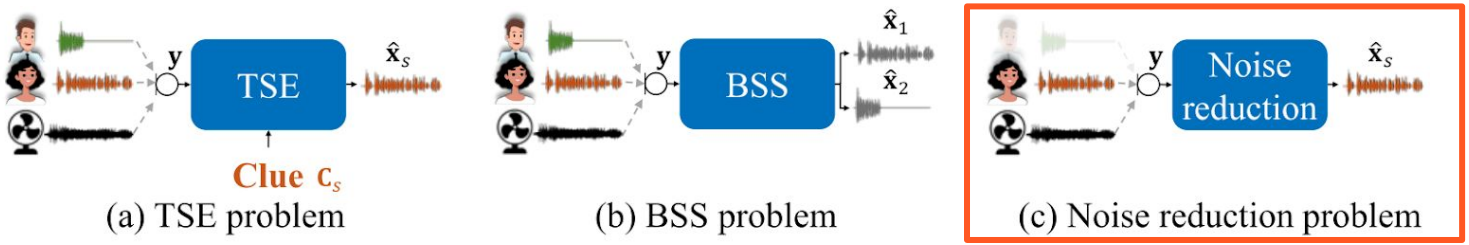


Fig. 2. Comparison of TSE with BSS and noise reduction

Spectral subtraction

- We would like to remove background noise from the acquired signal

$$y[n] = x[n] + s[n]$$

- Where: **y[n]** is the acquired signal, **x[n]** is the desired signal, and **s[n]** is stationary background noise signal

Image Source

Speech Enhancement- Noise Reduction

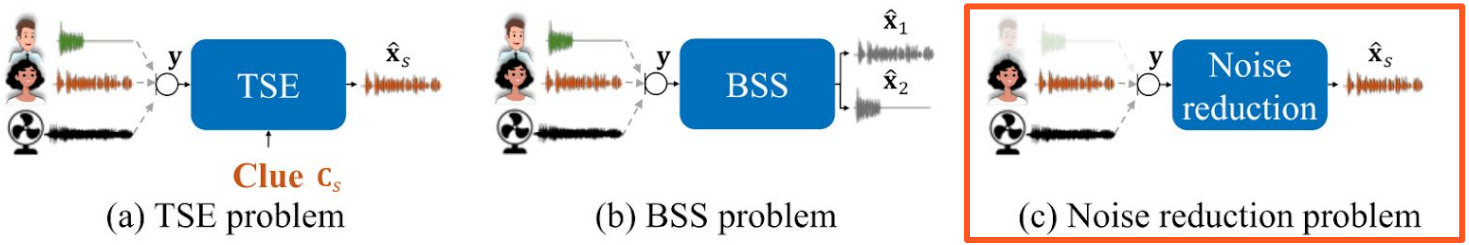


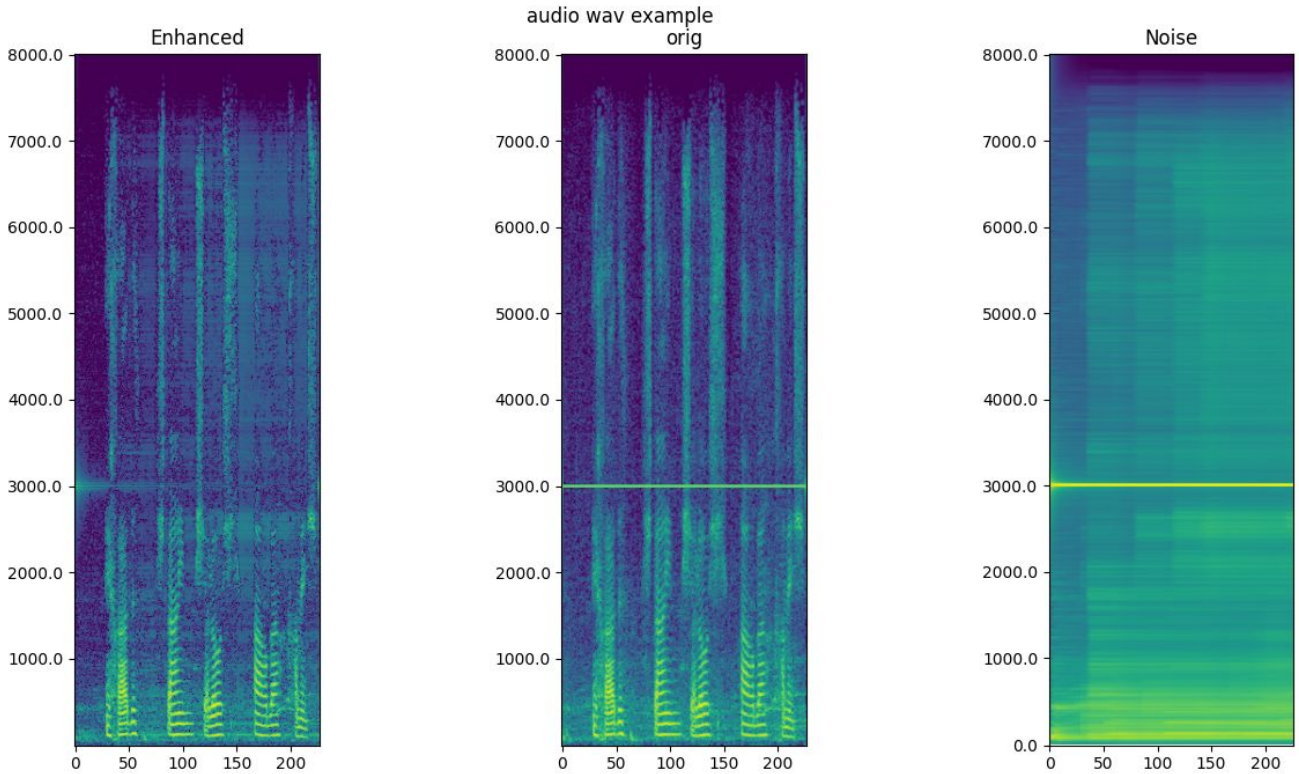
Fig. 2. Comparison of TSE with BSS and noise reduction

Spectral subtraction

- Approach: Subtract the estimated noisy magnitude from the acquired signal.
- Therefore, we need to estimate the noise:
  - Locate segments / frames where there is no speech (using VAD)
  - Aggregate statistics to a buffer: (determine buffer size)
  - Average the buffer to form the noise estimation (assuming non stationary signals will be averaged out)
  - Subtracting the noise estimation from the input signal y[n]

Image Source

Speech Enhancement- Noise Reduction



Speech Enhancement- Noise Reduction

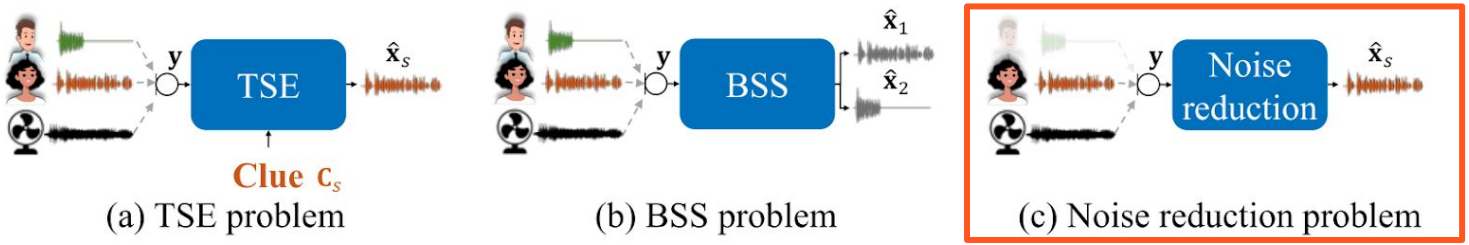


Fig. 2. Comparison of TSE with BSS and noise reduction

Spectral subtraction

Important notes:

- Filters in the spectral domain on the **magnitude only**. Reconstructing with the noisy phase may result in **musical noise**.
- It relies on VAD which has its own errors
- It removes only stationary noise, as transient noises are averaged.

Image Source

Speech Enhancement- Noise Reduction

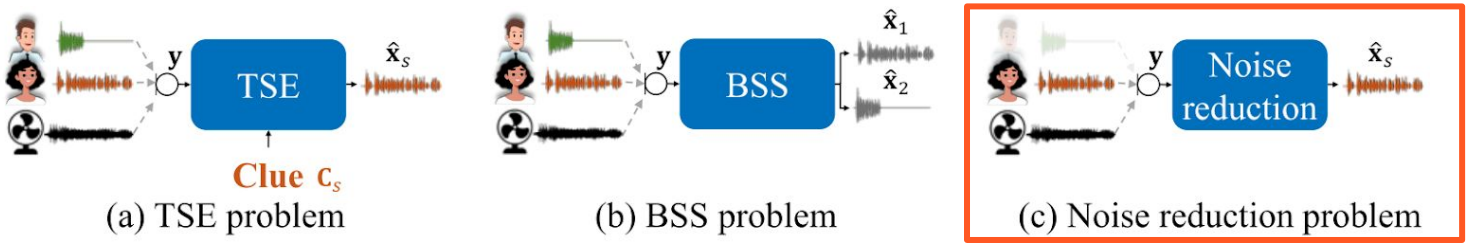


Fig. 2. Comparison of TSE with BSS and noise reduction

- Spectral subtraction
- **Wiener filter - Credit David Levine**
- Beamforming
- NN-based methods

Image Source

Speech Enhancement- Noise Reduction

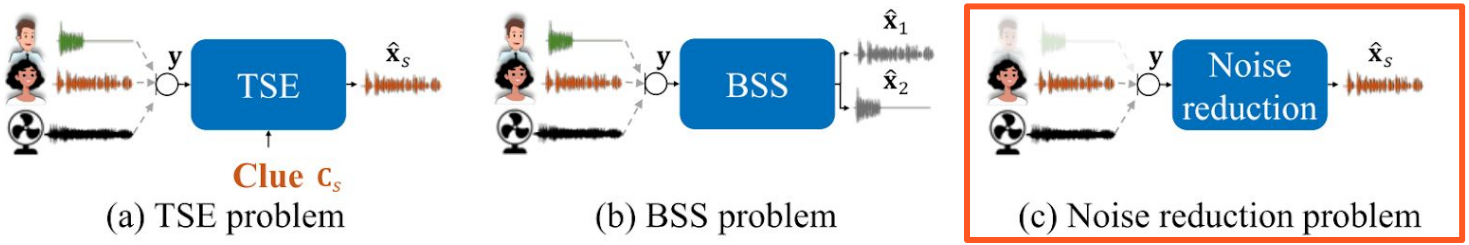


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter $X = S + V$

Wiener filter applies a weight on the received signal X, to get an estimated quantity of S, termed $\hat{S}$:

$$\hat{S} = wX = w(S + V)$$

The estimation error (or deviation) *D* has the form,

$$D = \hat{S} - S = (w - 1)S + wV$$

Image Source

Speech Enhancement- Noise Reduction

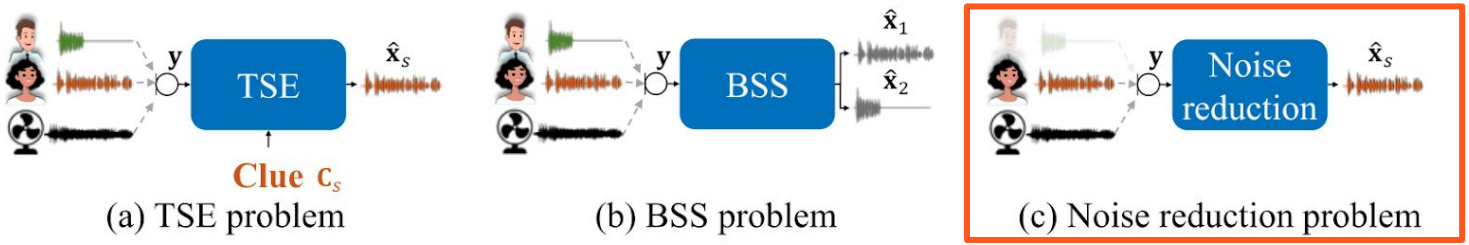


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter

$$D = \hat{S} - S = (w - 1)S + wV$$

We wish to **find the weights W** that minimize the error in the mean-square sense

$$w_{opt} = argmin_w E[D^2]$$

The following statistical assumptions are made:

1. *S* is zero-mean; i.e, *E{S}* = 0
2. V is zero-mean; i.e, E{V } = 0
3. S and V are uncorrelated; thus E{SV } = 0 .

Furthermore, the variances of S and V are known:

1. $E[S^2] = \sigma_s^2$
2. $E[V^2] = \sigma_v^2$

Image Source

Solution for finding w_{opt}:

Let us square the estimation error:

$$D^2 = [(w-1)S + wV]^2 = (w-1)^2S^2 + w^2V^2 + 2(w-1)wVS$$

The mean (expectation) square error $E[D^2]$ is then:

$$E[D^2] = (w-1)^2E[S^2] + w^2E[V^2] + 2(w-1)wE[VS] = (w-1)^2\sigma_s^2 + w^2\sigma_v^2$$

The derivative of this quantity is w.r.t *w* is:

$$\frac{d}{dw}E[D^2] = \frac{d}{dw}((w-1)^2\sigma_s^2 + w^2\sigma_v^2) = 2(w-1)\sigma_s^2 + 2w\sigma_v^2$$

Equating to zero provides the optimal value for w:

$$\frac{d}{dw}E[D^2] = 0$$

$$2(w-1)\sigma_s^2 + 2w\sigma_v^2 = 0$$

$$w_{opt} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2}$$

Image Source

Speech Enhancement- Noise Reduction

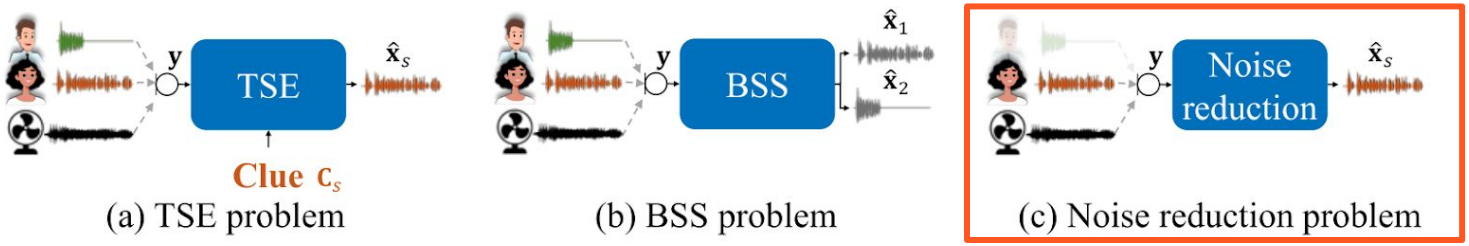


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter

$$D = \hat{S} - S = (w-1)S + wV \qquad w_{opt} = argmin_w E[D^2]$$

What does it mean?- intuition

w_{opt} minimizes $D = \hat{S} - S$, the *mean-square distance between the desired and estimated signal.*

Let's have a look on *w*:

$$w_{opt} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2}\frac{\sigma_v^2}{\sigma_v^2} = \frac{SNR}{1+SNR}$$

Image Source

Speech Enhancement- Noise Reduction

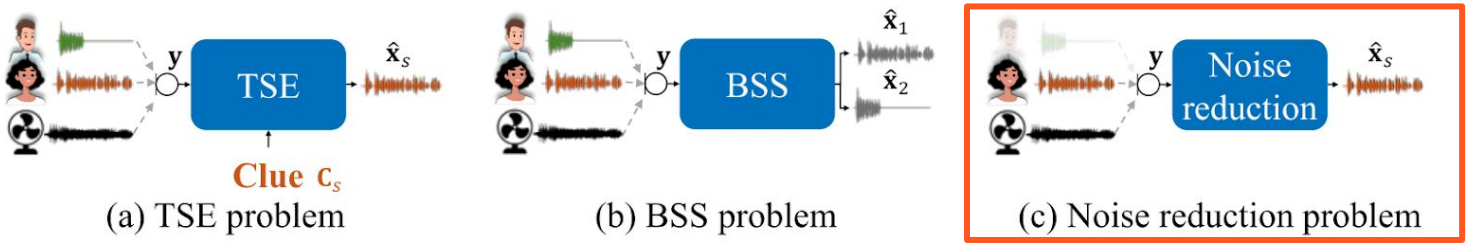


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter

$$D = \hat{S} - S = (w-1)S + wV \qquad w_{opt} = argmin_w E[D^2]$$

$$w_{opt} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2}\frac{\sigma_v^2}{\sigma_v^2} = \frac{SNR}{1+SNR}$$

The variance of the signal can be seen as its energy.

This is a monotonic function of the SNR:

- when $SNR \to \infty$, $w_{opt} \to 1$
- when $SNR \to 0$, $w_{opt} \to 0$

Image Source

Speech Enhancement- Noise Reduction

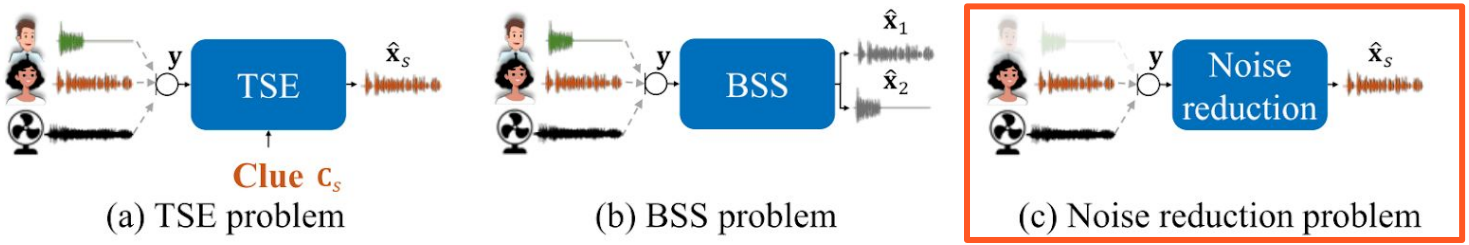


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter

$$D = \hat{S} - S = (w-1)S + wV \qquad w_{opt} = argmin_w E[D^2]$$

$$w_{opt} = \frac{\sigma_s^2}{\sigma_s^2+\sigma_v^2} = \frac{\frac{s}{\sigma_s^2+\sigma_v^2}}{\frac{v}{\sigma_v^2}} = \frac{SNR}{1+SNR}$$

- when $SNR \to \infty$, $w_{opt} \to 1$
- when $SNR \to 0$, $w_{opt} \to 0$

- When the signal is stronger than the noise, S is remained untouched, (multiply it by 1)- because it's already (relatively) clean.
- When the noise is stronger than the signal, we want to zero the output as we assume we cannot reconstruct the desired signal.

In between there is a monotonic shift between the edge cases.

Image Source

Speech Enhancement- Noise Reduction

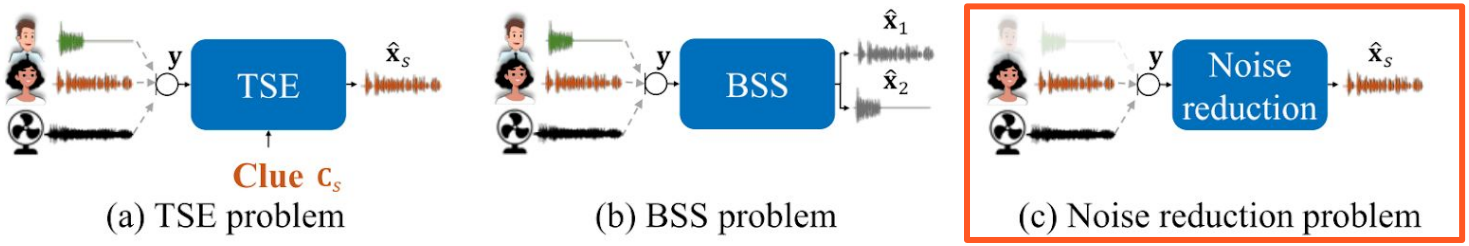


Fig. 2. Comparison of TSE with BSS and noise reduction

Wiener Filter

- This is a single channel Wiener filter, it has an extension to multi-channel
- This can be generalized to complex-valued signals (in the spectral domain).

Image Source

Speech Enhancement- Noise Reduction

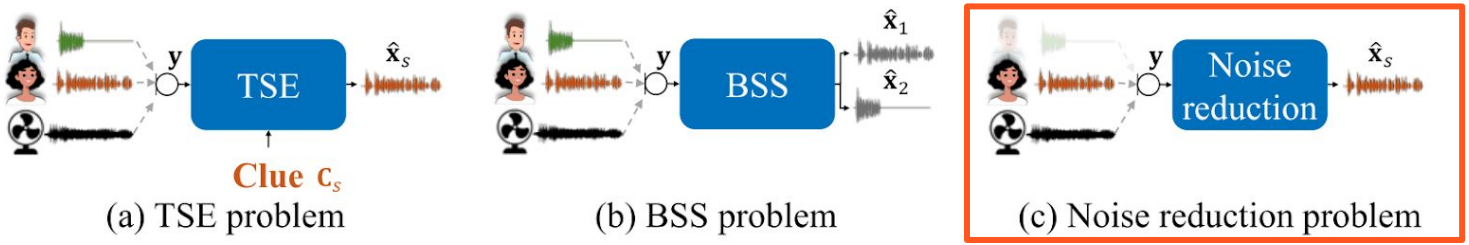


Fig. 2. Comparison of TSE with BSS and noise reduction

- Spectral subtraction
- Wiener filter
- **Beamforming - Credit David Levine**
- NN-based methods

Image Source

Speech Enhancement- Noise Reduction

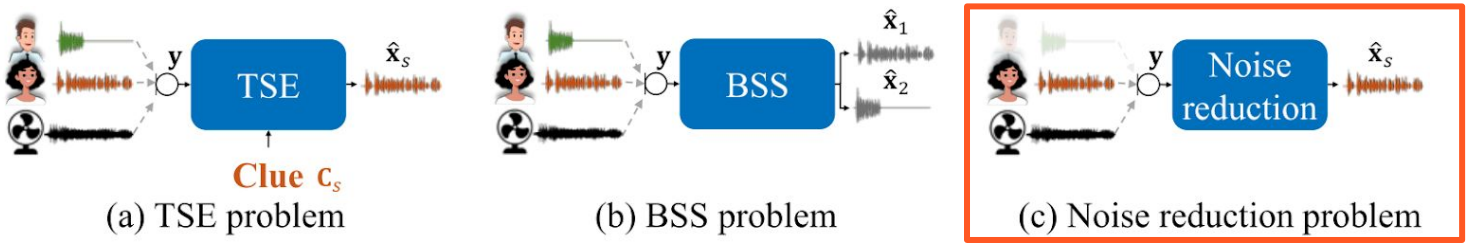
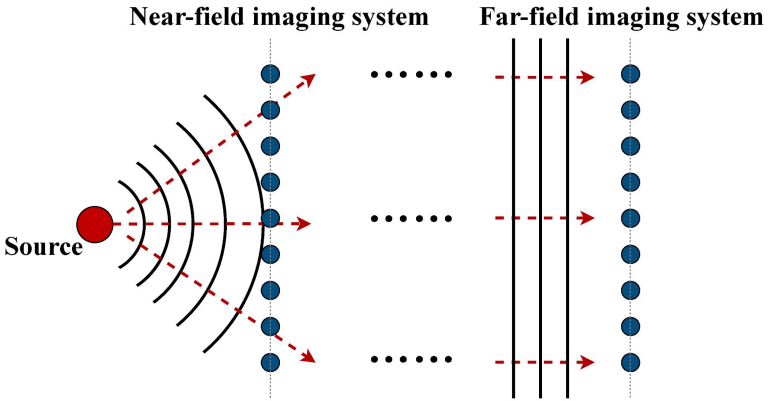


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming- Near and Far field



Source 1, 2

Speech Enhancement- Noise Reduction

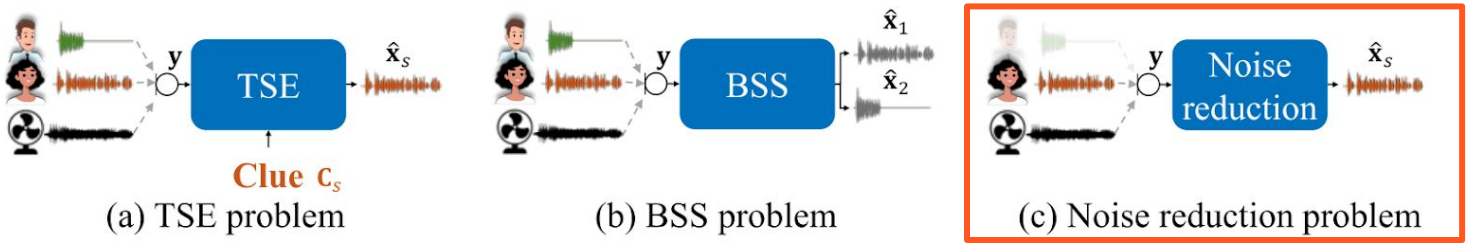
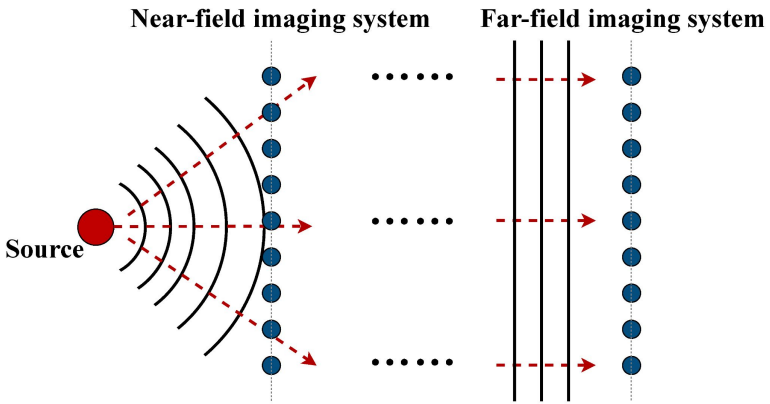


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming- Near and Far field

The term far field refers to a scenario where:

- The wave front is approximately a plane
- The attenuation between microphones within the array is negligible



Source 1, 2

Speech Enhancement- Noise Reduction

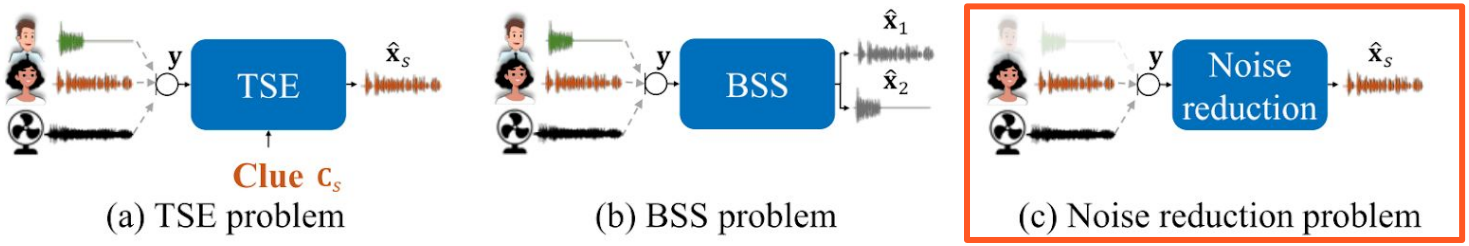


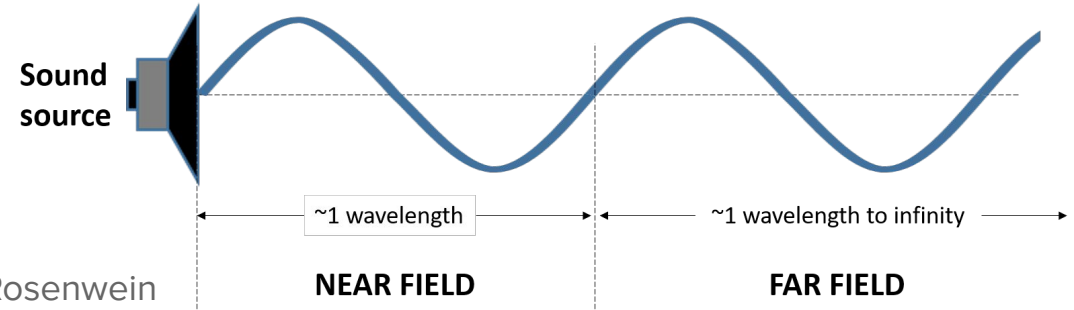
Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming- Near and Far field

How do we determine if a signal should be treated as a near / far field?

In order to assume far-field, two rules should be satisfied:

1. Frequency dependent: $\lambda << R$ The wave-length λ should be much smaller than the distance propagated R



Source 1, 2

Speech Enhancement- Noise Reduction

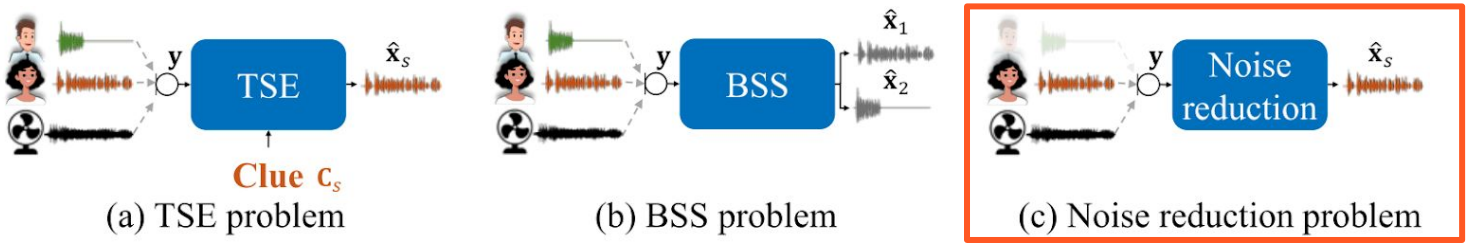


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming- Near and Far field

How do we determine if a signal should be treated as a near / far field?

In order to assume far-field, two rules should be satisfied:

1. Frequency dependent: $\lambda << R$ The wave-length λ should be much smaller than the distance propagated R
2. $\Delta << R$ The array aperture Δ (distance between microphones) must be much smaller than the distance propagated R,

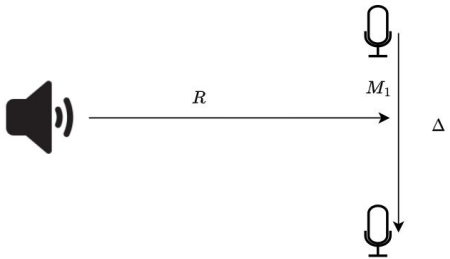


Image Source

Speech Enhancement- Noise Reduction

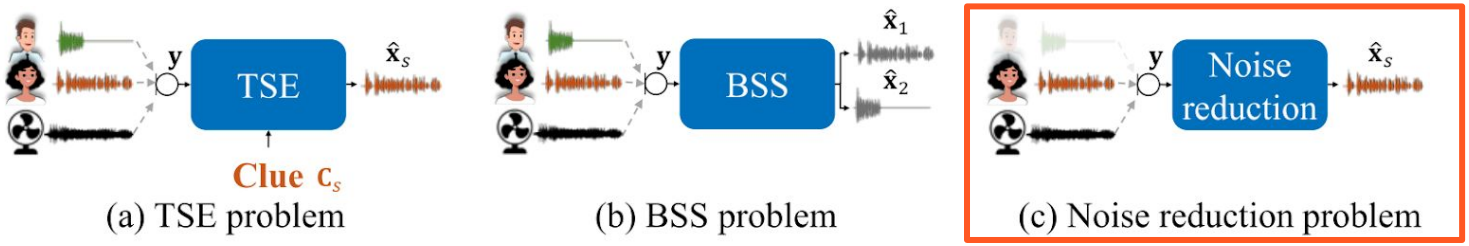


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming

The goal of beamforming can be seen as making a **directional microphone** from a microphone array.

- Each of the microphones within the array has no directionality
- The combination of them can form a ‘directional microphone’.

Image Source

Speech Enhancement- Noise Reduction

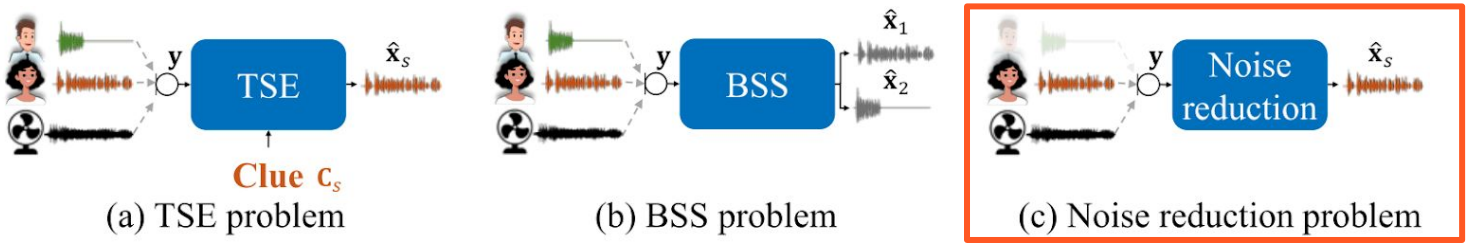


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming

The goal of beamforming can be seen as making a **directional microphone** from a microphone array.

- Each of the microphones within the array has no directionality
- The combination of them can form a ‘directional microphone’.

*The directionality is sometimes referred to as **steering**, as one wants to be able to control the angle of the directionality.

Image Source

Speech Enhancement- Noise Reduction

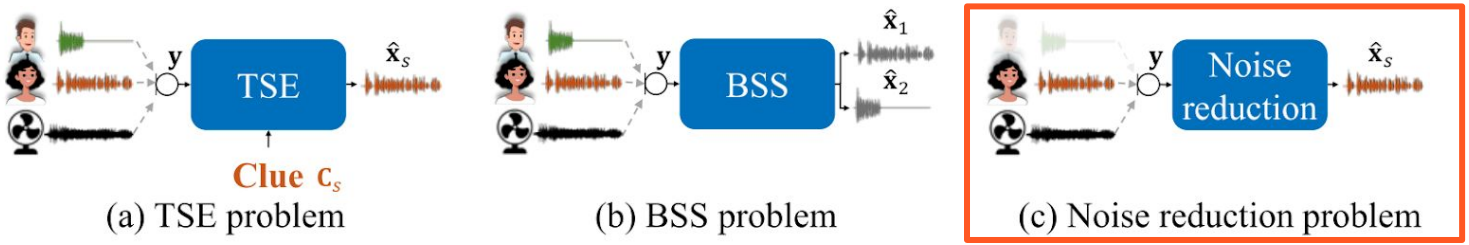


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum

Most elementary solution for beam forming: ‘Delay and Sum’.

Image Source

Speech Enhancement- Noise Reduction

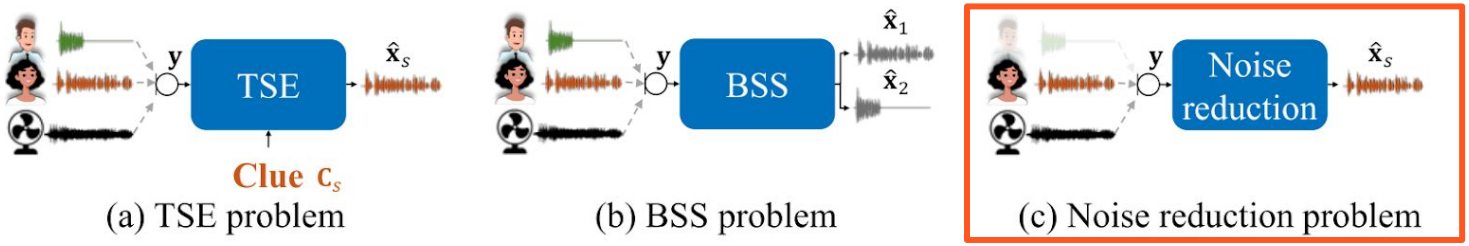


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum

- Delay and sum is a simple method that literally adds (averages) delayed versions of the microphones (each microphone can have a different delay) within the array.
- The amount of delay defines the directionality (angle) that the microphone array will form.
- Because we assume a far field scenario, the acoustic transfer function (ATF) of the microphone changes only in the phase (and not the magnitude), so the delay defines the steering vector entirely.

Image Source

Speech Enhancement- Noise Reduction

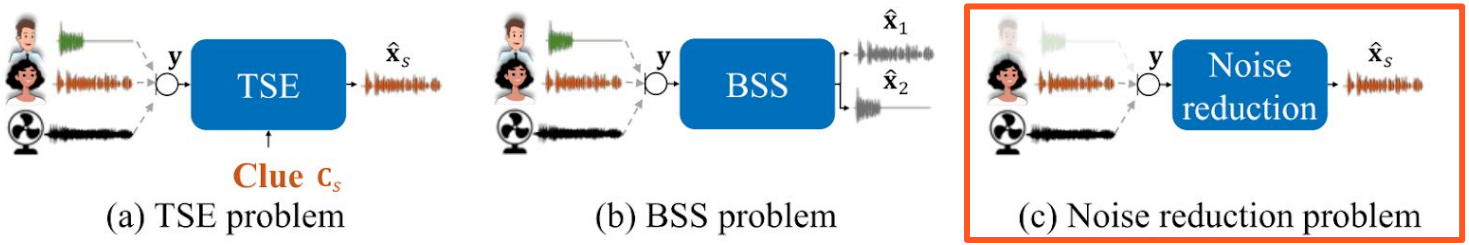


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

An Example:

Given a microphone array of size 2, which are placed one above the other on the same plane *Z=0*. only their Y coordinate is different. Each microphone sample rate is $F_s = 16K\ [Hz]$, the y-axis distance between the microphones is $d = 0.03\ [m]$. We know that the speed of sound is $C = 343\ [\frac{m}{s}]$.

Image Source

Speech Enhancement- Noise Reduction

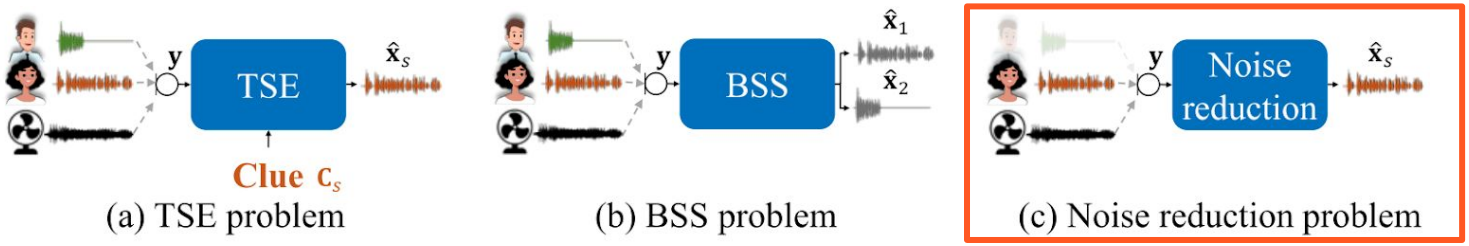


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- The audio/acoustic wave propagates from left to right
- The wave front is perpendicular to its direction of propagation
- Hence it arrives at the same time (far field assumption) to each of the microphones M1 and M2.

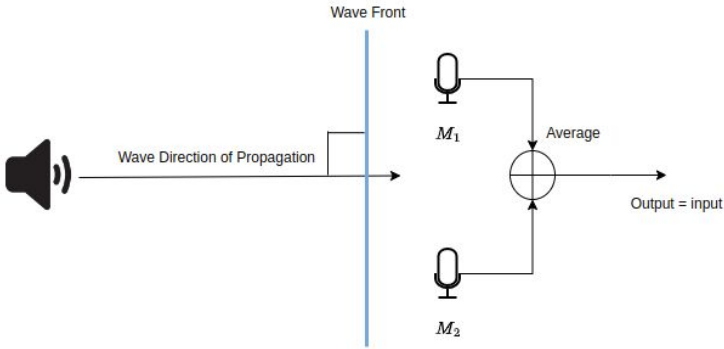


Image Source

Speech Enhancement- Noise Reduction

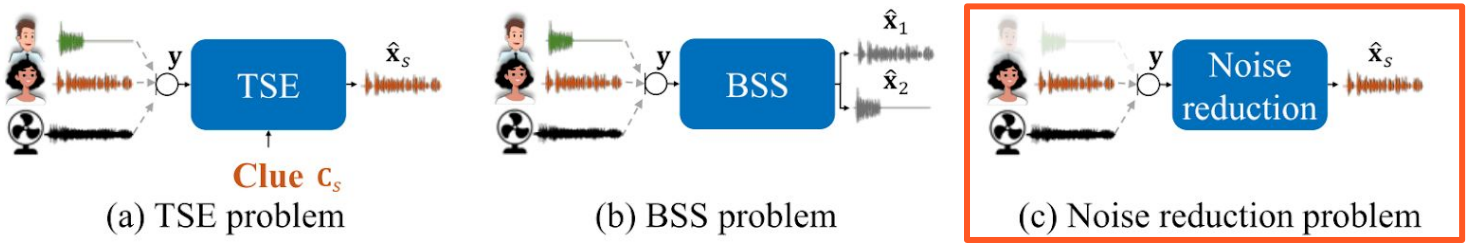


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- Assuming far-field, the same amount of energy is received in the different microphones.
- Averaging the outputs of the microphones results in the input as desired.

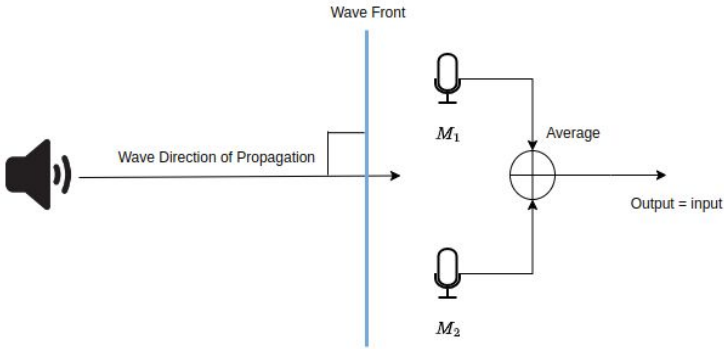


Image Source

Speech Enhancement- Noise Reduction

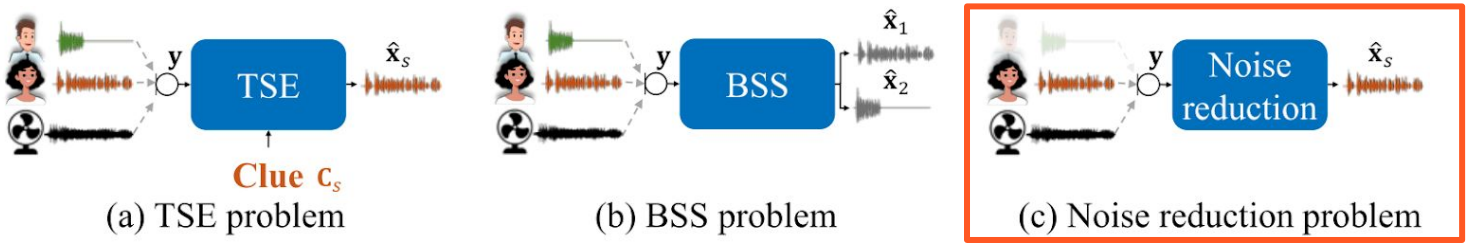


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- when the waveform propagates from top to bottom the audio will arrive first to microphone M1 and then to microphone M2
- The delay (in samples) between the microphones

$$d = 0.03\ [m]$$
$$F_s = 16,000\ [sample\ /\ sec]$$
$$C = 343\ [m/sec]$$
$$delay = \frac{d*F_s}{C} = \frac{0.03*16000}{343} = \sim 1.4[samples]$$

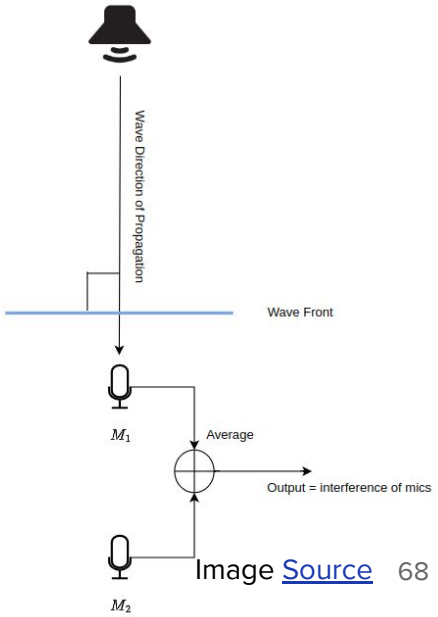


Image Source

Speech Enhancement- Noise Reduction

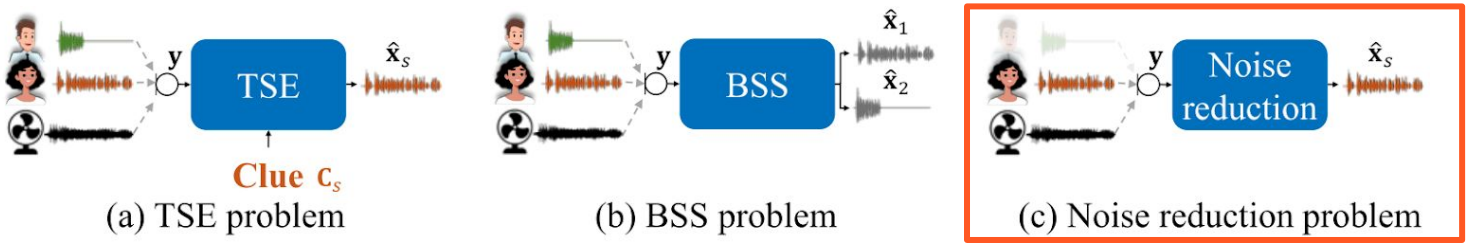


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- when the waveform propagates from top to bottom the audio will arrive first to microphone M1 and then to microphone M2
- The delay (in samples) between the microphones
- Meaning that the microphone M1 will acquire the signal 1.4 samples before microphone M2.

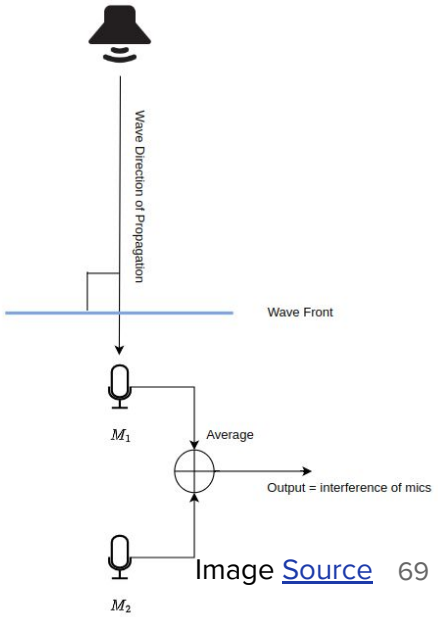


Image Source

Speech Enhancement- Noise Reduction

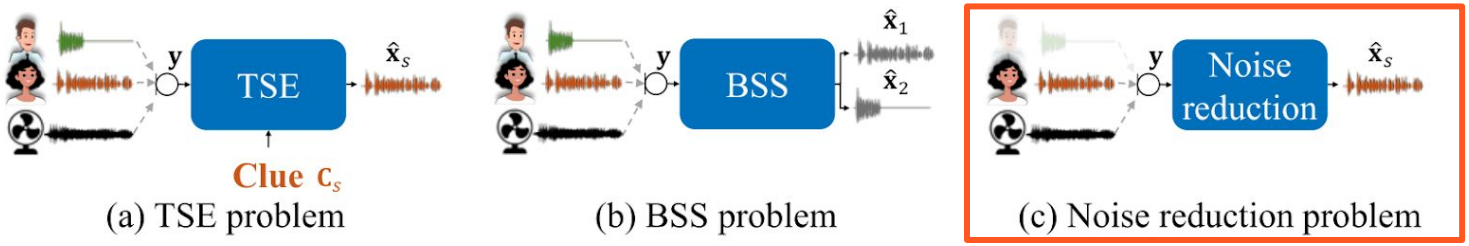


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- when the waveform propagates from top to bottom the audio will arrive first to microphone M1 and then to microphone M2
- When we average the signals, we will get a **distorted version** of the input.
- If we would like to fix it, we need to delay microphone 1 by 1.4 samples, and then average the two signals.

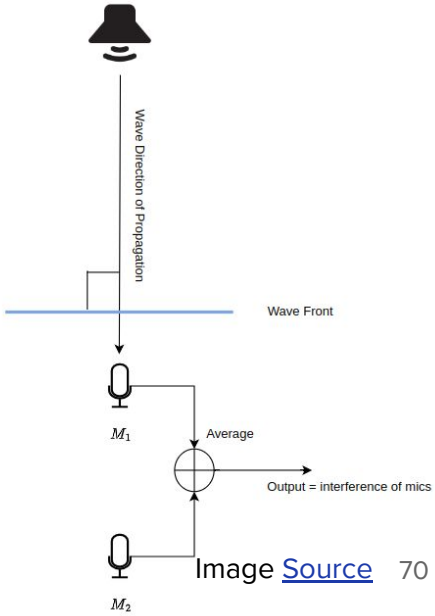


Image Source

Speech Enhancement- Noise Reduction

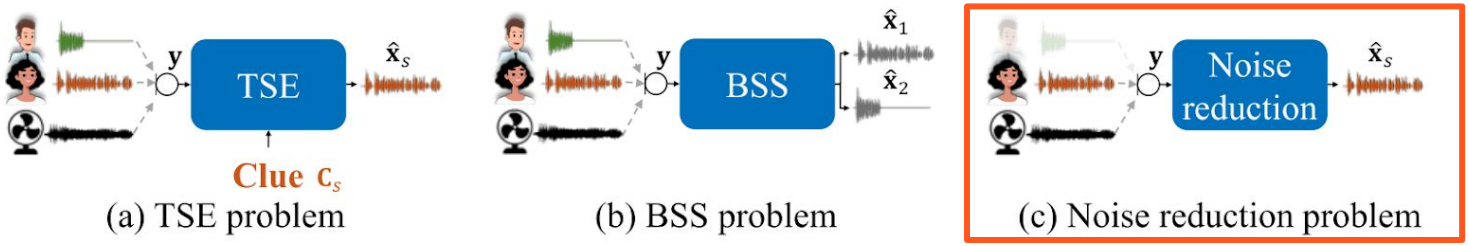


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

- when the waveform propagates from top to bottom the audio will arrive first to microphone M1 and then to microphone M2
- When [illegible] the input.
- If we would like to fix it, we need to delay microphone 1 by 1.4 samples, and then average the two signals.

This is exactly how the ‘delay and sum’ method works.

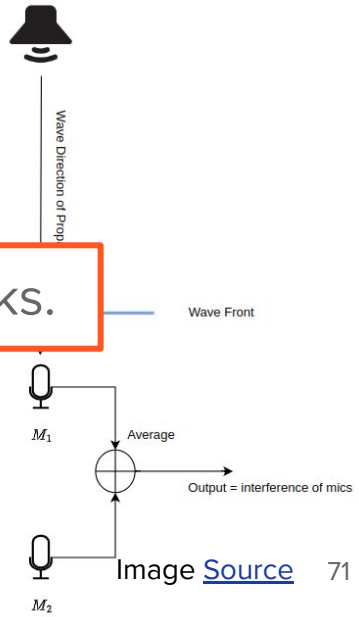


Image Source

Speech Enhancement- Noise Reduction

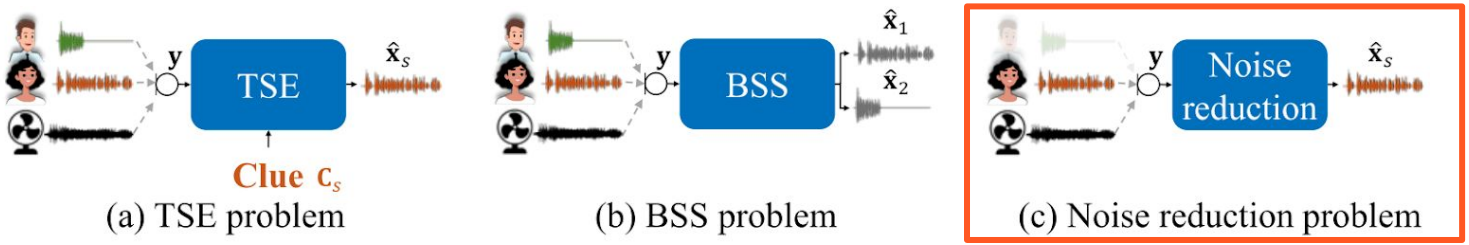


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum Example

General case

- The green line indicates the distance that it will take until the wave front will arrive at microphone M2 after it arrived to microphone M1, given the angle it is propagating and far-field assumption.

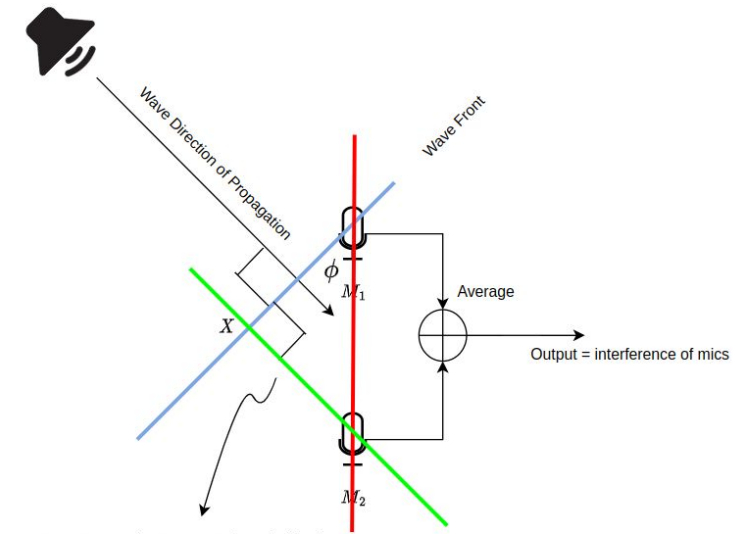


Image Source

Speech Enhancement- Noise Reduction

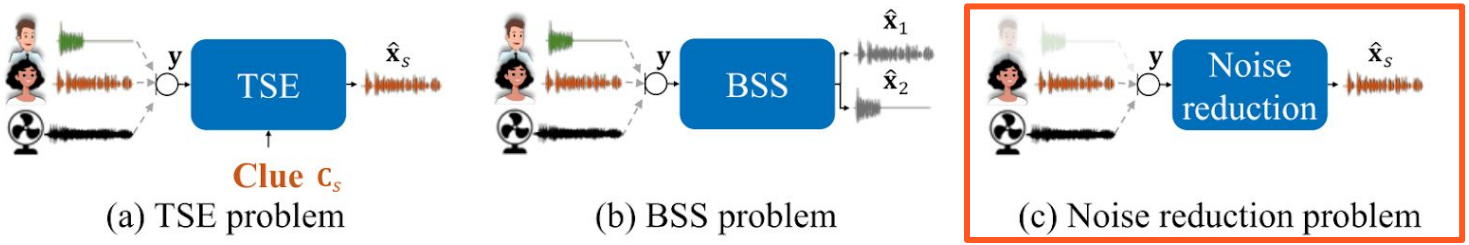


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum

So how does it work in practice?

- We set the angle we want to amplify.
- We adjust the steering vector (here delays)
- Apply the delay and sum algorithm.

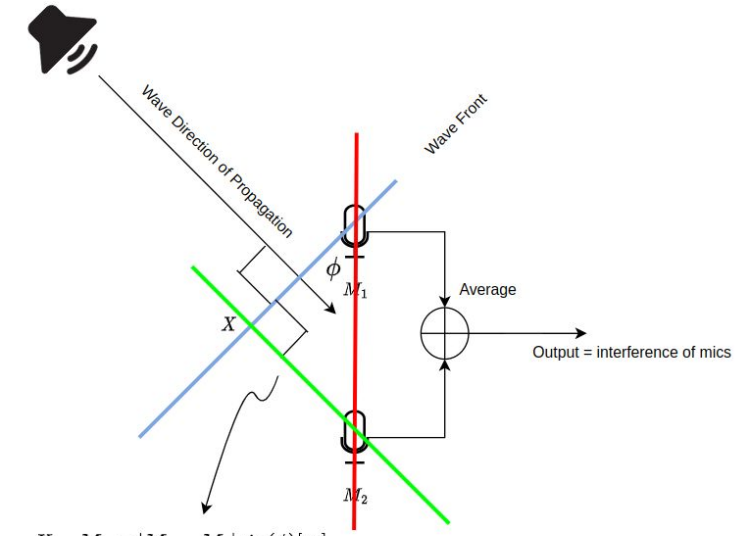


Image Source

Speech Enhancement- Noise Reduction

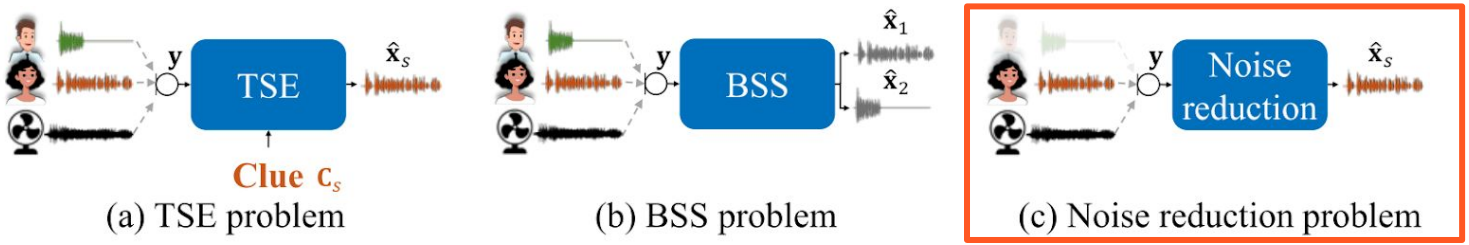
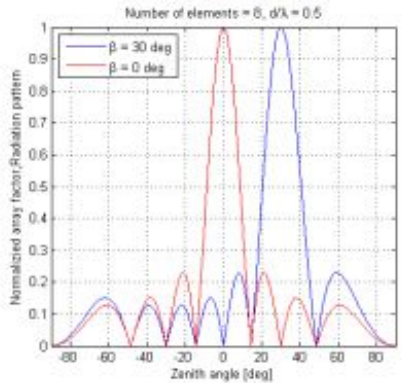


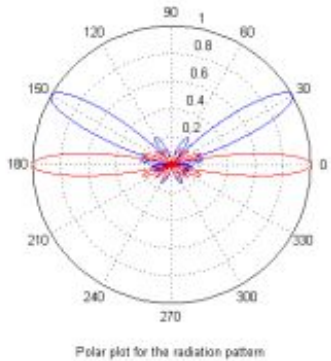
Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum

So how does it work in practice?

- We set the angle we want to amplify.
- We adjust the steering vector (here delays)
- Apply the delay and sum algorithm.





As there is a different attenuation per angle, researchers often plot the '**beam pattern**'.

Source 1, 2

Speech Enhancement- Noise Reduction

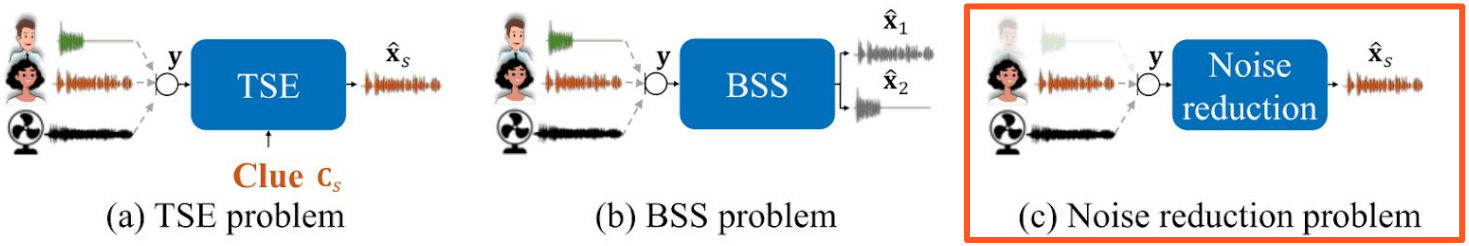


Fig. 2. Comparison of TSE with BSS and noise reduction

Beamforming - Delay and Sum

Input **0deg**

Out DoA **90deg**

Out DoA **0deg**

Image Source

Speech Enhancement- Noise Reduction

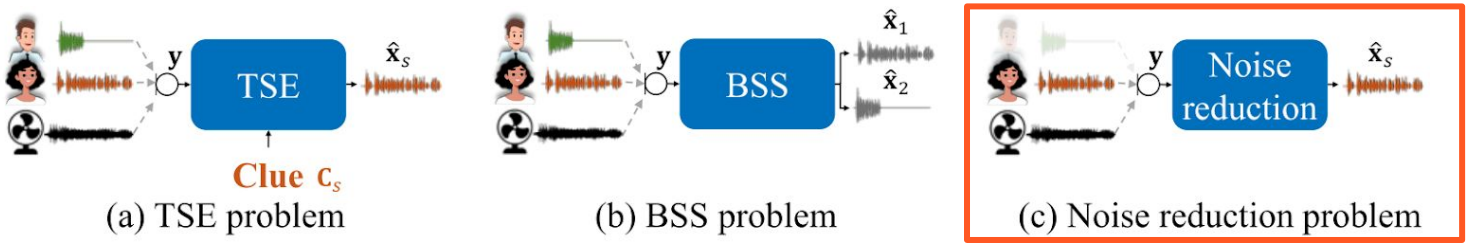


Fig. 2. Comparison of TSE with BSS and noise reduction

- Spectral subtraction
- Wiener filter
- Beamforming
- **NN-based methods**

Image Source

Speech Enhancement- Noise Reduction

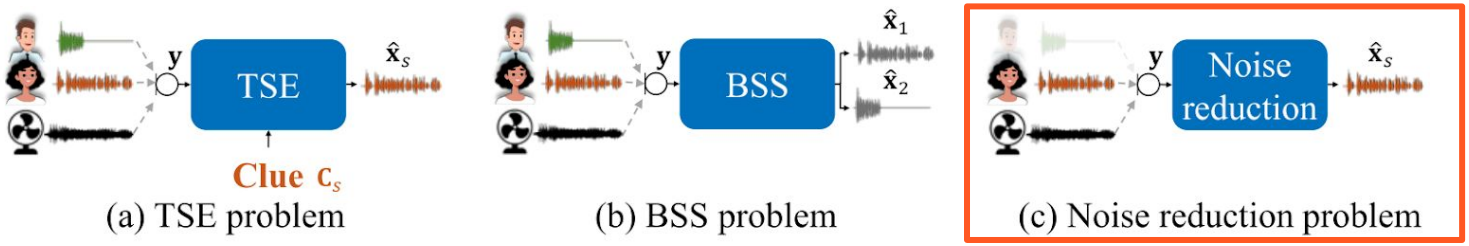


Fig. 2. Comparison of TSE with BSS and noise reduction

NN-based methods - Demucs

- Demucs is a deep learning model that directly operates on the **raw waveform** and generates a waveform for each source
- Demucs is inspired by models for **music synthesis** rather than masking approaches.

Source 1, 2

Speech Enhancement- Noise Reduction

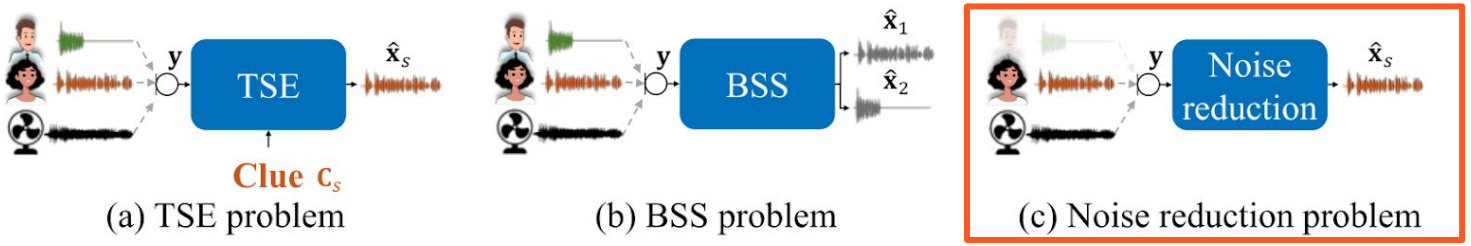
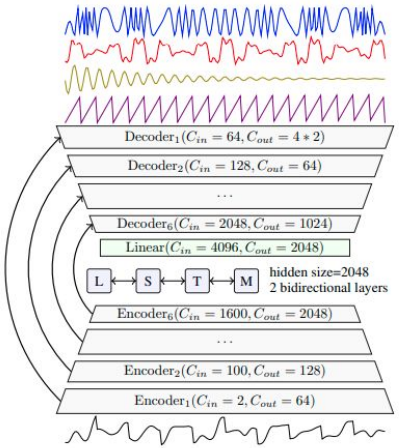


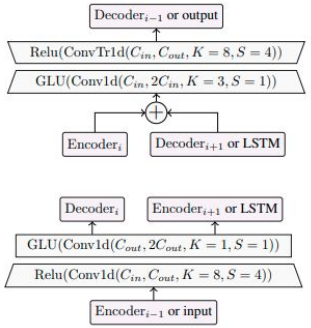
Fig. 2. Comparison of TSE with BSS and noise reduction

NN-based methods - Demucs

- It is a **U-net architecture** with a convolutional encoder and a decoder based on wide transposed convolutions with large strides.
- The encoder and decoder are linked with skip U-Net connections.
- Bidirectional LSTM between the encoder and the decoder.



(a) Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections.



(b) Detailed view of the layers Decoder$_i$ on the top and Encoder$_i$ on the bottom. Arrows represent connections to other parts of the model. For convolutions, C_in (resp C_out) is the number of input channels (resp output), K the kernel size and S the stride.

Figure 2: Demucs complete architecture on the left, with detailed representation of the encoder and decoder layers on the right.

Speech Enhancement- Noise Reduction

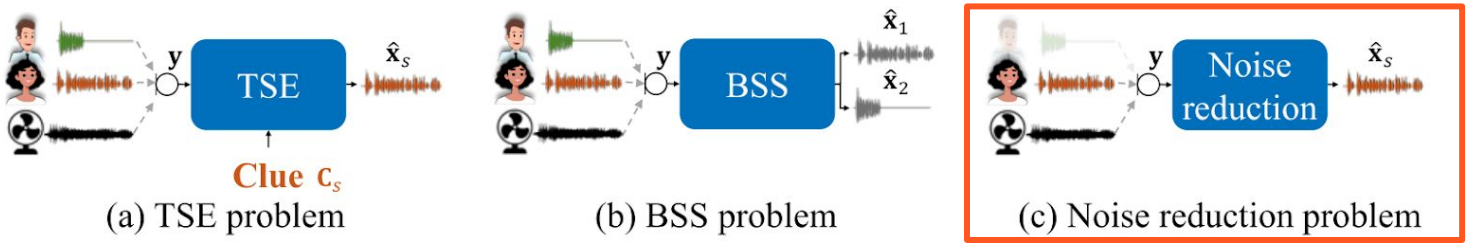


Fig. 2. Comparison of TSE with BSS and noise reduction

NN-based methods - Demucs

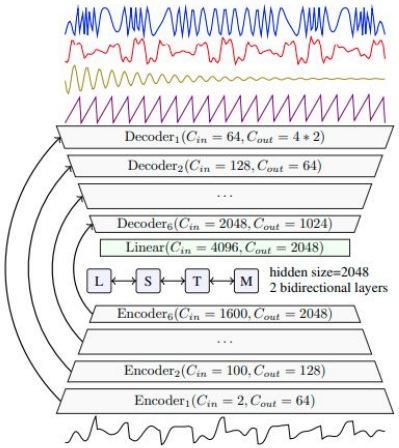
- Objective

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

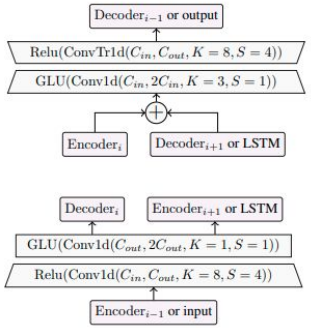
$$L_{\text{stft}}(\boldsymbol{y},\hat{\boldsymbol{y}})=L_{sc}(\boldsymbol{y},\hat{\boldsymbol{y}})+L_{mag}(\boldsymbol{y},\hat{\boldsymbol{y}})$$

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

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



(a) Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections.



(b) Detailed view of the layers Decoder$_i$ on the top and Encoder$_i$ on the bottom. Arrows represent connections to other parts of the model. For convolutions, C_in (resp C_out) is the number of input channels (resp output), K the kernel size and S the stride.

Figure 2: Demucs complete architecture on the left, with detailed representation of the encoder and decoder layers on the right.

Speech Enhancement- Noise Reduction

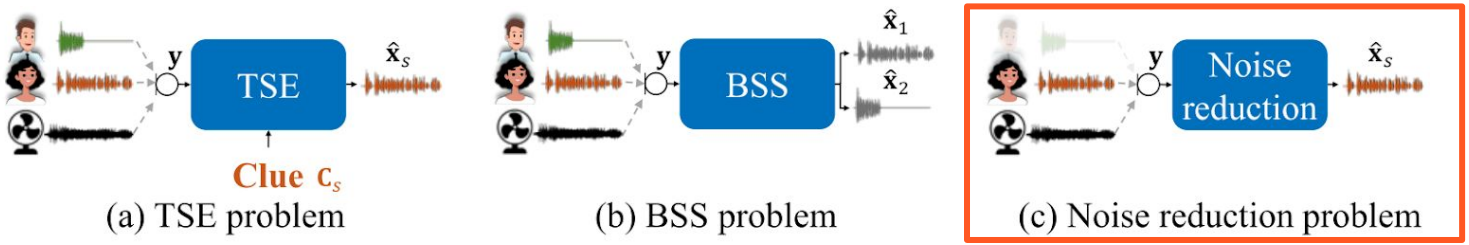


Fig. 2. Comparison of TSE with BSS and noise reduction

- Spectral subtraction
- Wiener filter
- Beamforming
- NN-based methods

Image Source

Speech Enhancement- BSS

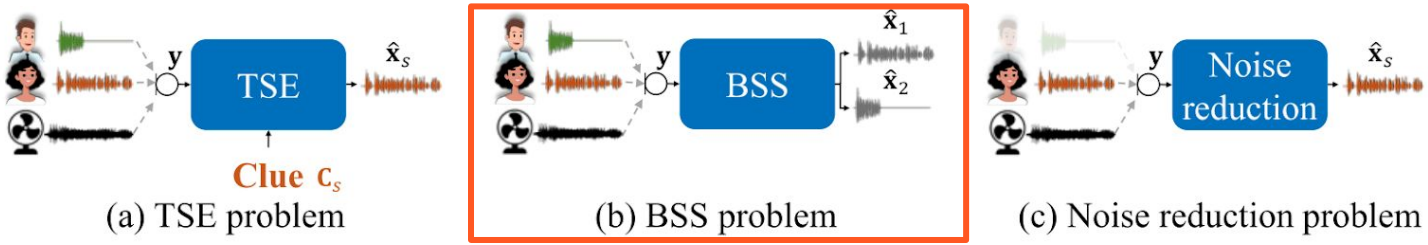


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- **Blind source separation.**
- Target speech enhancement.

Image Source

Speech Enhancement- BSS- Problem Formulation

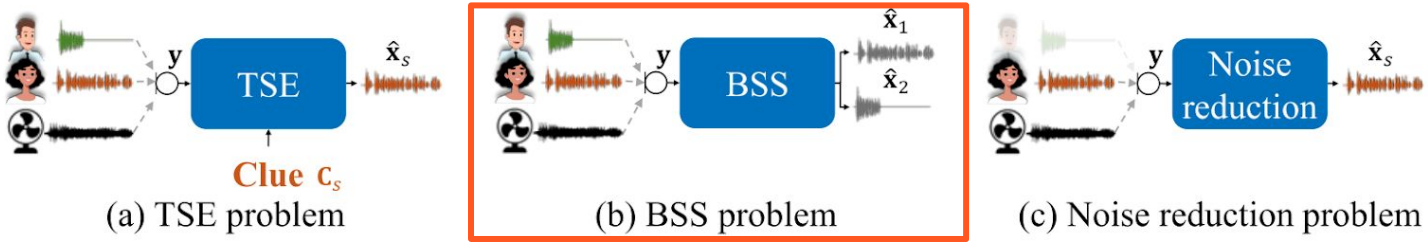


Fig. 2. Comparison of TSE with BSS and noise reduction

estimates all the source signals in a mixture without requiring clues:

$$\{\hat{x}_1,\ldots,\hat{x}_K\} = BSS(y;\theta_{BSS}),$$

where $BSS(\cdot;\theta_{BSS})$ represents a separation system with parameters θ_{BSS}, $\hat{x}_k$ are the estimates of the speech sources, and K is the number of sources in the mixture. The number of sources ***K*** **must be known or estimated**.

Image Source

Speech Enhancement- BSS- Problem Formulation

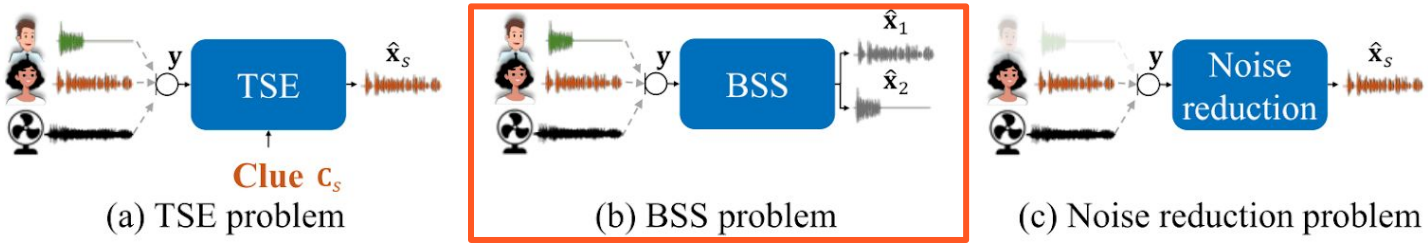


Fig. 2. Comparison of TSE with BSS and noise reduction

The simplest BSS model assumes the existence of n independent signals $s_1(t)$, ..., $s_n(t)$ and the observation of as many mixtures $x_1(t)$, ..., $x_n(t)$, these mixtures being linear, i.e.

$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t)$ for each i = 1, n. (in the above equation we replace *y* is *s*)

Image Source

Speech Enhancement- BSS- Problem Formulation

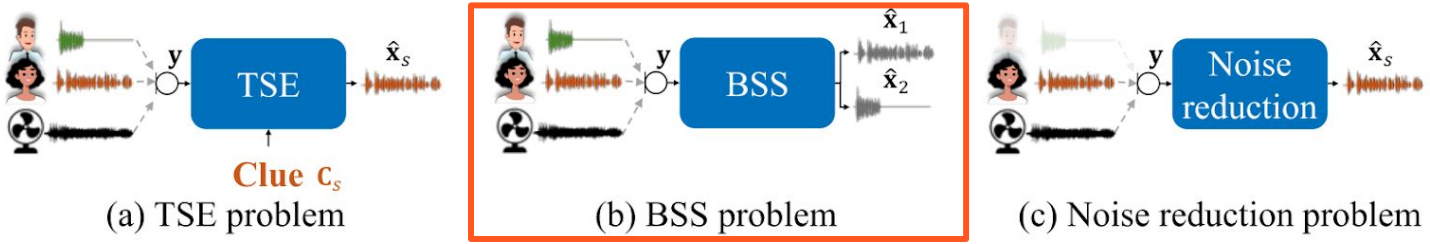
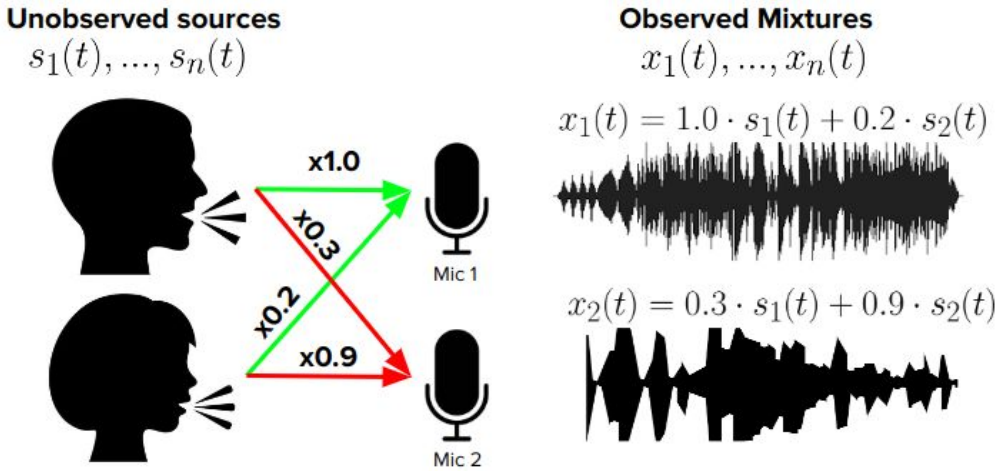


Fig. 2. Comparison of TSE with BSS and noise reduction



Source 1, 2, 3, 4, 5

Speech Enhancement- BSS- Problem Formulation

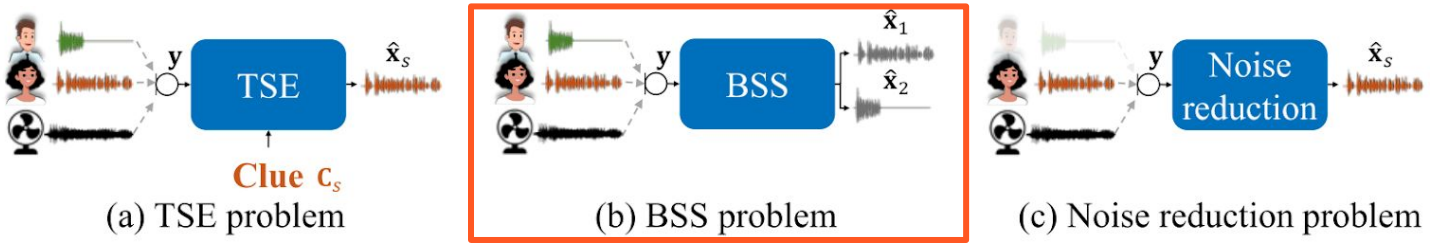


Fig. 2. Comparison of TSE with BSS and noise reduction

- Since the number of outputs is given by the number of sources, the number of sources K must be known or estimated - **may be challenging.**
- There is a global permutation ambiguity problem between the outputs and the speakers.

Image Source

Speech Enhancement- BSS- Background

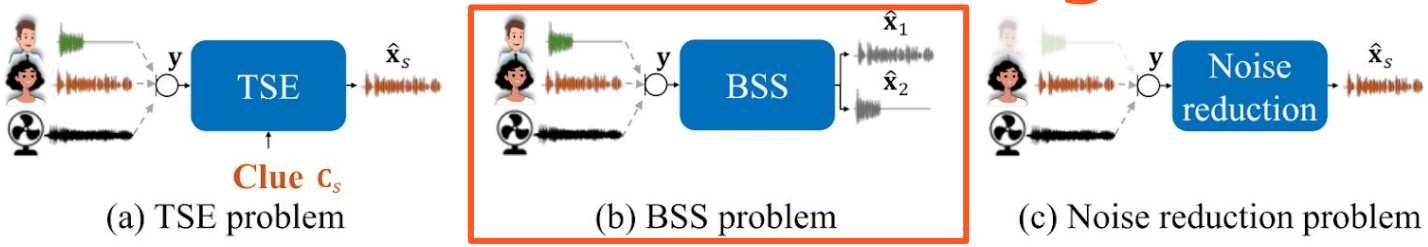


Fig. 2. Comparison of TSE with BSS and noise reduction

The technique of blind source separation (BSS) has been studied for decades, and the research is still in progress.

The adjective ‘blind’ stresses the fact that:

- the source signals are not observed
- No information is available about the mixture.

Image Source

Speech Enhancement- BSS- Background

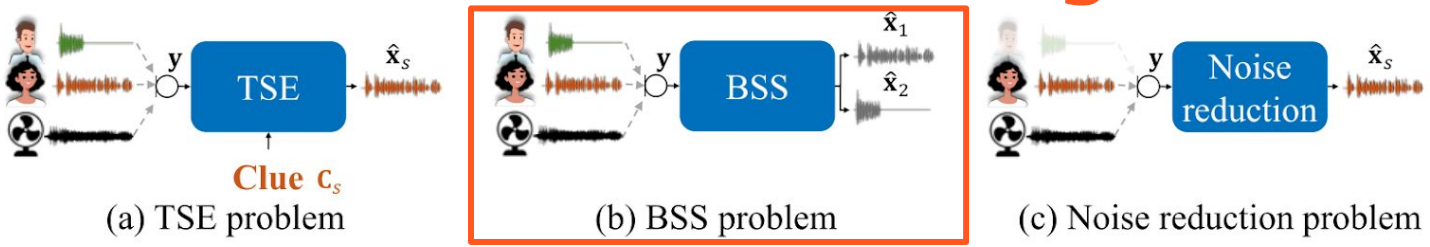


Fig. 2. Comparison of TSE with BSS and noise reduction

- Source separation is a fundamental problem in machine learning and signal processing
- An additional difficulty arises when the sources to separate belong to the same class of signals
  - separating overlapped speech
  - isolating appliance electric consumption from meter reading
  - separating overlapped fingerprints
  - retrieving individual compounds in chemical mixtures from spectroscopy

Source 1, 2

Speech Enhancement- BSS- Background

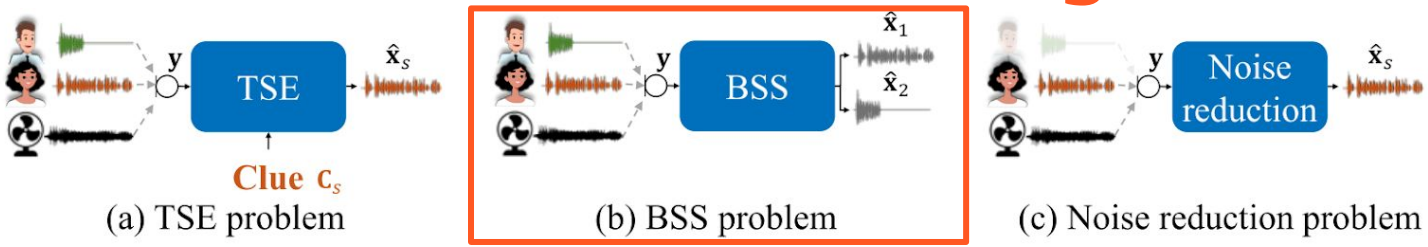


Fig. 2. Comparison of TSE with BSS and noise reduction

The technique of blind source separation (BSS) has been studied for decades, and the research is still in progress.

The adjective ‘blind’ stresses the fact that:

- the source signals are not observed
- No information is available about the mixture.

Image Source

Speech Enhancement- BSS- Background

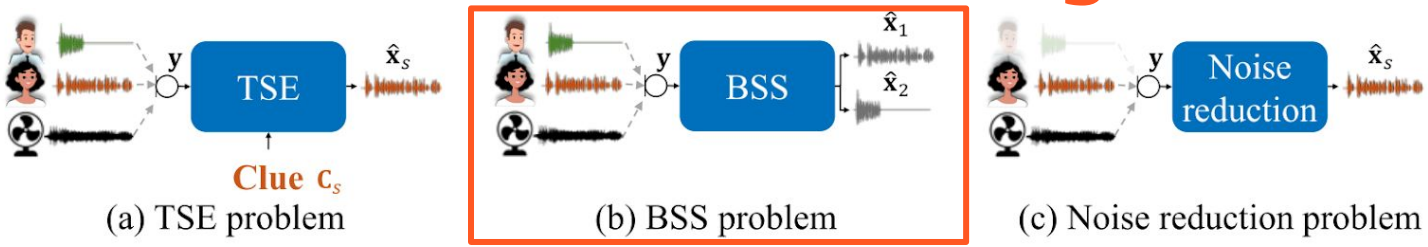


Fig. 2. Comparison of TSE with BSS and noise reduction

BSS **in audio** can be used in various applications, such as:

- Solving the cocktail party problem- was noticed by Cherry at 1953.
- Extracting the target speech in a noisy environment for better speech recognition results
- separating each musical instrumental part of an orchestra performance for music analysis.

Image Source

Speech Enhancement- BSS- Background

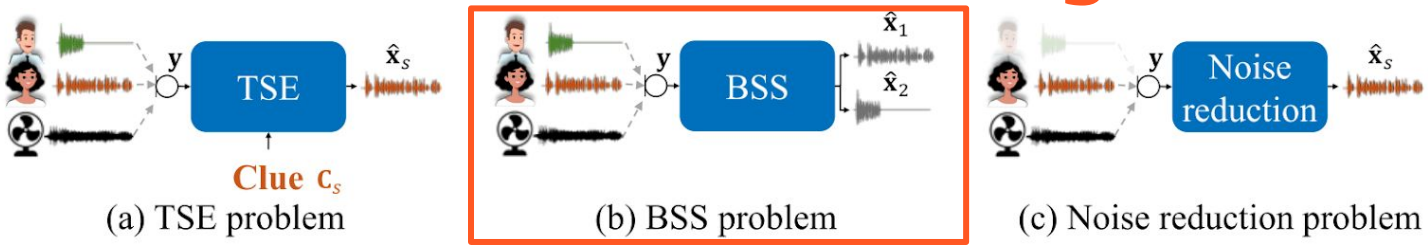


Fig. 2. Comparison of TSE with BSS and noise reduction

Various signal processing and machine learning methods have been proposed for BSS. They can be classified using **two categories**:

- number *M* of microphones used to observe sound mixtures.
  - Single channel (M = 1)
  - Multichannel M ≥ 2 -the spatial information of a source signal can be utilized for separation.
- Number *N* of source signals.
  - N <= M- the separation can be achieved using linear filters.
  - N > M- one popular approach is based on clustering.

Image Source

Speech Enhancement- BSS- Methods

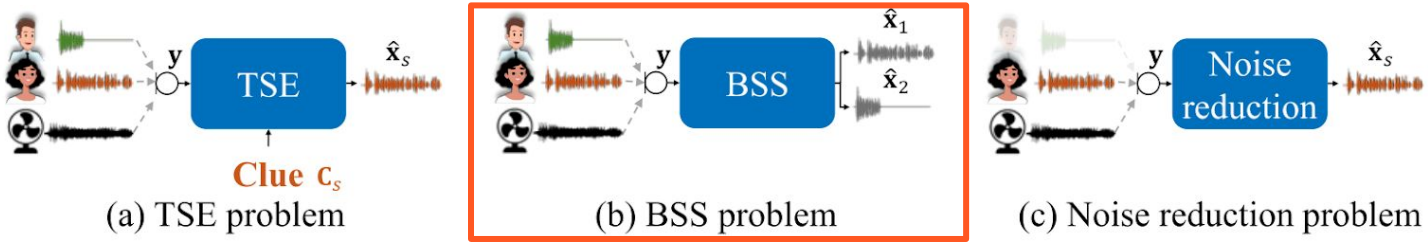


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
  - TF-GridNet
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI

Image Source

Speech Enhancement- BSS- Methods- Trad.

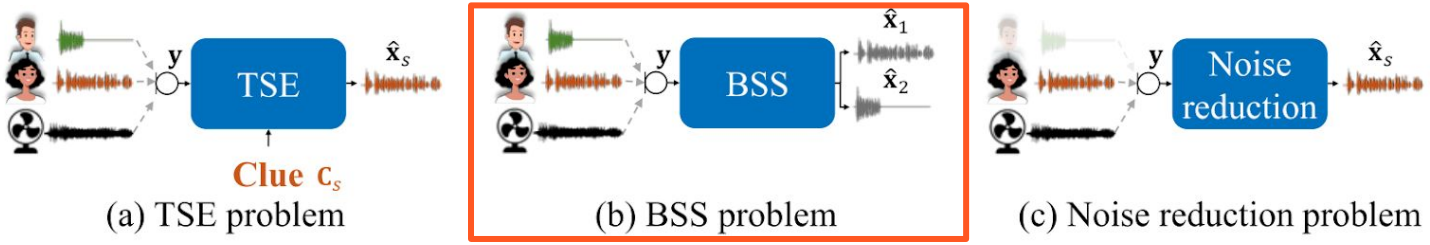


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Traditional:**
  - **Single mic: Non-Negative Matrix Factorization (NMF)**
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- Trad.

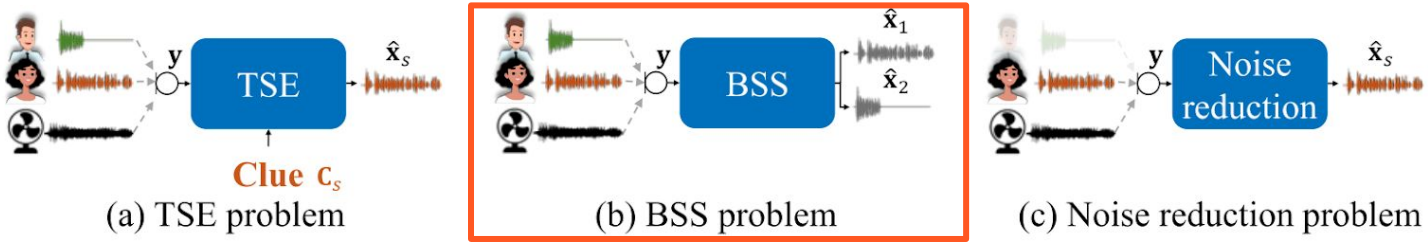


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF)

- Our goal is to find the **underlying structure of data in an unsupervised manner.**
- Family of linear algebra algorithms for identifying the latent structure (sources) in data represented as a non-negative matrix.

Image Source

Speech Enhancement- BSS- Methods- Trad.

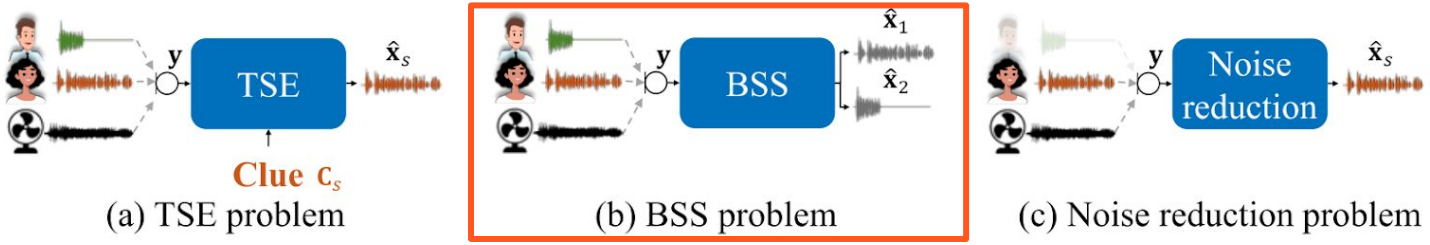
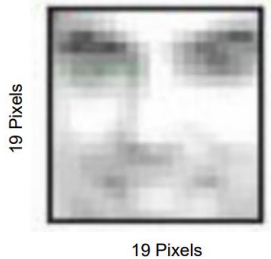


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Example

- A database of *N* images.
- Each image is of size *M=19x19=361* pixels.
  - Each pixel has a non-negative value (in the range of [0, 256])
- Reshape the image into a M=361 pixels (Mx1) column vector.
  - The dataset X will be of dimensions: (M pixels xN images)



Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

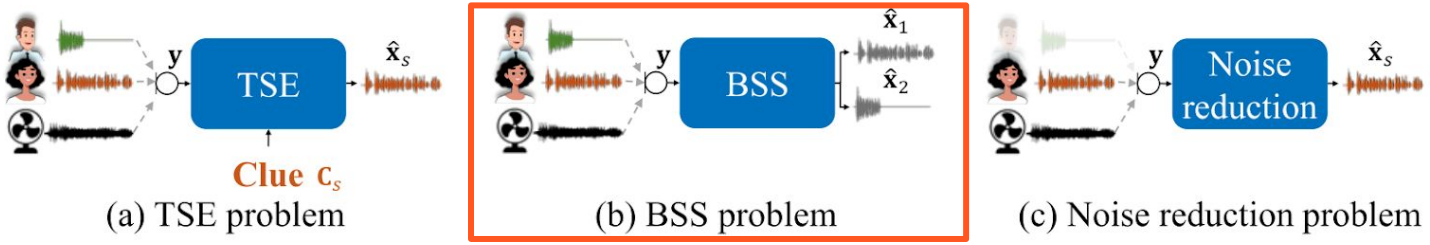
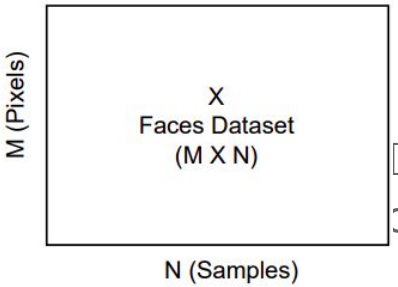
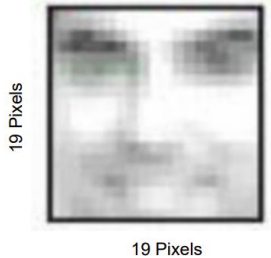


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Example

- A database of *N* images.
- Each image is of size *M=19*
  - Each pixel has a non-negati [0, 256])
- Reshape the image into a column vector.
  - The dataset X will be of dim





Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

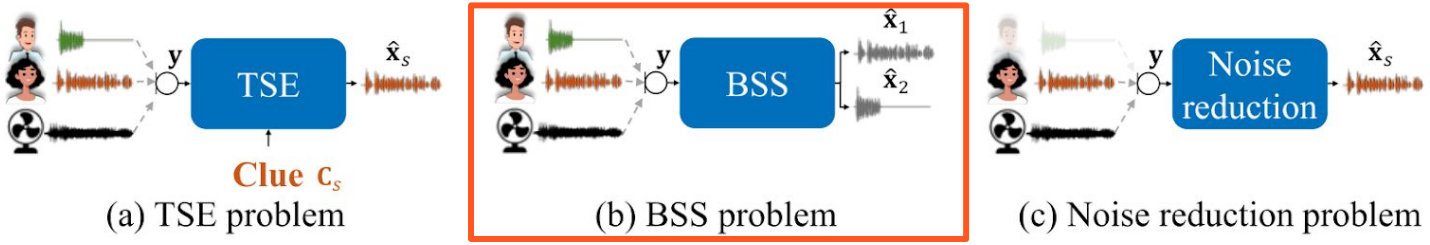
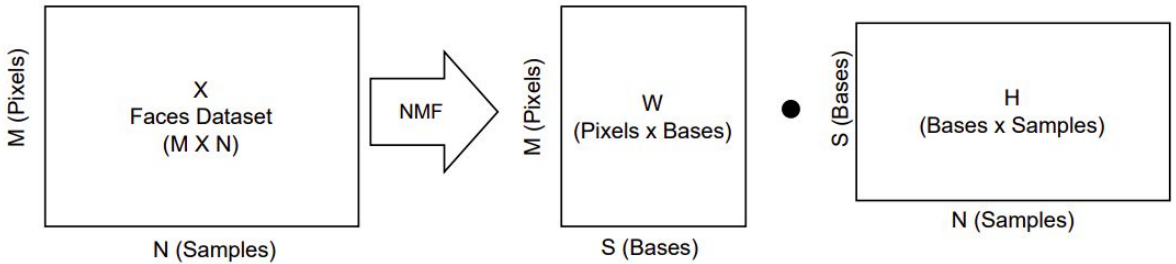


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Example

- We approximate X by two non-negative matrices W and H.
  - Decompose matrix X into two matrices.

$$X \approx WH$$
$$W, H \geq 0$$
$$X \geq 0$$



Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

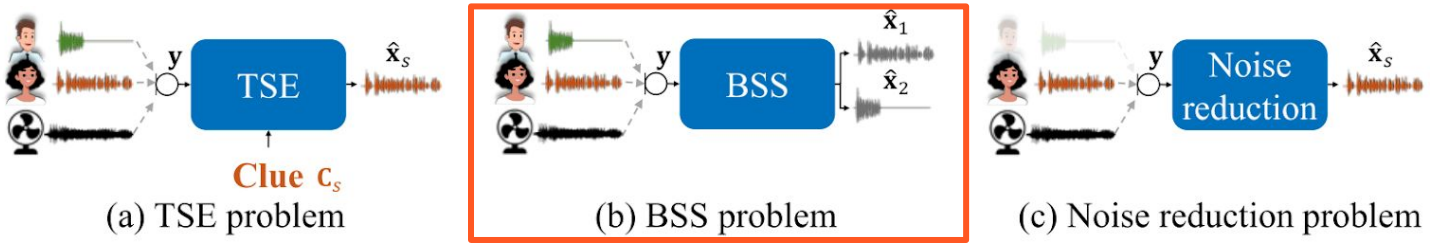
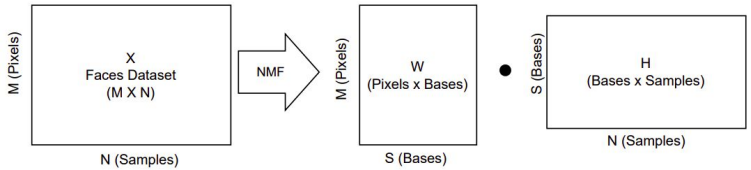


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Example



- We approximate X by two non-negative matrices W and H.
  - Decompose matrix X into two matrices.
- Where W and H have dimension of *MxS* and *SxN*, respectively.
- The inner dimension *S* is set by the user
  - constraint of: S < min(M,N).
  - *S*- # bases we would like to 'find' in our dataset. In this example, we will set S=49.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

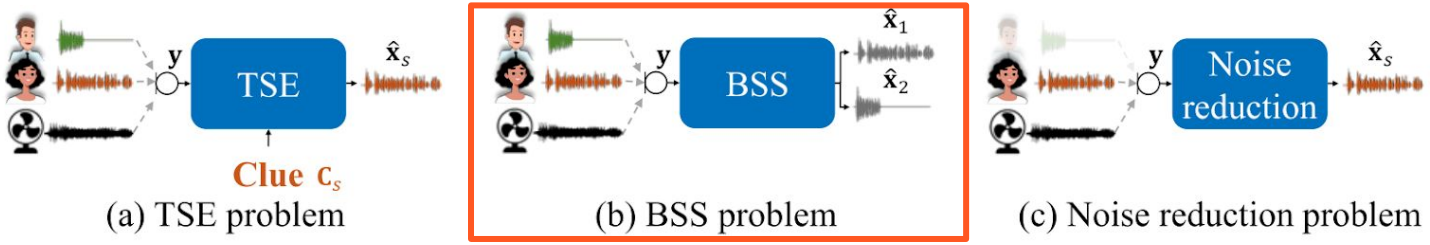
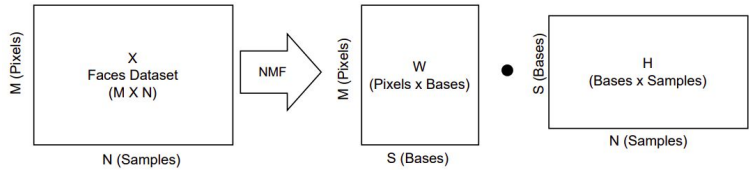


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Example



- Let's say we've found the optimal *W* and *H* matrices.
- Using *W* and *H*, we would like to reconstruct the i'th face image

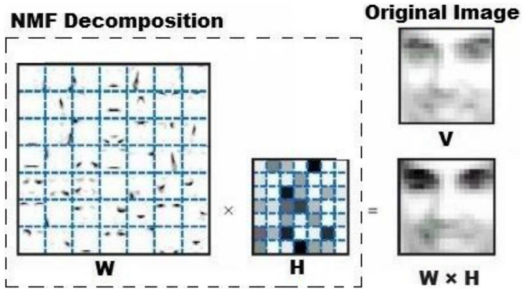


Fig. 2 Decompression of a face image using the NMF method

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

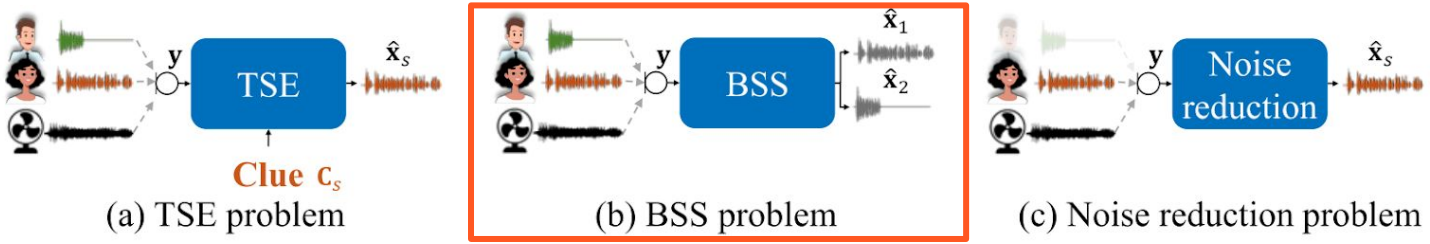
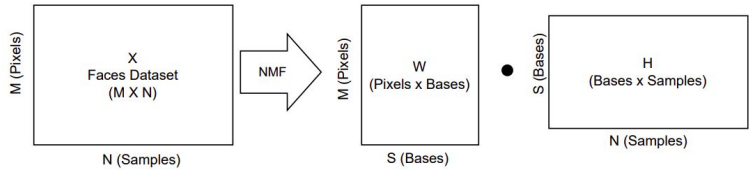


Fig. 2. Comparison of TSE with BSS and noise reduction



Non-Negative Matrix Factorization (NMF) - Example

W matrix:

- Can be observed as a codebook.
  - Each entry is a different base (S=7x7=49 bases).
  - Each base (code) has # pixels as in the original image.
  - We can see a structure- nose, eyebrows, etc.
- They reshaped W to a 7x7 matrix (S=49 entries) for visualization:
  - Each entry is an image of size 19x19 (M=361 pixels).
  - A matrix of size (SxM), where S=49 and M=361.

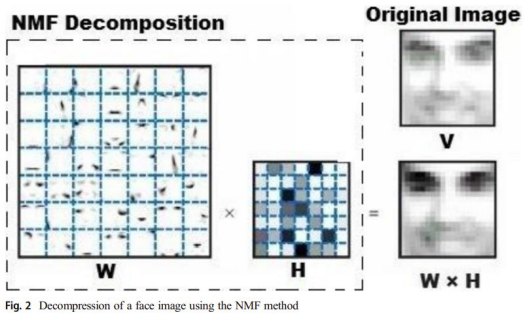


Fig. 2 Decompression of a face image using the NMF method

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

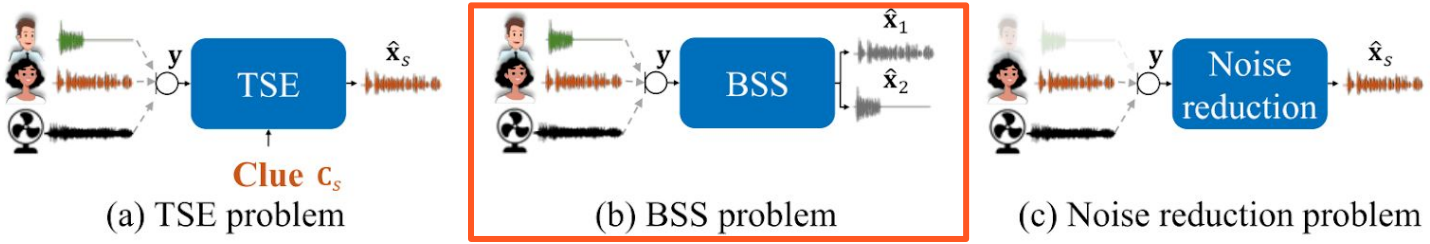
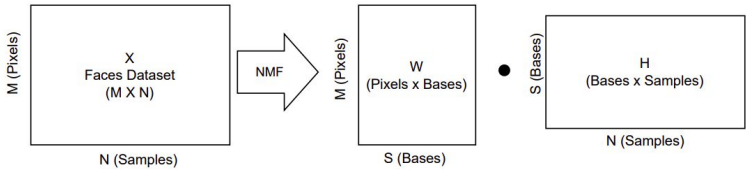


Fig. 2. Comparison of TSE with BSS and noise reduction



Non-Negative Matrix Factorization (NMF) - Example

H matrix:

- Stores the weights of each code/base.
- Apply a **non-negative linear combination** of the bases.
  - On the right- it is a reshaped vector from matrix H, as it has S=7x7=49 pixels.
- The dimensions of H are (SxN).
- The weight vector is sparse- only applies weights to certain features.

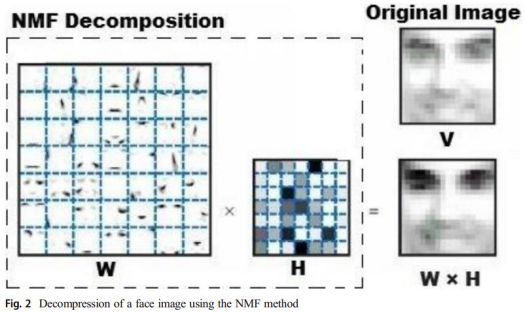


Fig. 2 Decompression of a face image using the NMF method

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

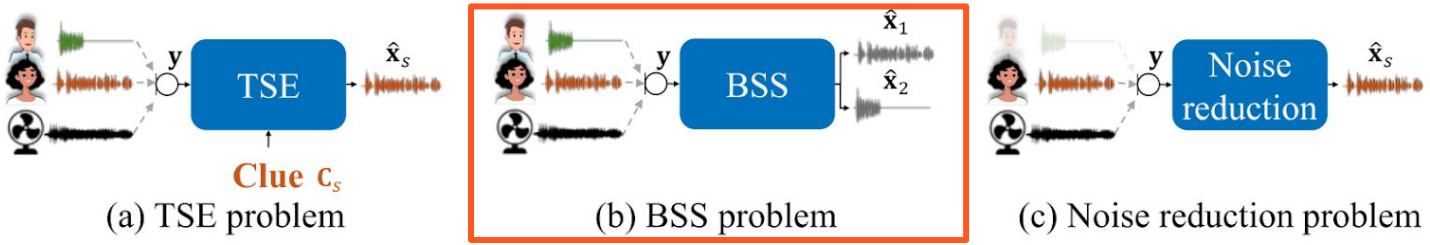
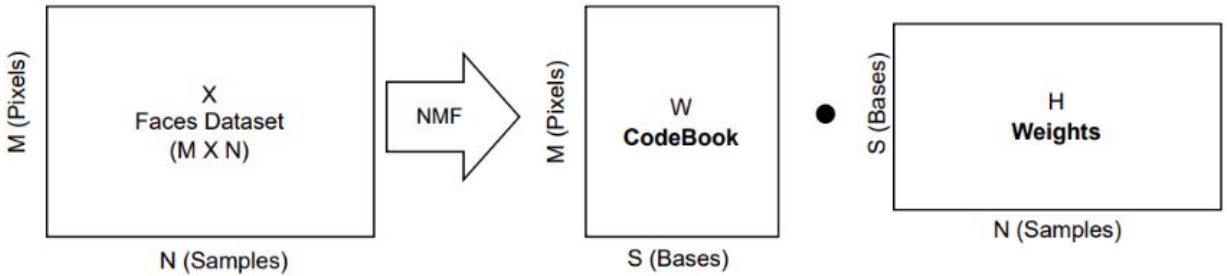


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Reconstruction

To reconstruct the i'th face, we need to multiply: $X_i = W \times H_i$. where H_i is a row vector that has weight per base (code). This returns the reconstructed image.



Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

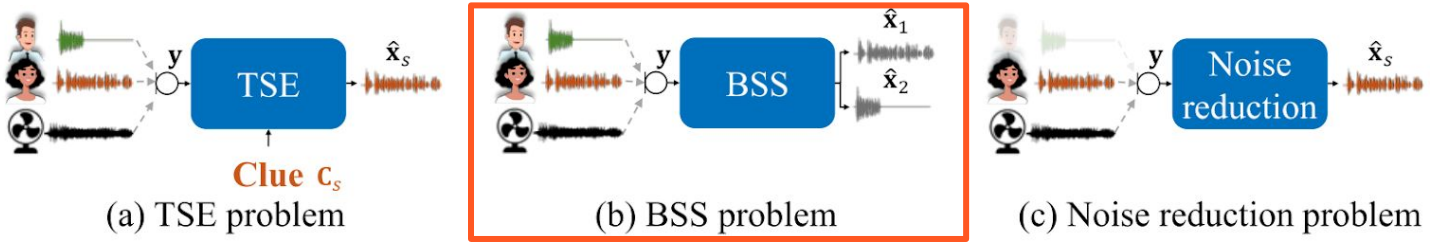


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Finding H and W

A minimization problem: minimizing the distance between X and WH, with respect to W and H.

$$W, H = min_{W,H} |X - WH|^2, s.t. W, H \geq 0$$

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

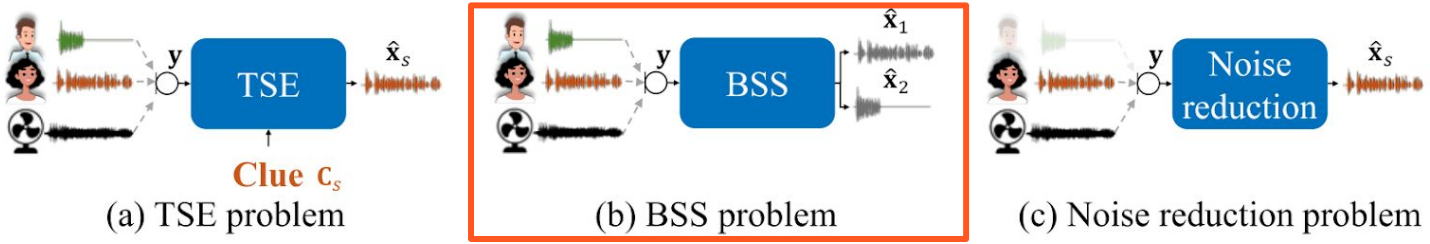


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Finding H and W

There are several approaches to solve this task. The most commonly used is the Multiplicative Update algorithm, an **iterative algorithm**, that was introduced by Lee and Seung in 2000. This often finds a local minimum and not the global one, but it was found to be useful.

1. Randomly initialize non-negative W, H
2. Fix W and update H (element wise multiplication and division):

$$H \leftarrow H \frac{W^T X}{W^T W H}$$

3. Fix H and update W (element wise multiplication and division):

$$W \leftarrow W \frac{W^T X}{W H H^T}$$

4. Calculator the distance from X, and continue step 2 until convergence.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

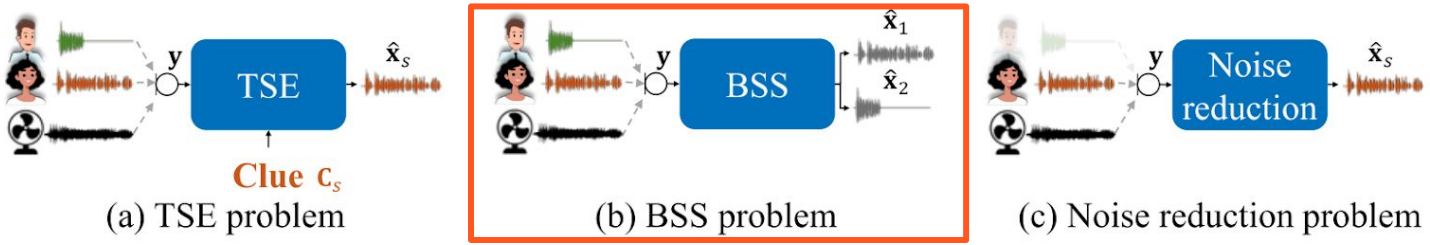


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Audio Application

- Given a mixture of sources in an audio we would like to decompose it to its sources.
- The input audio is first transformed using STFT to the frequency domain
  - Only the magnitude is manipulated- **it's non-negative.**
  - The magnitude dimensions are FxT, where F represents the frequency and T the temporal axis.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

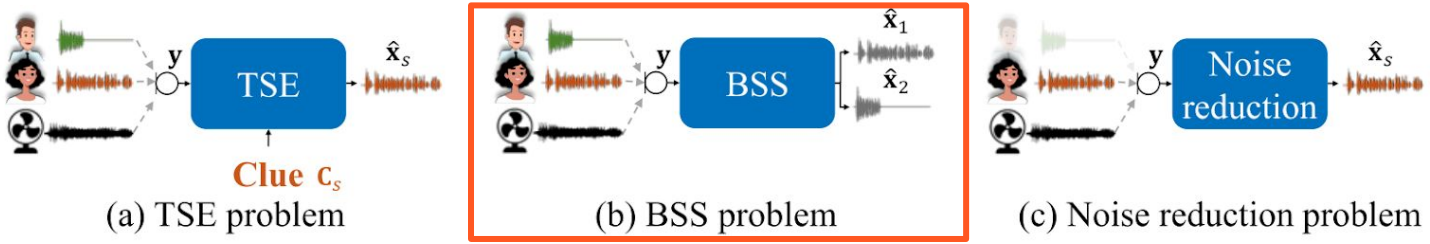


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Audio Application

- We assume that there is an underlying (hidden) structure to the data, and NMF constructs sparse bases (clusters/codes) (W matrix) and weighting (H matrix).
  - W and H have dimension of FxS and SxT, respectively.
- The non-negativity implies that only additive combinations of the bases are allowed.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

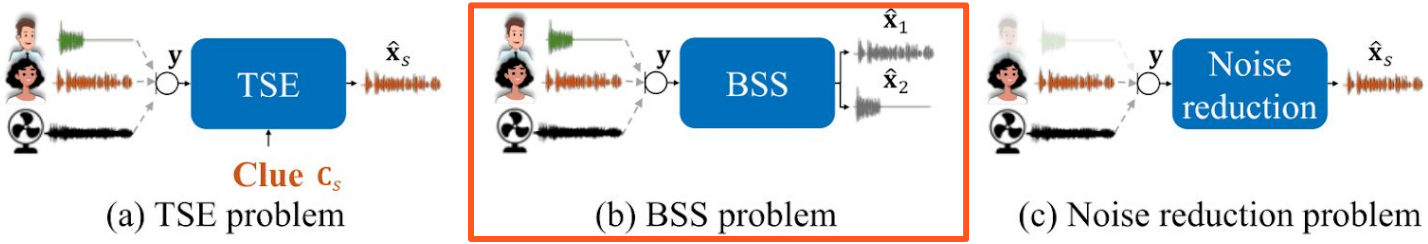
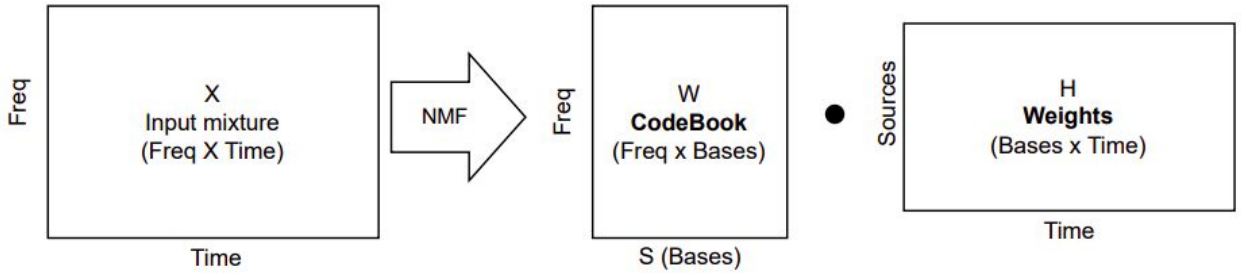


Fig. 2. Comparison of TSE with BSS and noise reduction

Non-Negative Matrix Factorization (NMF) - Audio Application



*This method can be viewed as an **unsupervised clustering** that operates on mono.

Source 1, 2

Speech Enhancement- BSS- Methods

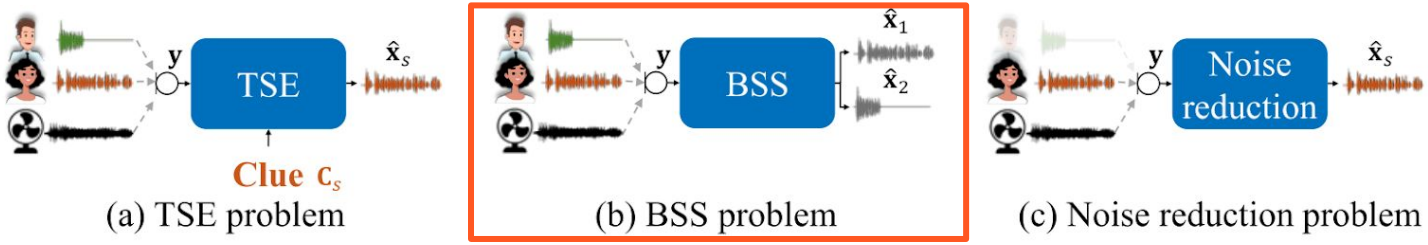


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Traditional:**
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - **Multi mic: Independent component analysis (ICA)**
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- Trad.

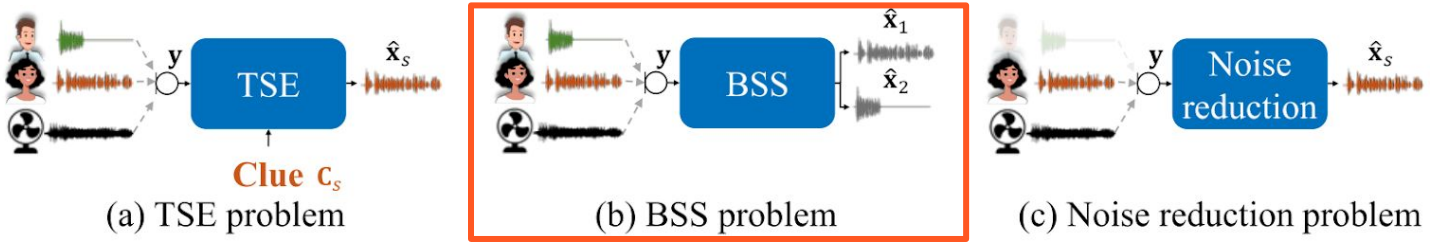


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Introduction

Imagine two people talking simultaneously during a cocktail party

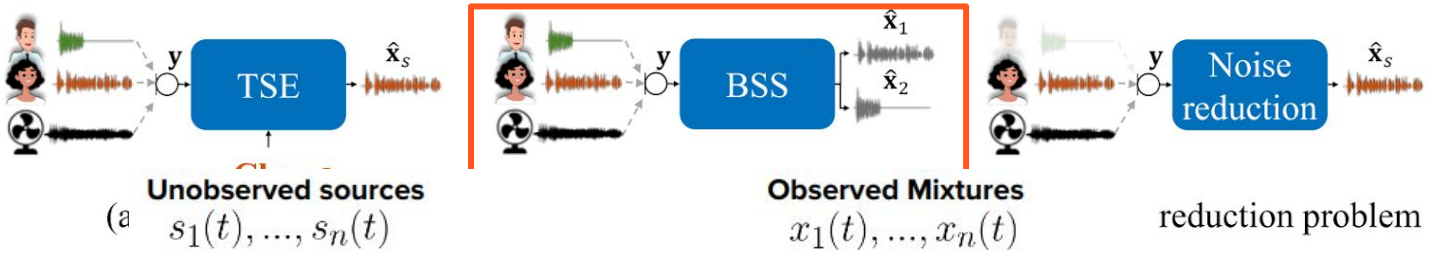
- Two microphones placed near both speakers.
- Both voices are heard by both microphones at different volumes based on the distance between the person and the microphone.

In other words, we record two files that include audio from the two speakers mixed together.

- How can we separate the two voices in each file to obtain isolated recordings of each speaker?

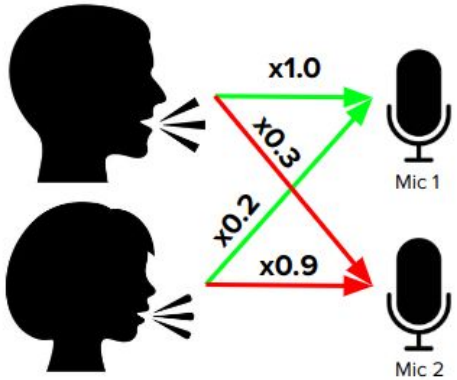
Source 1

Speech Enhancement- BSS- Methods- Trad.



(a reduction problem

Fig. 2. Comparison



$x_1(t) = 1.0 \cdot s_1(t) + 0.2 \cdot s_2(t)$

$x_2(t) = 0.3 \cdot s_1(t) + 0.9 \cdot s_2(t)$

Independent com

Imagine two people

- Two microphc
- Both voices a ed on the distance between the p

In other words, we record two files that include audio from the two speakers mixed together.

- How can we separate the two voices in each file to obtain isolated recordings of each speaker?

Source 1, 2, 3, 4, 5

Speech Enhancement- BSS- Methods- Trad.

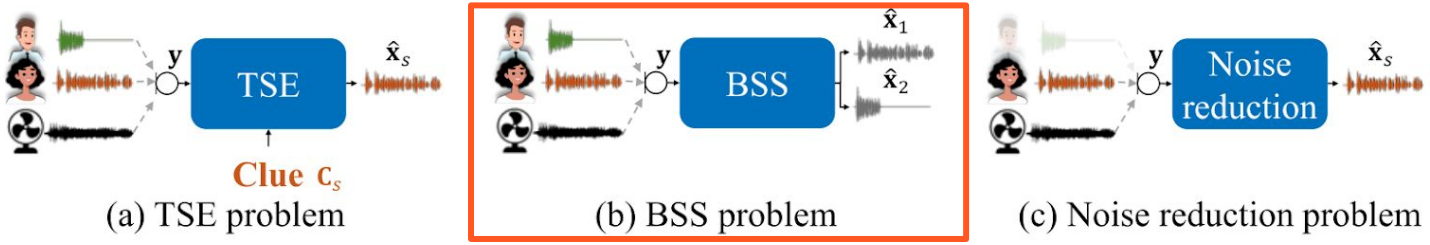


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Introduction

ICA consists in recovering unobserved (latent) signals or 'sources' from several observed mixtures. Typically, the observations are obtained at the output of a set of sensors, each sensor receiving a different combination of the 'source signals'.

Source 1

Speech Enhancement- BSS- Methods- Trad.

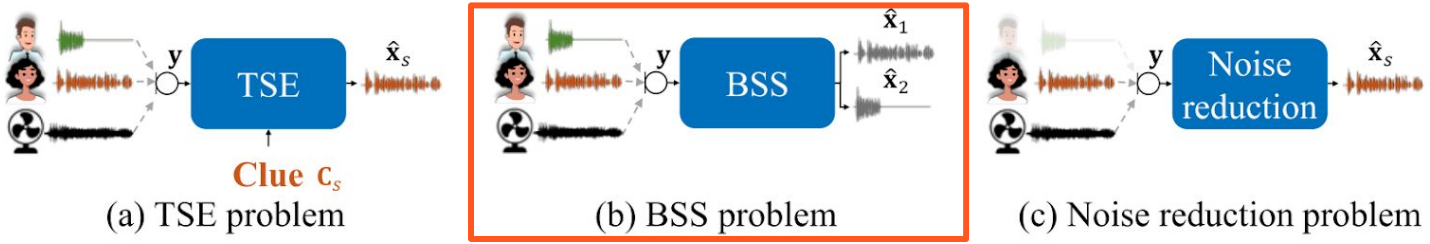
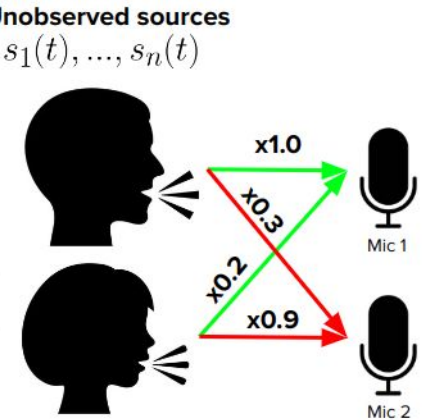
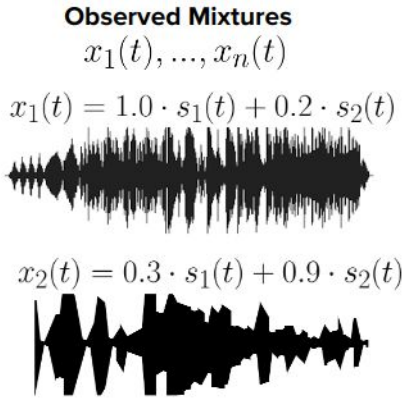


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Introdu

- The (**unobserved**) **sources** are the male and female speakers.
- The **observed mixtures** are the audio recorded by the microphones 1 & 2. Each microphone receives a different combination of the source audio.





Source 1, 2, 3, 4, 5

Speech Enhancement- BSS- Methods- Trad.

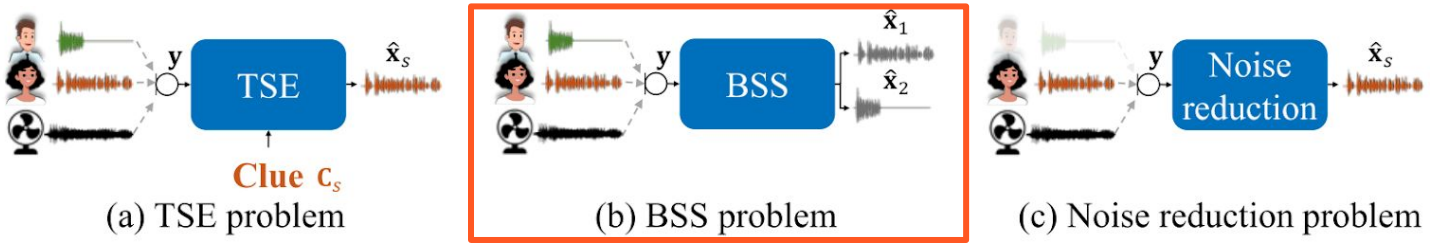
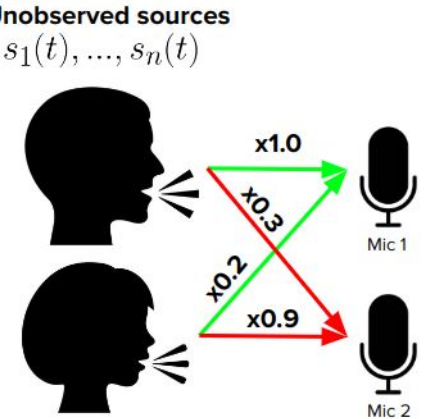
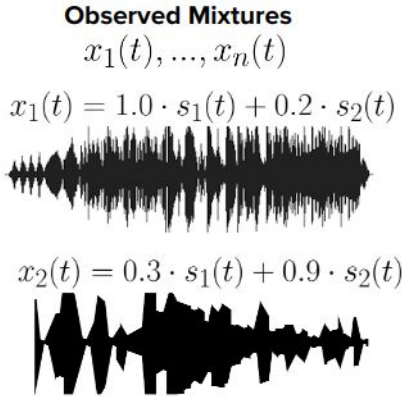


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Introdu

The lack of a priori knowledge about the mixture is compensated by a statistically strong but often physically plausible assumption of **independence between the source signals**.





Source 1, 2, 3, 4, 5

Speech Enhancement- BSS- Methods- Trad.

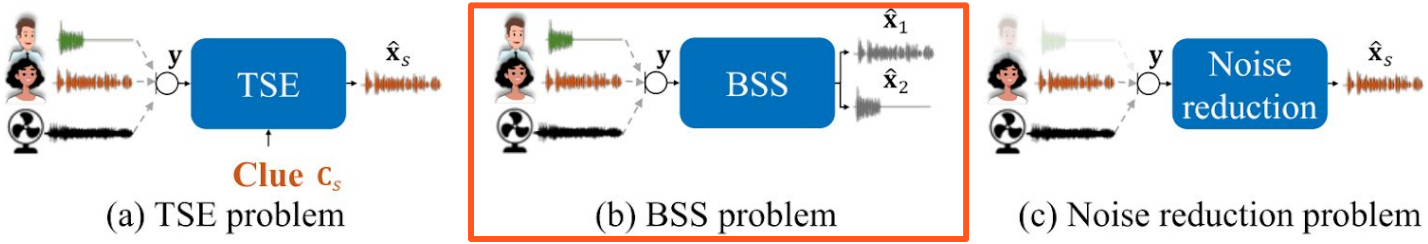


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Introduction

Original

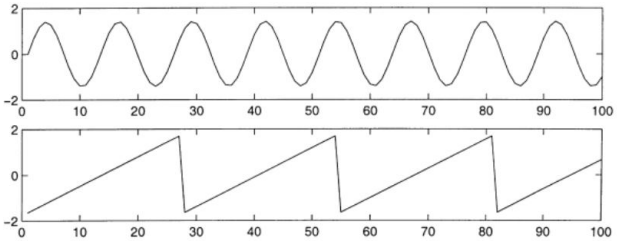


Fig. 1. The original signals.

Observed

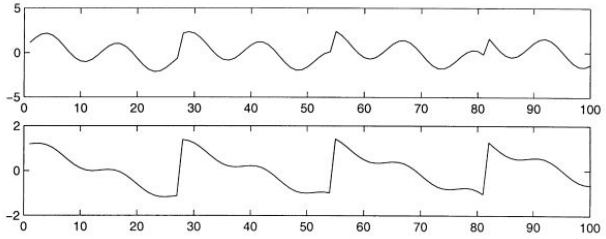
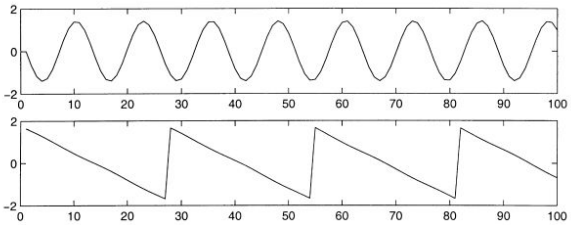


Fig. 2. The observed mixtures of the source signals in Fig. 1.

Estimated

(ambiguity in scale, sign, and perm.)



of the original source signals, estimated using only the observed signals in Fig. 2. The original signals were very accu
ıs.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

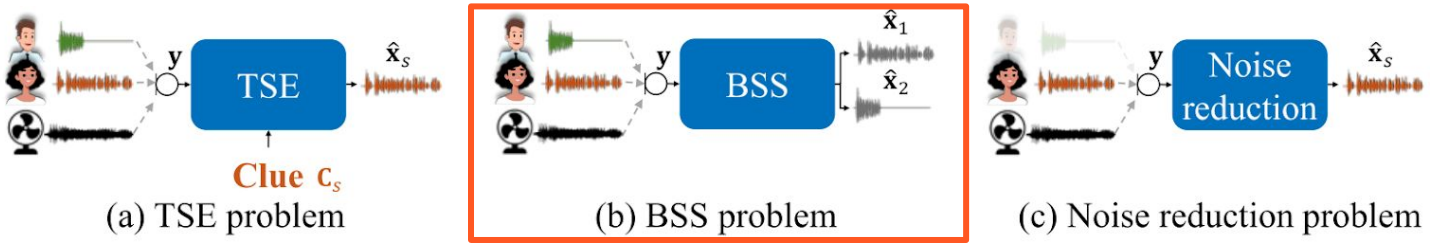


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

The simplest BSS model assumes the existence of n independent signals $s_1(t)$, ..., $s_n(t)$ and the observation of as many mixtures $x_1(t)$, ..., $x_n(t)$, these mixtures being linear, i.e.

$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t)$ for each i = 1, n.

This is compactly represented by the mixing equation

$$\mathbf{x}(\mathbf{t}) = A\mathbf{s}(t)$$

Where: $\mathbf{s}(\mathbf{t}) = [s_1(t), ..., s_n(t)]^T$ is an n × 1 column vector collecting the source signals, vector $\mathbf{x}(\mathbf{t}) = [x_1(t), ..., x_n(t)]^T$ similarly collects the *n* observed signals and the square *n*×*n* 'mixing matrix' *A* contains the mixture coefficients.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

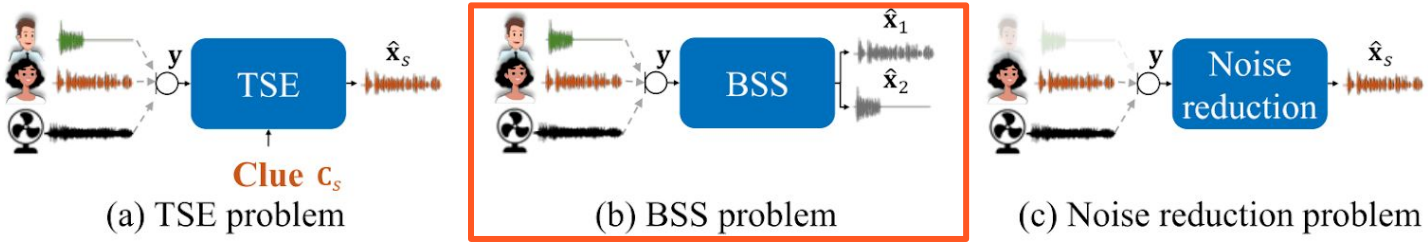


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

The basic assumption of ICA is that each mixture $x_j(t)$ as well as each independent component $s_k(t)$ is a random variable, instead of a proper time signal.

- Both the mixture variables and the independent components (sources) have zero mean
  - If it's not the case, we subtract the sample mean

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

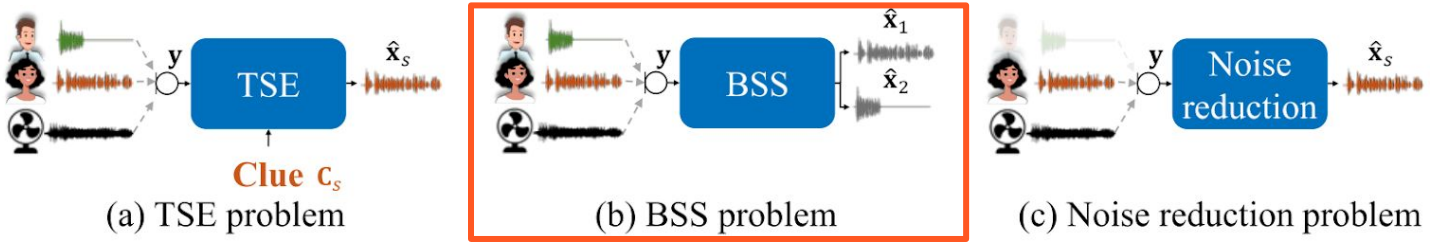


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

The BSS problem consists in recovering the source vector ***s(t)*** using only the observed data ***x(t)***. It can be formulated as the computation of an $n \times n$ 'separating matrix' B whose output

$\mathbf{y}(\mathbf{t}) = B\mathbf{x}(t)$ is an estimate of the vector ***s(t)*** of the source signals.

$$\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} = \mathbf{s} \rightarrow \boxed{A} \xrightarrow{\mathbf{x}} \boxed{B} \rightarrow \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \hat{\mathbf{s}}$$

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

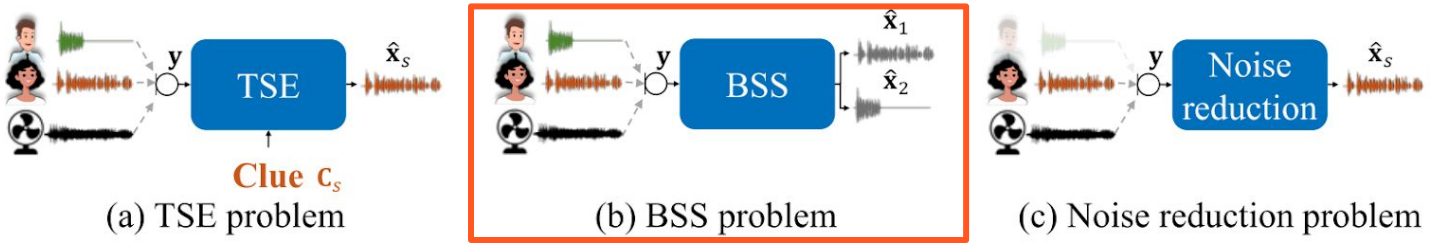


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

- Given the mixing equation: x=As
- Assumptions:
  - the components s_i(t) are statistically independent.
  - The independent component must have non-Gaussian distributions.
  - We do not assume these distributions are known.
- We also assume that the unknown mixing matrix (A) is square- can be sometimes relaxed.
- After estimating the matrix A, we can compute its inverse B, and obtain the independent component simply by: s=Bx

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

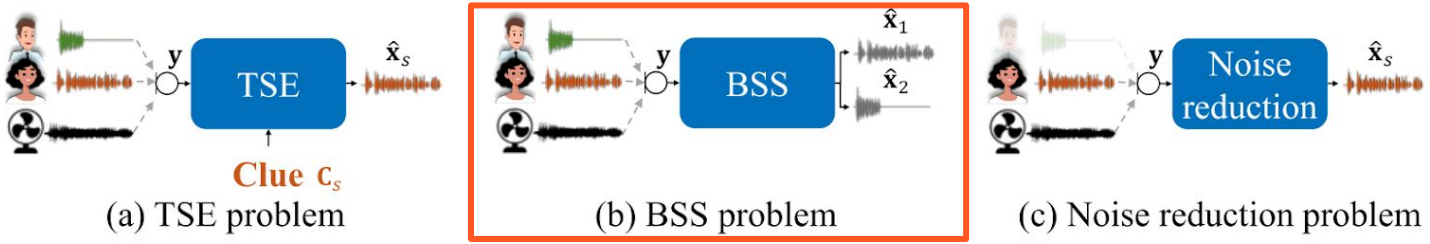


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

- Giv
- Ass
  - $\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} = \mathbf{s} \rightarrow A \xrightarrow{\mathbf{x}} B \rightarrow \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \hat{\mathbf{s}}$
  -
  -
- We also assume that the unknown mixing matrix (A) is square- can be sometimes relaxed.
- After estimating the matrix A, we can compute its inverse B, and obtain the independent component simply by: s=Bx

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

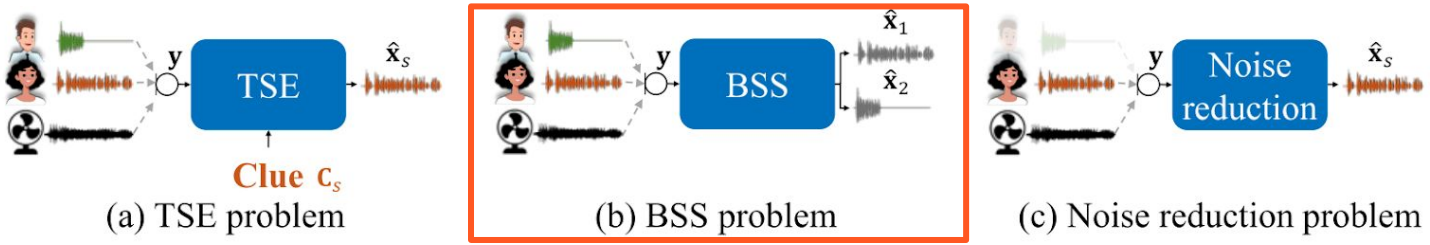


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Problem formulation

The mixing equation (x=As) implies that the following ambiguities will hold:

- We cannot determine the **variances (energies)** of the independent components.
  - Both s and A are unknown ➔ any scalar multiplier in one of the sources si(t) could always be canceled by dividing the corresponding column ai of A by the same scalar
- The **order** of the independent components cannot be determined.
  - The reason is that we can freely change the order of the terms in the sum in x=As.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

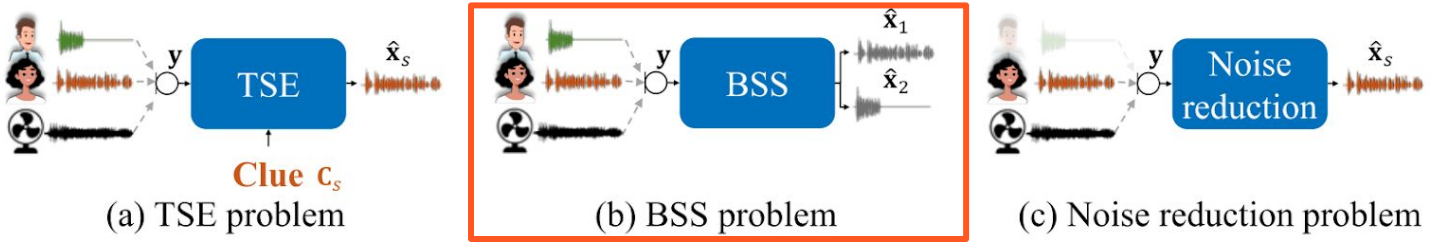


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

consider two independent components (*S*) that have the following uniform distributions:

- This uniform distribution has zero mean and variance equal to one.
- The joint density of s1 and s2 is then uniform on a square

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

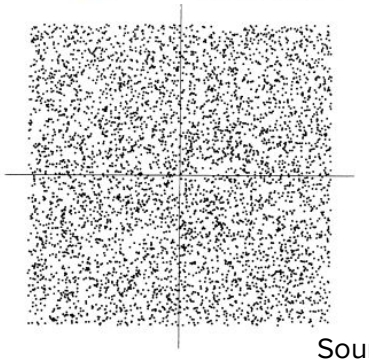


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

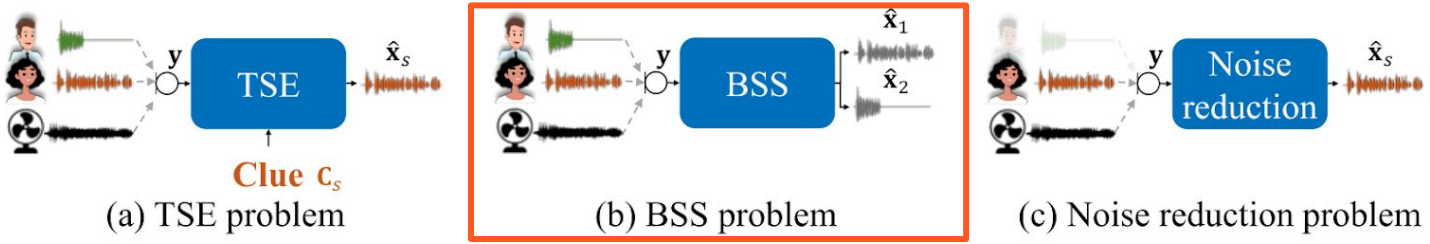


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

- The joint density of the two independent variables is (by definition) the product of their marginal densities:

$$p(s_1, s_2) = p_1(s_1)p_2(s_2)$$

- Using the following mixing matrix we will mix the independent components

$$\mathbf{A}_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \le \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

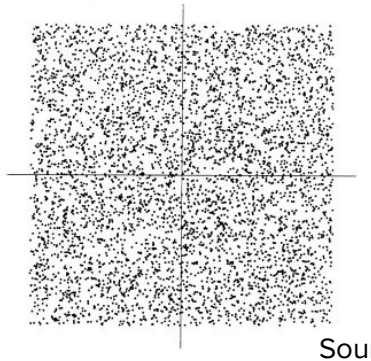


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

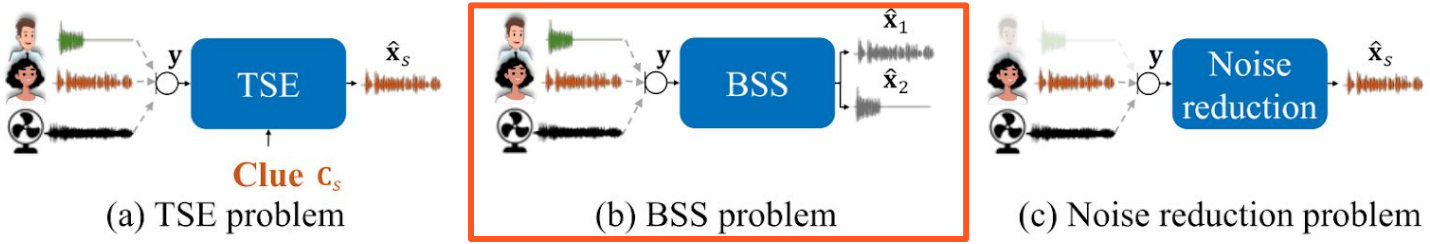


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

- This gives us two mixed variables, ***x1* and *x2***.
- It is easily computed that the mixed data has a uniform distribution on a parallelogram.

$$\mathbf{A}_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

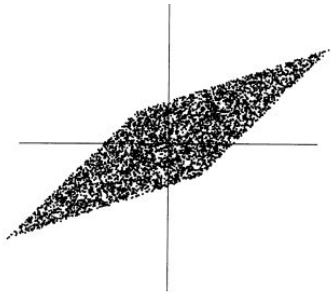


Fig. 6. The joint distribution of the observed mixtures x_1 and x_2. Horizontal axis: x_1, vertical axis: x_2.

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \le \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

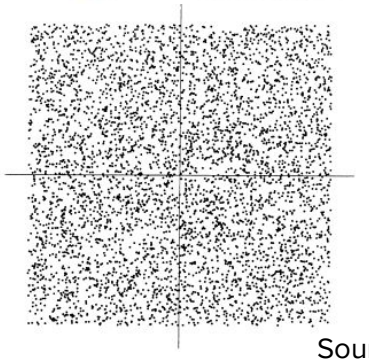


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

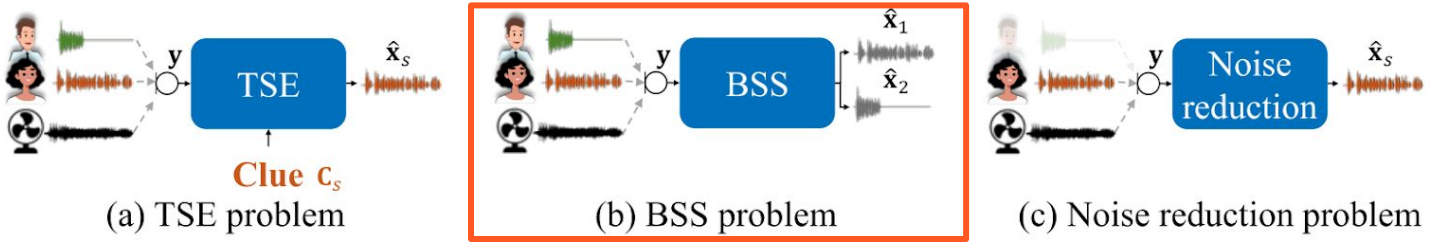


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

- random variables ***x1*** **and** ***x2*** are **not independent** anymore (unlike *s1* and *s2*).
- Clearly if x1 attains one of its maximum or minimum values, then this completely determines the value of x2.

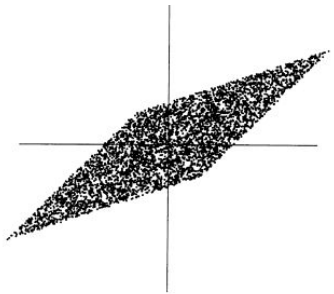


Fig. 6. The joint distribution of the observed mixtures x_1 and x_2. Horizontal axis: x_1, vertical axis: x_2.

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \le \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

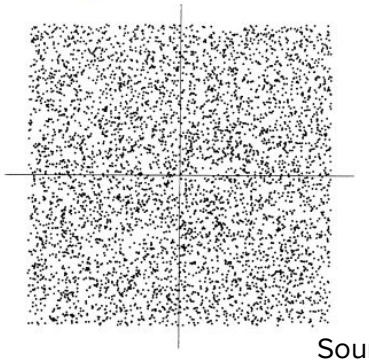


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

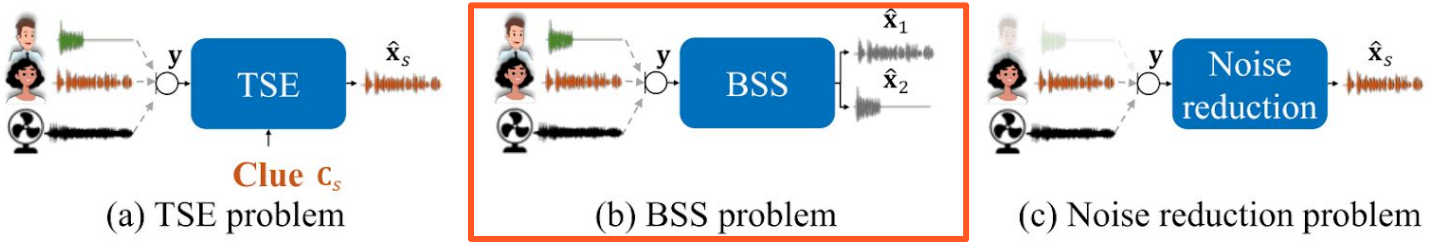


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

- random variables ***x1*** **and** ***x2*** are **not independent** anymore (unlike

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \le \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

ICA goal is to estimate the mixing matrix A using only information contained in the mixtures *x1* and *x2*.

maximum or minimum values, then this completely determines the value of x2.

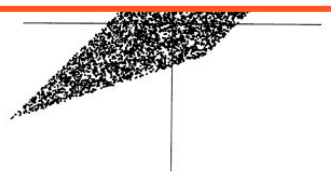


Fig. 6. The joint distribution of the observed mixtures x_1 and x_2. Horizontal axis: x_1, vertical axis: x_2.

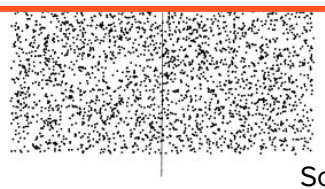


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

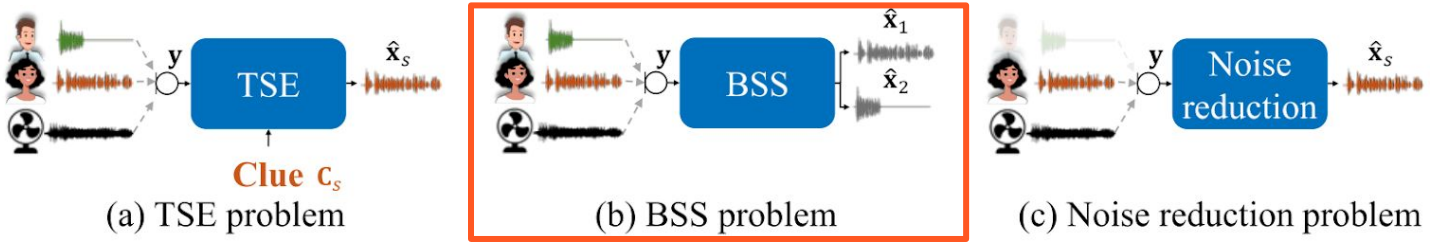


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Illustration & Intuition

- intuitive way of estimating A: the edges of the parallelogram are in the directions of the columns of A.
- In reality, it has limited results as it requires the variables to have exactly uniform distributions, and is computationally complicated to implement.

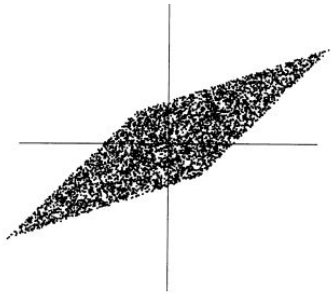


Fig. 6. The joint distribution of the observed mixtures x_1 and x_2. Horizontal axis: x_1, vertical axis: x_2.

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

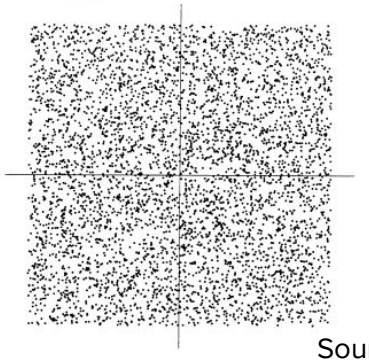


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

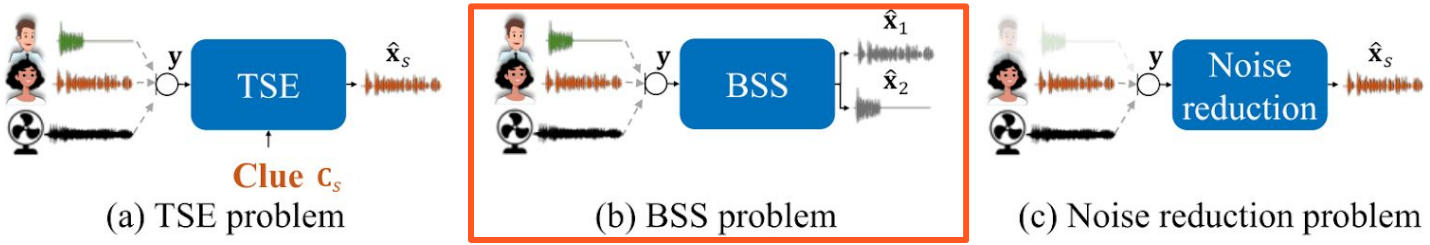


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Method

- The approach of ICA is essentially 'spatial': looking for **structure across the sensors, not across time**.
- We would like to identify the probability distribution of a vector **x** = A**s** given a sample distribution (***x***).
- In this perspective, the statistical model has **two components**:
  - The **mixing matrix A**
  - The **probability distribution** of the source vector **s**.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

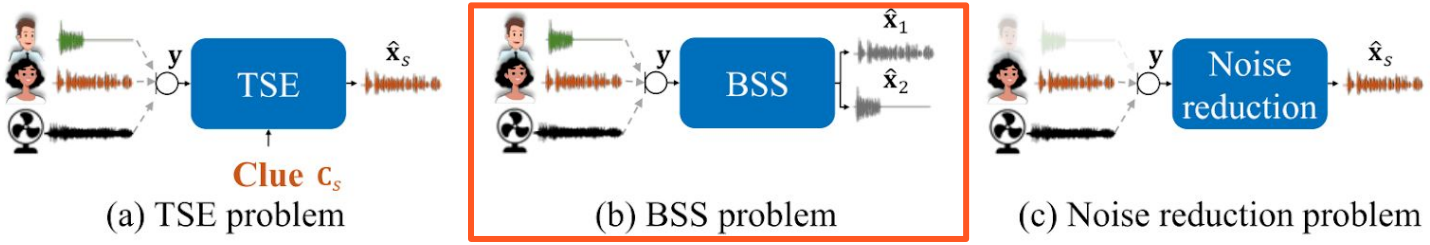


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Non-Gaussian assumption

- The fundamental restriction in ICA is that the independent components (*si*) must be **non-Gaussian** for ICA to be possible
- If *si* were Gaussian and the mixing matrix is orthogonal, then x1(t) and x2(t) are also Gaussian, uncorrelated, and of unit variance.
- Their joint density is given by

$$p(x_1, x_2) = \frac{1}{2\pi} exp(-\frac{x_1^2+x_2^2}{2})$$

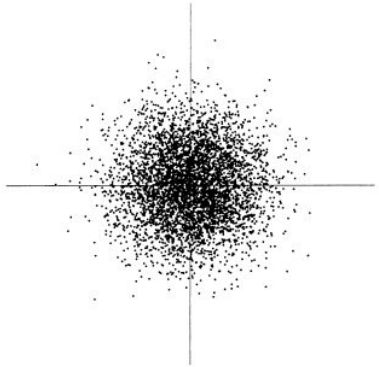


Fig. 7. The multivariate distribution of two independent Gaussian variables.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

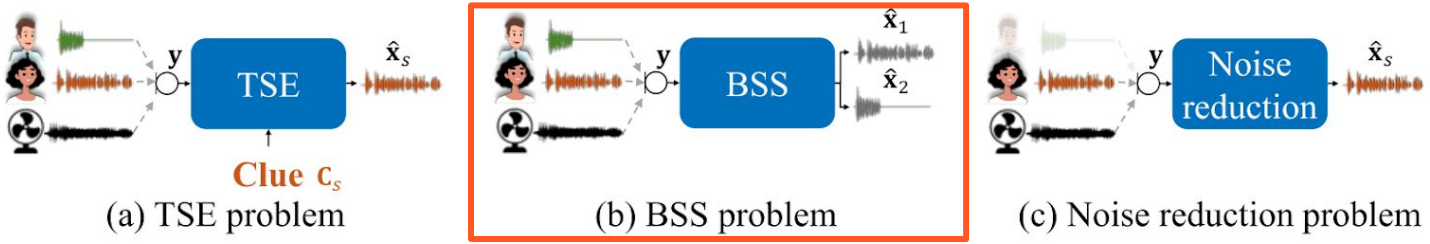


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Non-Gaussian assumption

- If si were Gaussian:
  - The joint density of x is **symmetric**.
  - Therefore, it **does not contain any information on the directions of the columns of the mixing matrix A**.
- This is why A cannot be estimated.

$$p(x_1, x_2) = \frac{1}{2\pi} exp(-\frac{x_1^2+x_2^2}{2})$$

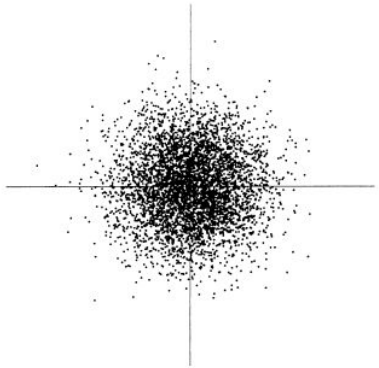


Fig. 7. The multivariate distribution of two independent Gaussian variables.

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

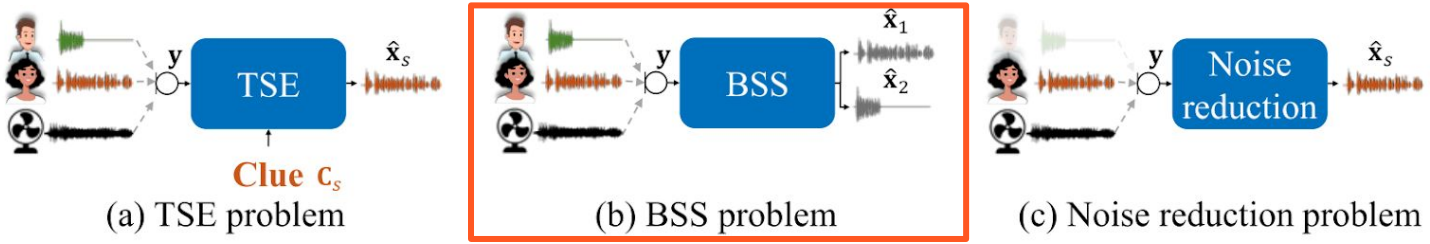


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Non-Gaussian assumption

From the Central Limit Theorem, the sum of two independent random variables has a distribution that is closer to Gaussian than either of the original variables.

- ICA combines this idea, non-Gaussianity measures, and the non-Gaussian assumption to uncover independent components hidden in data.
- This allows us to frame ICA as an optimization problem

$$B^* = max_B\, kurt(BX)$$

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

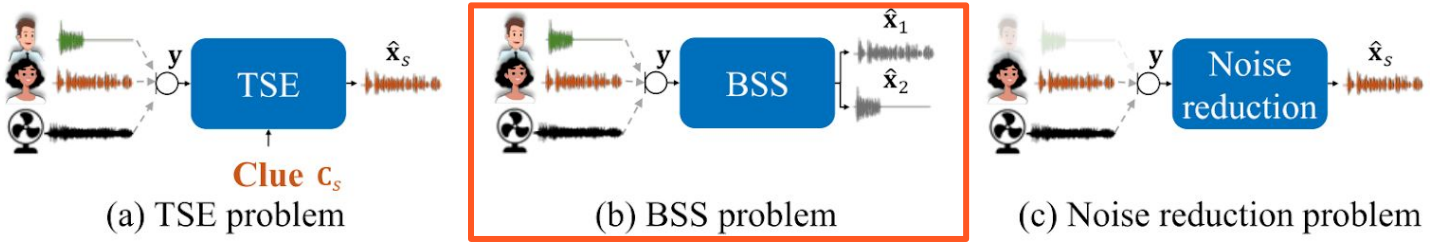
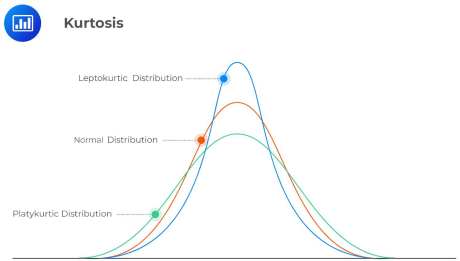


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Non-Gaussian assumption

- Non-Gaussianity quantifies how far the distribution of a random variable is from being Gaussian. Example measures of non-Gaussianity are **kurtosis** (4th moment) and **negentropy**.

$$\hat{\kappa}(x) = \frac{1}{N}\sum_{i=1}^{M}\left(\frac{x_i - \mu_x}{\sigma}\right)^4$$



$$B^* = max_B\, kurt(BX)$$

Source 1, 2 ,3

Speech Enhancement- BSS- Methods- Trad.

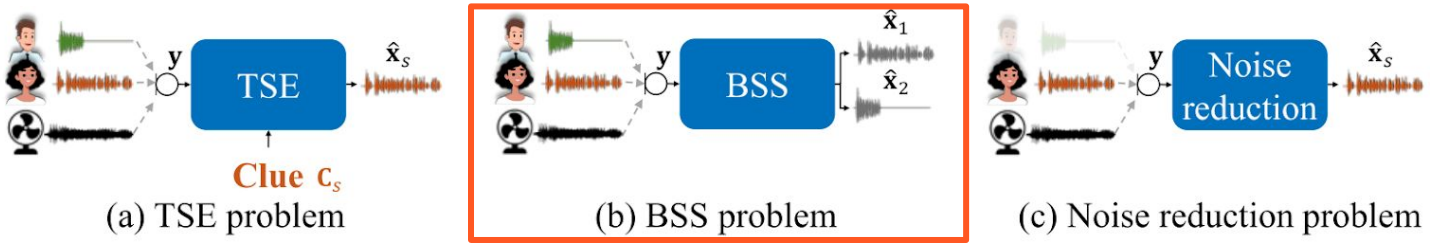


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Non-Gaussian assumption

- Non-Gaussianity quantifies how far the distribution of a random variable is from being Gaussian. Example measures of non-Gaussianity are **kurtosis** (4th moment) and **negentropy**.
  - kurtosis is zero for a Gaussian random variable. For most (but not quite all) non-Gaussian random variables, kurtosis is non-zero

$$B^* = max_B\, kurt(BX)$$

Source 1, 2 ,3

Speech Enhancement- BSS- Methods- Trad.

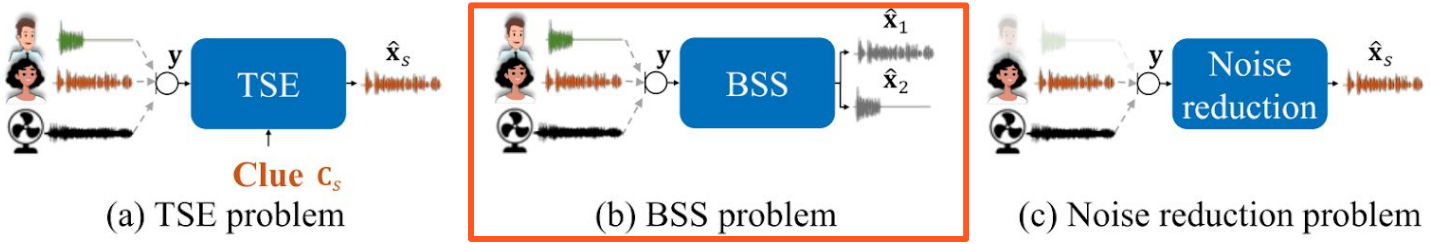


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Method

- ICA combines this idea, non-Gaussianity measures, and the non-Gaussian assumption to uncover independent components hidden in data.
- **This allows us to frame ICA as an optimization problem**
- ICA finds the de-mixing matrix B that **maximizes the non-Gaussianity** of the estimated sources.
- This implies (mathematically) that we will **find the independent components**, as if they were **dependent** their non-Gaussianity score would be **lower**.

$$B^* = max_B\, kurt(BX)$$

Source 1, 2

Speech Enhancement- BSS- Methods- Trad.

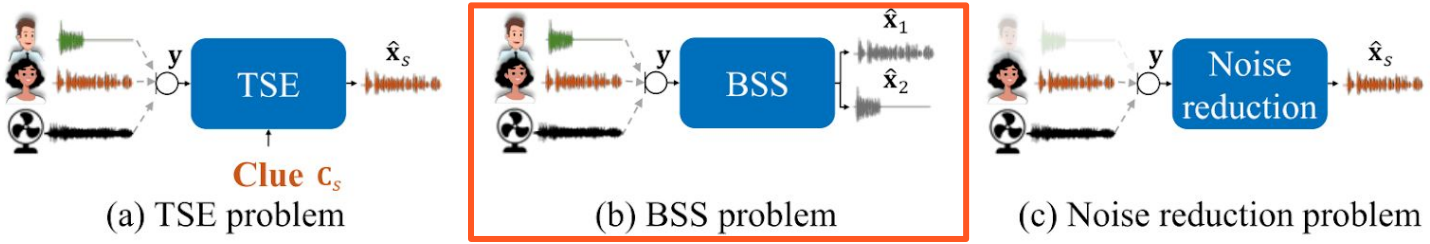


Fig. 2. Comparison of TSE with BSS and noise reduction

Independent component analysis (ICA)- Method

In practice, ICA is an **iterative maximization algorithm** that runs till convergence. The algorithm finds B, then computes *s* by multiplying *B* with the mixture *x*.

- ICA finds the de-mixing matrix B that **maximizes the non-Gaussianity** of the estimated sources.
- This implies (mathematically) that we will **find the independent components**, as if they were **dependent** their non-Gaussianity score would be **lower**.

$$B^* = max_B\, kurt(BX)$$

Source 1, 2

Speech Enhancement- BSS- Methods

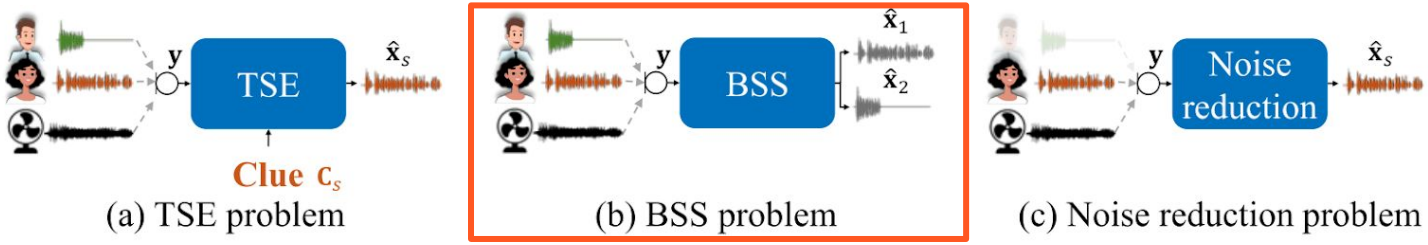


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

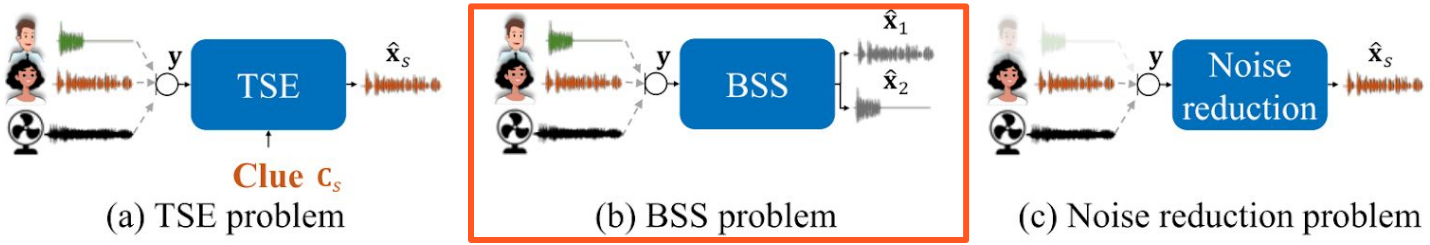


Fig. 2. Comparison of TSE with BSS and noise reduction

- With the development of deep learning, **fully supervised methods** have gained momentum.
- Unlike the **traditional approaches** that were trained in an **unsupervised** manner, **DNNs** are trained in a **supervised** manner.
  - NN approaches will not handle data from domain it wasn't trained on (BSS for speech will not handle music and vice versa).

*All methods described in this section (unless stated differently) process mono-channel.

Source 1

Speech Enhancement- BSS- Methods- NN

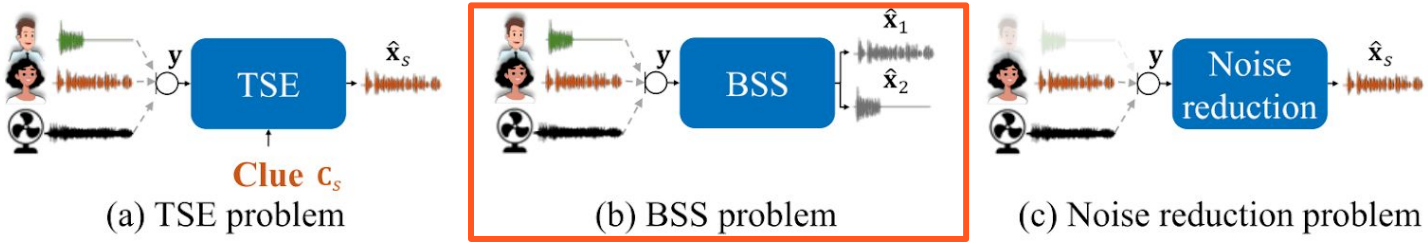


Fig. 2. Comparison of TSE with BSS and noise reduction

Training framework:

Most existing studies use fully supervised training, which requires a large amount of training data consisting of the pairs of speech mixture y, target and each of the sources $\{x_1,\ldots,x_K\}$ to learn parameters θ_{TSE}. Since this requires access to a clean target speech signals, such training data are usually simulated by artificially mixing clean speech signals and noise following the signal model that was introduced in the introduction:

$$\mathbf{y}^m = \mathbf{x}_s^m + \sum_{k \neq s} \mathbf{x}_k^m + \mathbf{v}^m,$$

Source 1

Speech Enhancement- BSS- Methods- NN

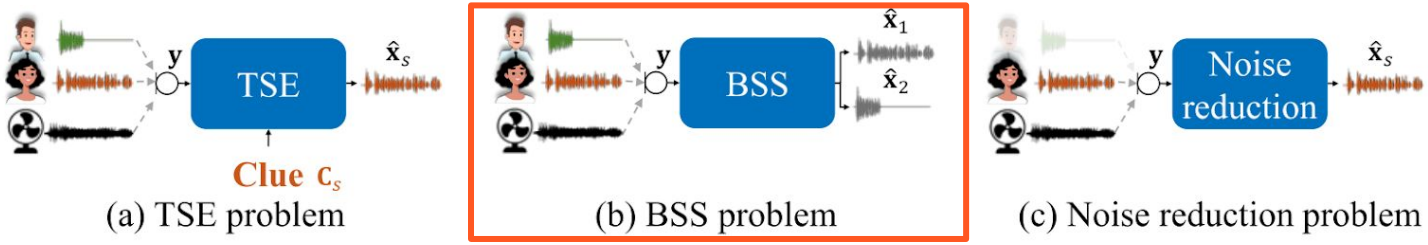


Fig. 2. Comparison of TSE with BSS and noise reduction

Training framework:

The data generation process (dynamic mixing) using a multi-speaker speech corpus:

- Generate a mixture using randomly selected speech signals from the target speaker, the interference speaker, and the background noise.
- Combine them (can add reverb and change the gains to determine desired SNR).

The loss function minimizes the difference between the predicted audio and the ground truth signals.

Source 1

Speech Enhancement- BSS- Methods- NN

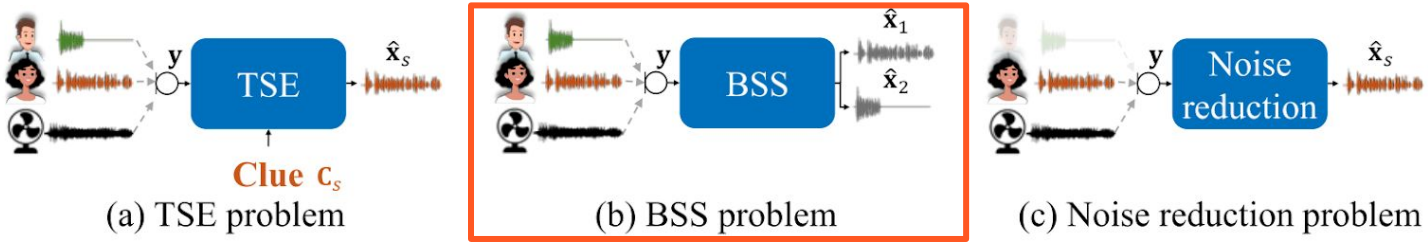


Fig. 2. Comparison of TSE with BSS and noise reduction

Source 1, 2

Speech Enhancement- BSS- Methods

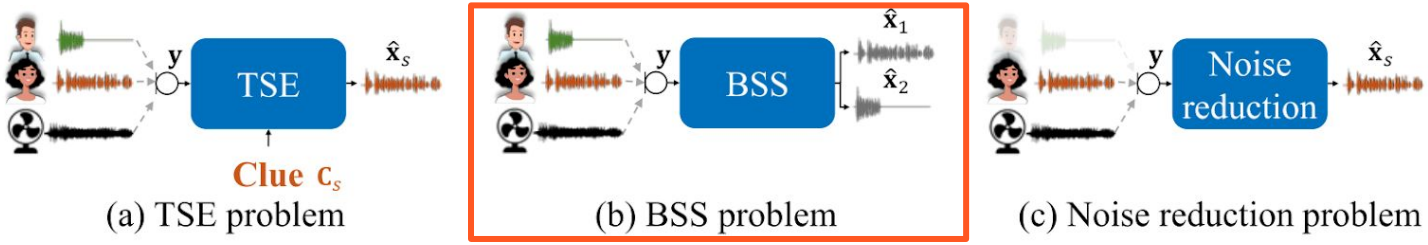


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - **Deep Clustering**
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

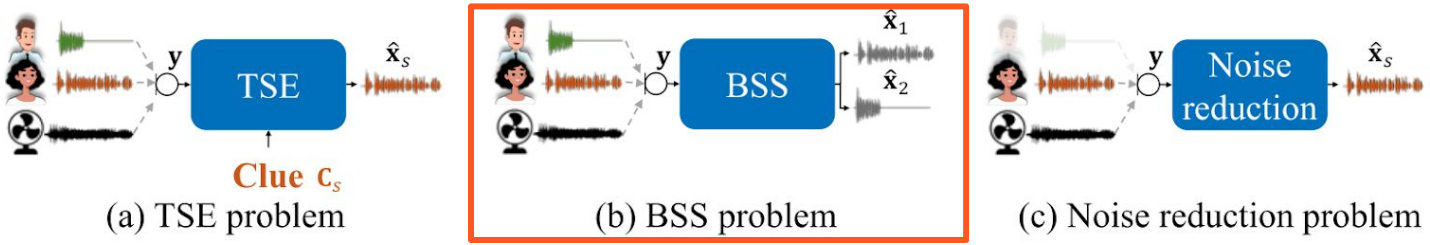


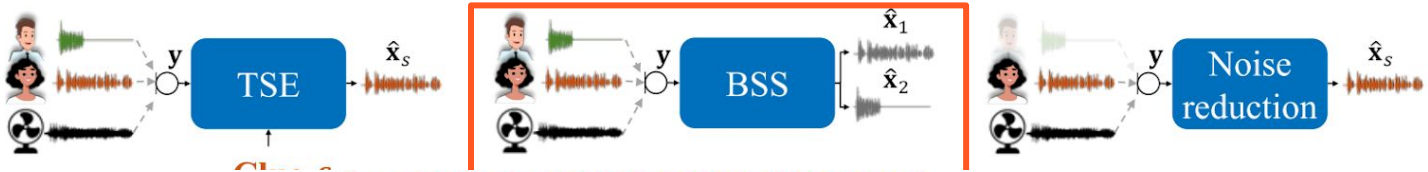
Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering

- Train an embedding based (termed partition-based segmentation) model, instead of a class-based segmentation model.
- The model enables to generalize to unseen classes (voice) during inference.
- Once we have the embeddings, the correct labeling can be determined by **simple clustering methods**.
- Use the segmentation as a **binary** mask

Source 1, 2

Speech Enhancement- BSS- Methods- NN



(a) TSE

Noise reduction problem

Fig. 2. Comparison of TSE

Deep Clustering

- Train an embeddi ation) model, instead of a class-based segme
- The model enables g inference.
- Once we have th be determined by **simple clustering method**
- Use the segmentat

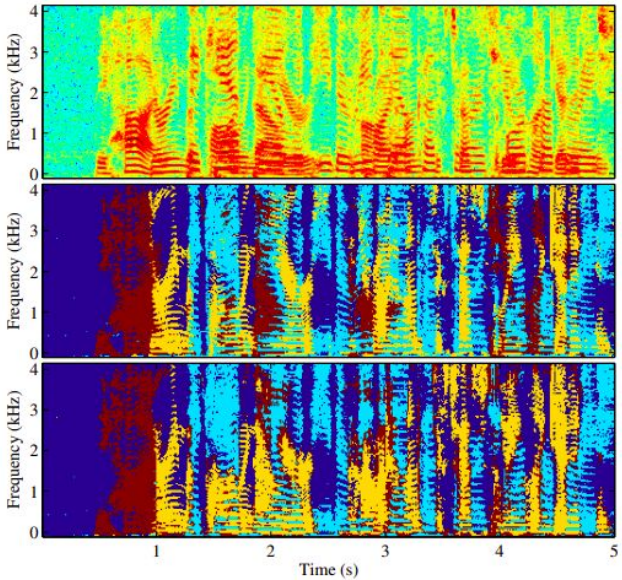


Figure 1: An example of three-speaker separation. Top: log spectrogram of the input mixture. Middle: ideal binary mask for three speakers. The dark blue shows the silence part of the mixture. Bottom: output mask from the proposed system trained on two-speaker mixtures.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

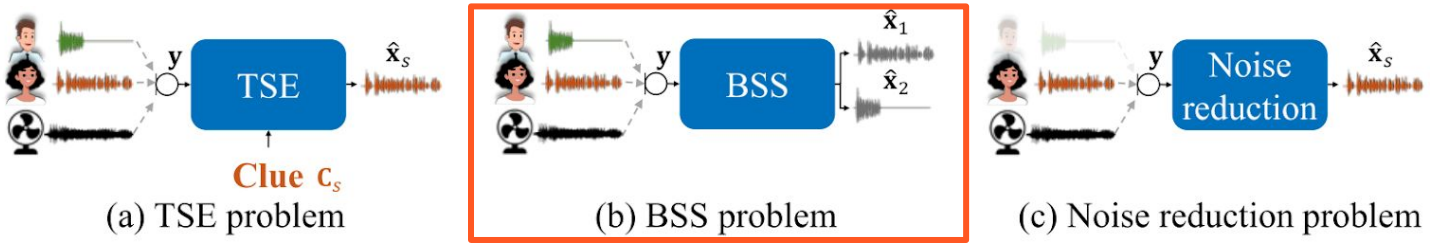


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering

- Let x(t) be the signal in the time-domain, they first apply STFT ➔ resulting in X.
  - For a given time-frequency index n, Xn=Xt,f is the value of the complex spectrogram.
- They assume that there exists a reasonable partition of the elements *n* into regions.
- These regions defined as the sets of **time-frequency bins** in which **each source dominates**
- Estimating such a partition would enable to build time-frequency **masks** to be applied to Xn, leading to time-frequency representations that can be inverted to obtain isolated sources (using iSTFT).

Source 1, 2

Speech Enhancement- BSS- Methods- NN

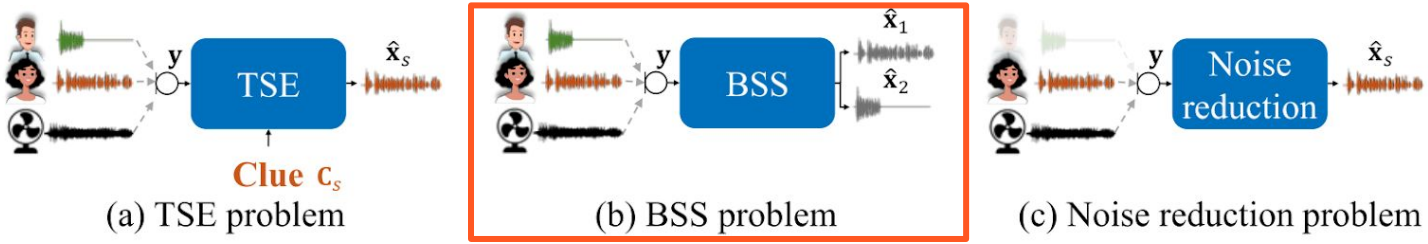


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering

The model f_θ predicts an embedding vector $v_n \in R^K$ for each entry n in the STFT (magnitude).

Resulting in a matrix $V = f_\theta(x) \in R^{NxK}$.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

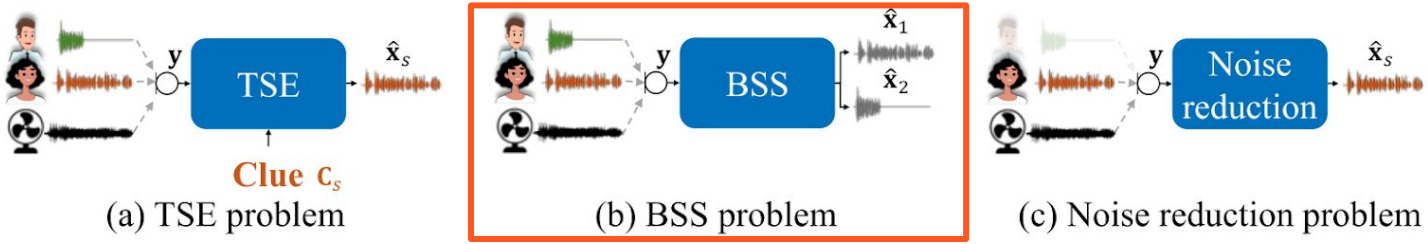


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering

The model f_θ predicts an embedding vector $v_n \in R^K$ for each entry n in the STFT (magnitude).

Resulting in a matrix $V = f_\theta(x) \in R^{NxK}$.

For each the the N elements (STFT entries) they have a reference label indicator $Y = \{y_{n,c}\}$, mapping each element n to each of c arbitrary partition classes, so that $y_{n,c} = 1$ if element n is in partition c, otherwise $y_{n,c} = 0$

Source 1, 2

Speech Enhancement- BSS- Methods- NN

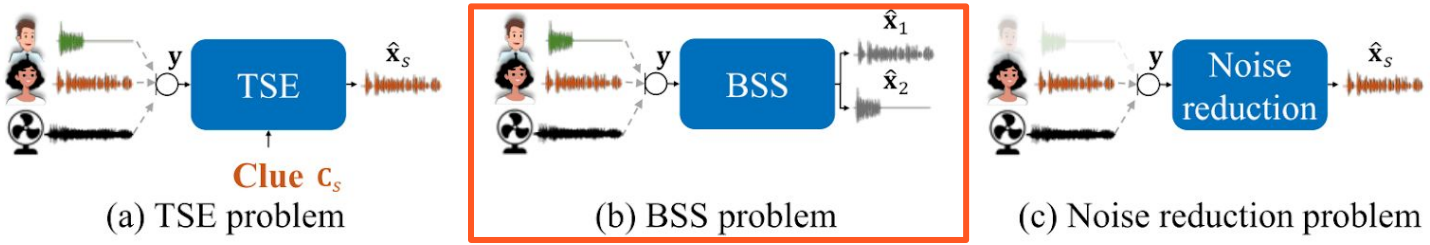


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering - Loss

- The loss tries to optimize the partition of the *N* vectors to be as close as possible to the target partition.
- Their objective pushes the inner product <vi,vj> to 1 when i and j are in the same partition, and to 0 when they are in different partitions.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

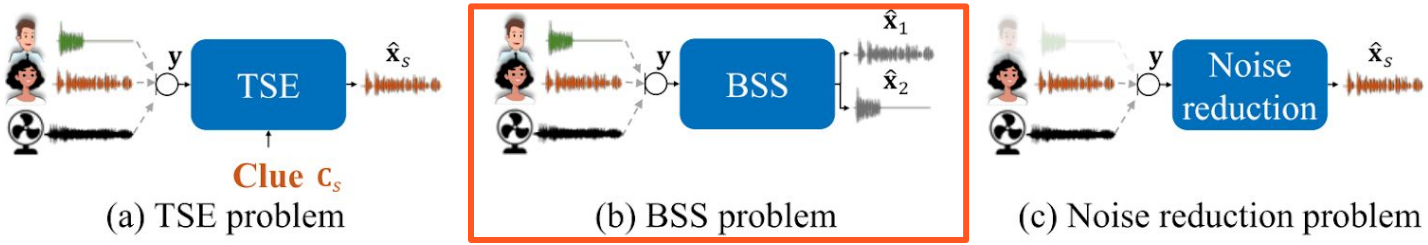


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering - Loss

$$C(\theta) = |V^T V - Y^T Y|^2$$

- $Y \in R^{NxC}$ - $N = T * F$, meaning that for each entry in the STFT matrix we have a one hot vector $y_n \in R^C$ that is equal to 1 if the entry assigned to the cluster *c*, and 0 otherwise.
- $Y^T Y \in R^{NxN}$ - it is a 1/0 metric that defines relations between entries *i,j (*different t,f bins in the STFT matrix). If they are of the same cluster (meaning that $y_{i,c'} = y_{j,c'}$ for every *c')*, the dot product $< y_i, y_j >= 1$. **It does not state to which cluster they belong, just that they belong to the same cluster.**
- $V \in R^{NxK}$, where *K* is the embedding dimension.
- $V^T V \in R^{NxN}$, where each entry is the similarity between predicted embeddings.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

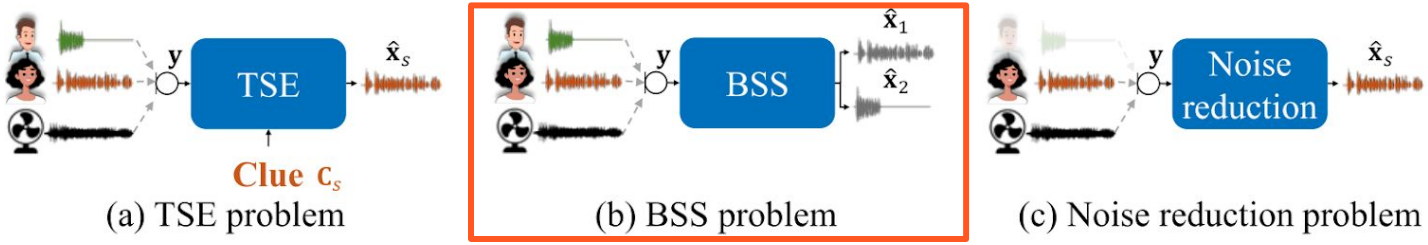


Fig. 2. Comparison of TSE with BSS and noise reduction

Deep Clustering - Inference

- At test time, they compute the embeddings V on the test signal, and cluster the rows vi, for example using k-means.
- Each cluster is then reconstructed back to audio using iSTFT
- Their model was trained on two speakers, but enables to generalize to >2 speakers (although might be out of domain because not seen during training)

Source 1, 2

Speech Enhancement- BSS- Methods

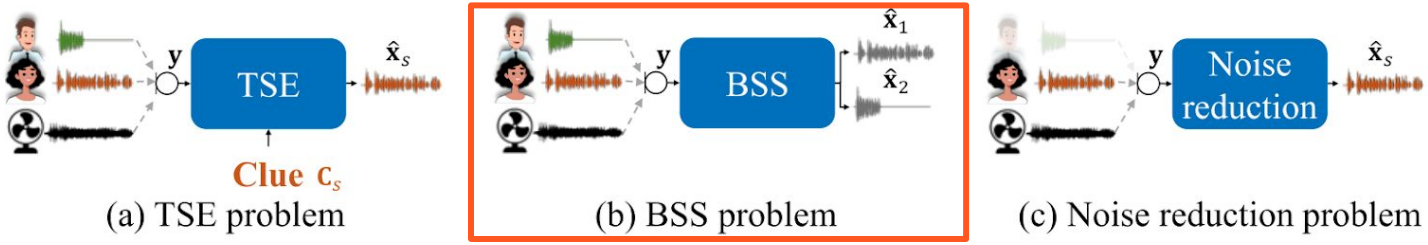


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - **Permutation Invariant loss**
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

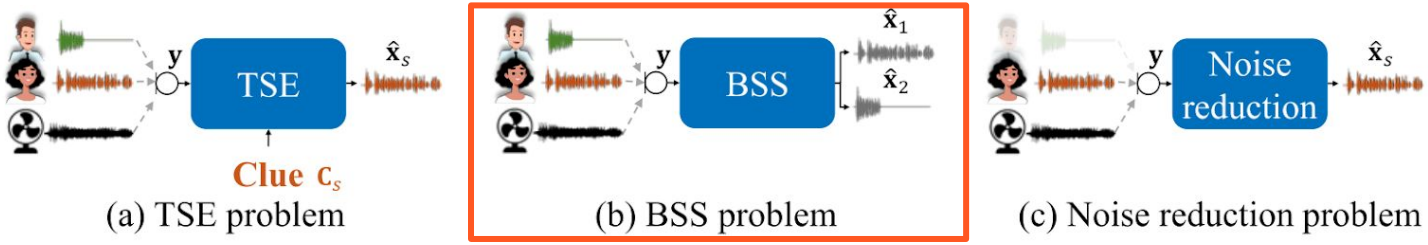


Fig. 2. Comparison of TSE with BSS and noise reduction

Permutation Invariant Loss (PIT)

- Between 2015-2018 people tried to find what is the best way to phrase the permutation invariant for speech.
- Dong Yu et. al. suggested permutation invariant training (PIT).
- This strategy effectively solves the long lasting label permutation problem that has prevented progress on deep learning based techniques for speech separation.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

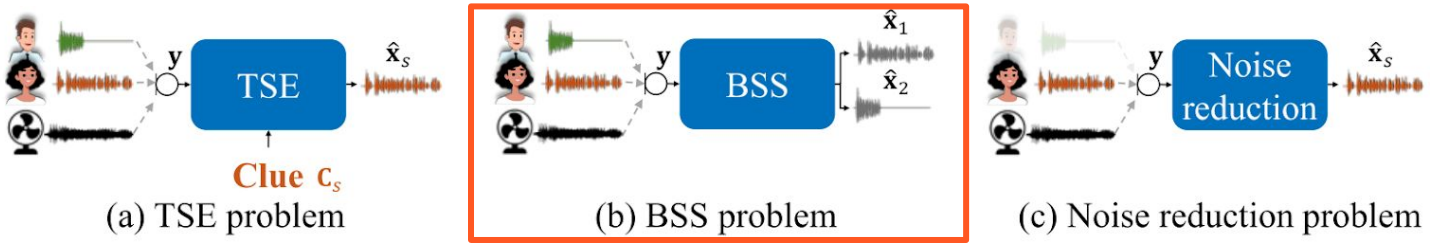


Fig. 2. Comparison of TSE with BSS and noise reduction

Permutation Invariant Loss (PIT)

- To associate references to the output layers:
  - Determine the (total number of S!) possible assignments between the references and the estimated sources.
  - Compute the total MSE for each assignment, which is defined as the combined pairwise MSE between each reference |Xs| and the estimated source |X˜ s|.
  - The assignment with the least total MSE is chosen and the model is optimized to reduce this particular MSE.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

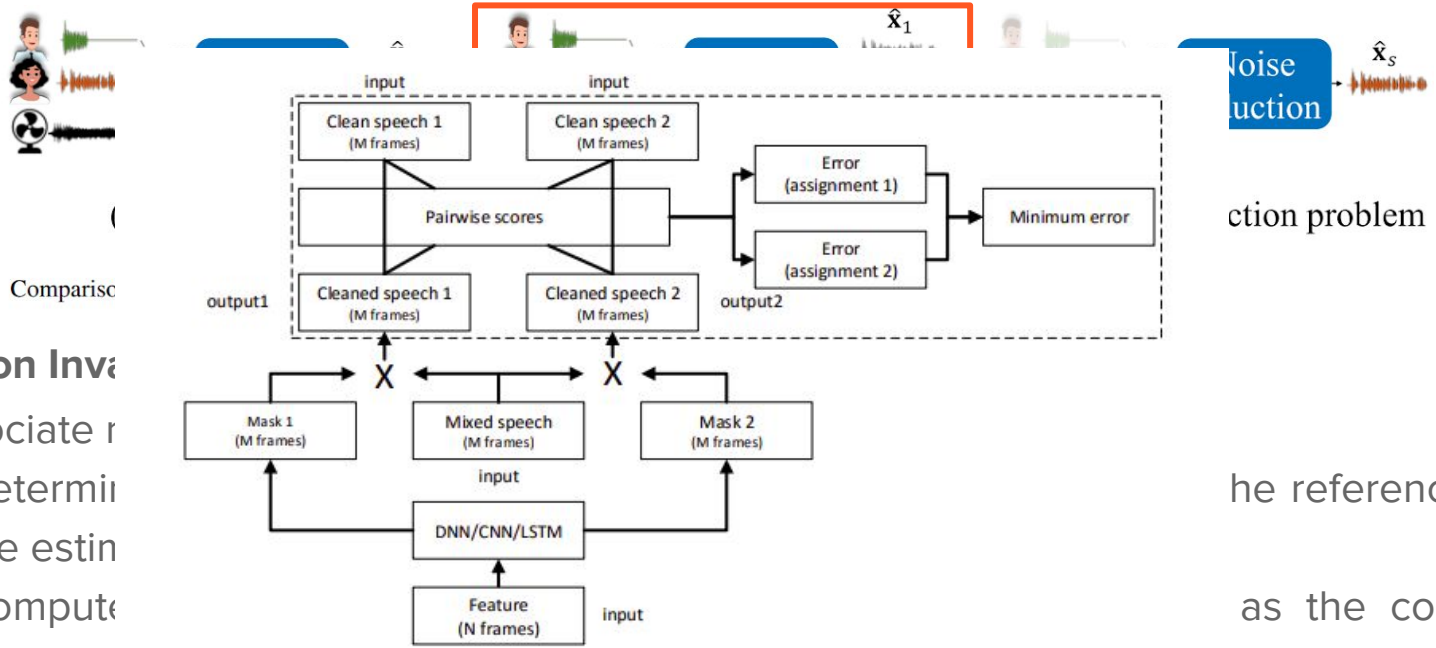


Fig. 1. The two-talker speech separation model with permutation invariant training.

ction problem

Fig. 2. Comparisc

Permutation Inva

- To associate r
  - Determi
    the estin
  - Compute
    pairwise
  - The ass
    reduce t

he references and

as the combined
e |X˜ s|.
lel is optimized to

Source 1, 2

Speech Enhancement- BSS- Methods- NN

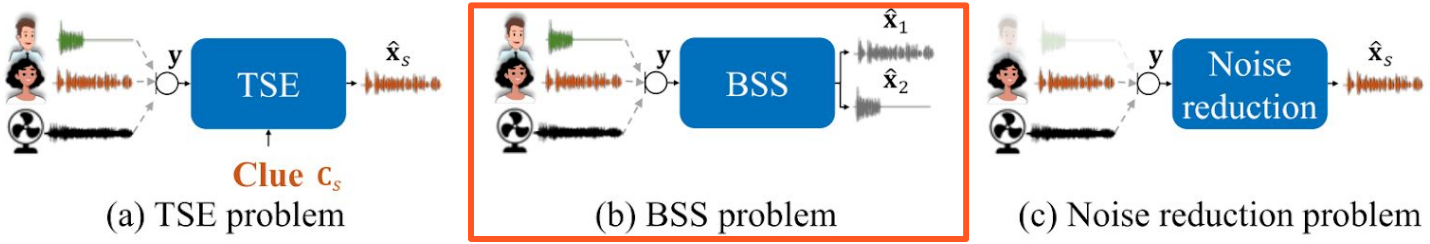


Fig. 2. Comparison of TSE with BSS and noise reduction

Permutation Invariant Loss (PIT)

- To associate references to the output layers:
  - Determine the (total number of S!) possible assignments between the references and the estimated sources. where Π_C is the set of all possible permutations of 1 . . . C.
  - Compute the SI-SNR for each assignment between the predicted and the target utterances, and sum over all assignments.
  - The assignment that maximizes the total SI-SNR is chosen and the model is optimized accordingly.

$$\ell(s, \hat{s}) = -\max_{\pi \in \Pi_C} \frac{1}{C} \sum_{i=1}^{C} \text{SI-SNR}(s_i \hat{s}_{\pi(i)})$$

Source 1, 2

Speech Enhancement- BSS- Methods

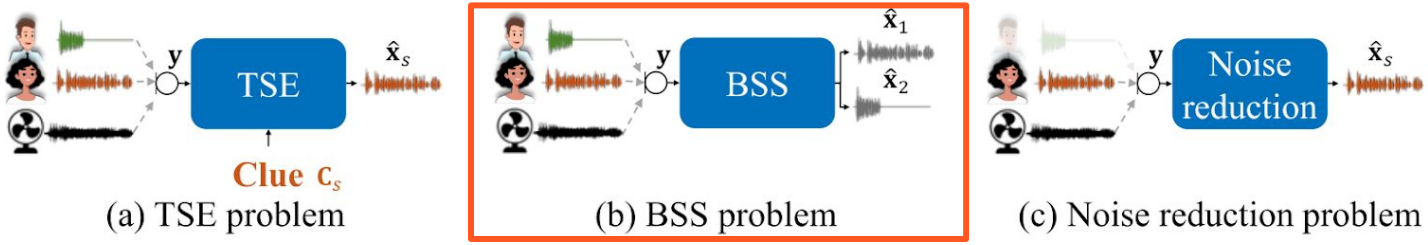


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - **TAS-NET**
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

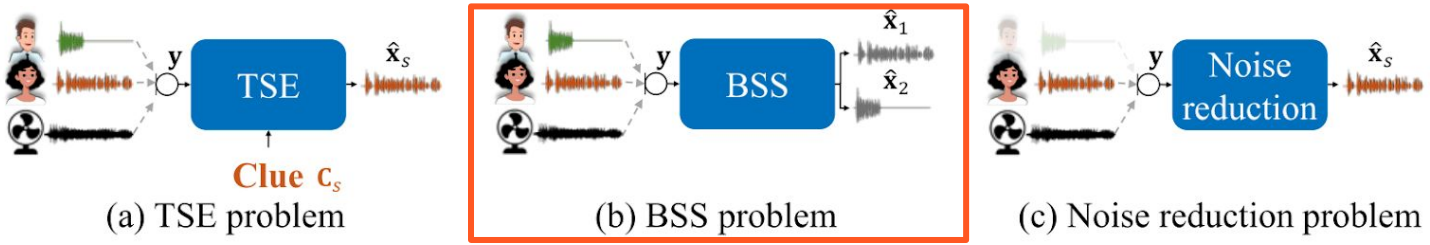


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Time-domain Audio Separation Network (TasNet)

- Current methods:
  - Construct a mask for each source in time-frequency representation of the mixture signal - not necessarily an optimal representation for speech separation.
  - Modify only the magnitude while reconstructing **using the noisy phase**
  - To achieve sufficient frequency resolution we need larger windows, resulting in high latency.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

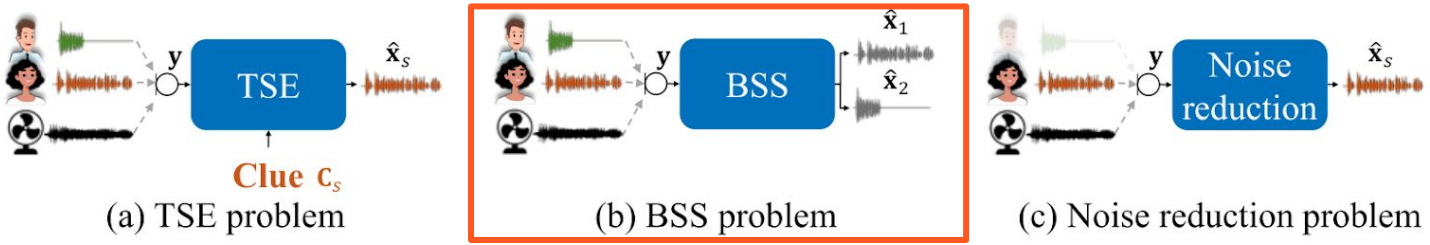
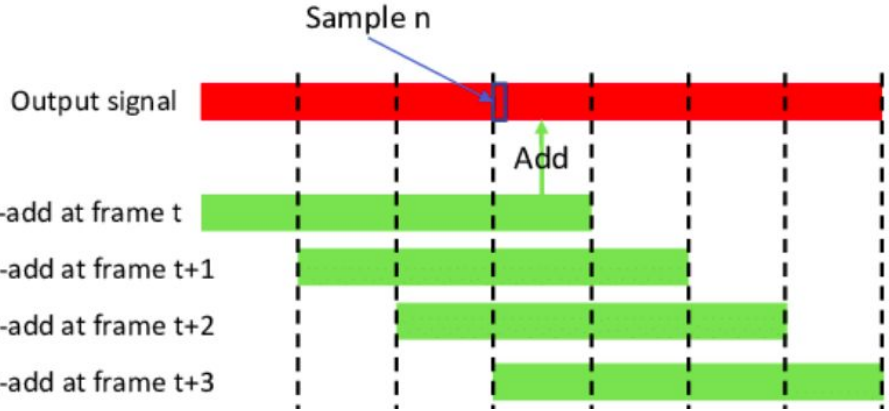


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- T

- Current
  - Co …n of the mixture sig… …on.
  - Mc… …e
  - To… …resulting in high lat…



Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

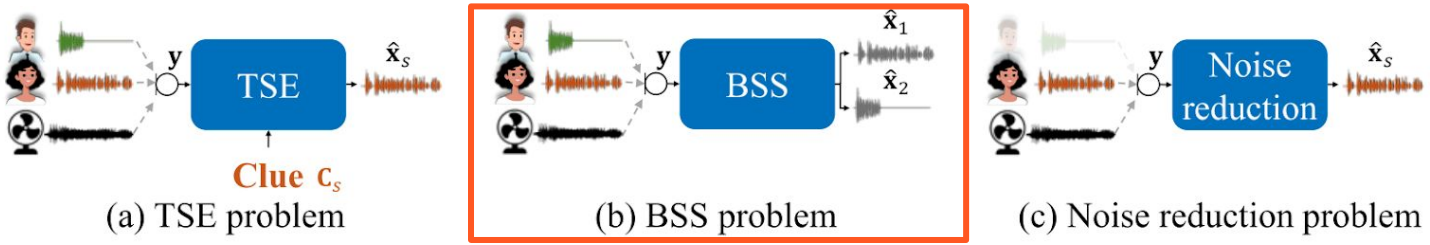


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Time-domain Audio Separation Network (TasNet)

- TasNet directly model the signal in the **time-domain** using an **encoder-decoder framework** and perform the **source separation on nonnegative encoder outputs**.
- Removes the frequency decomposition step and reduces the separation problem to estimation of source masks on encoder outputs which is then synthesized by the decoder.
- Address the challenge of real-time, short latency speech separation applications.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

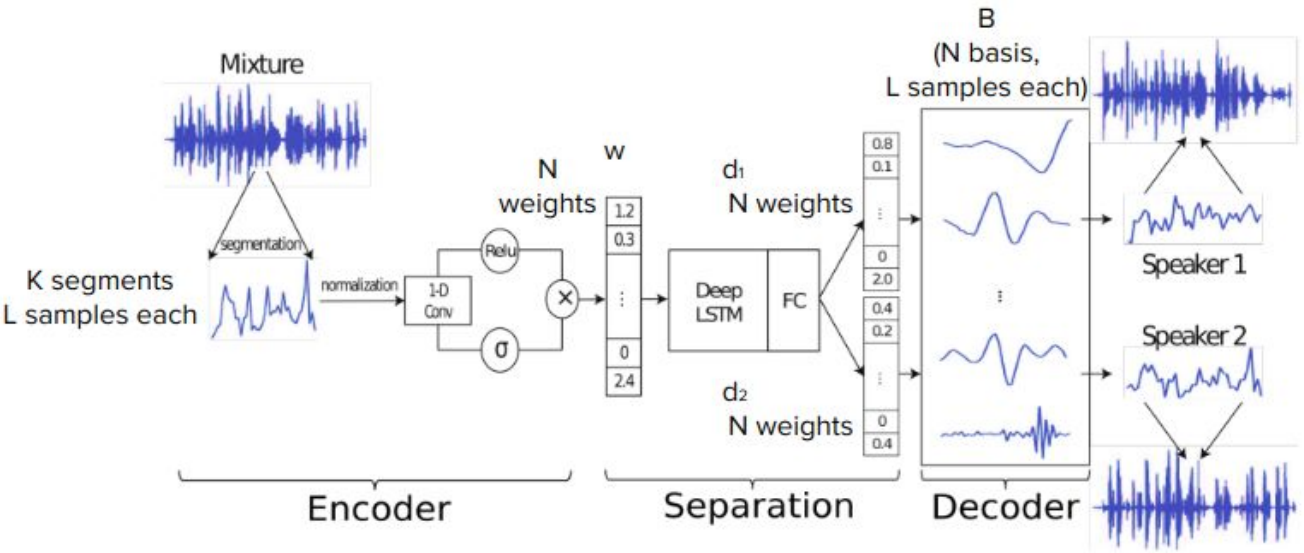


Fig. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder.

TAS-NE

- Tasl **framework** and
- Rem problem to esti e decoder.
- Add .

Source 1, 2

Speech Enhancement- BSS- Methods- NN

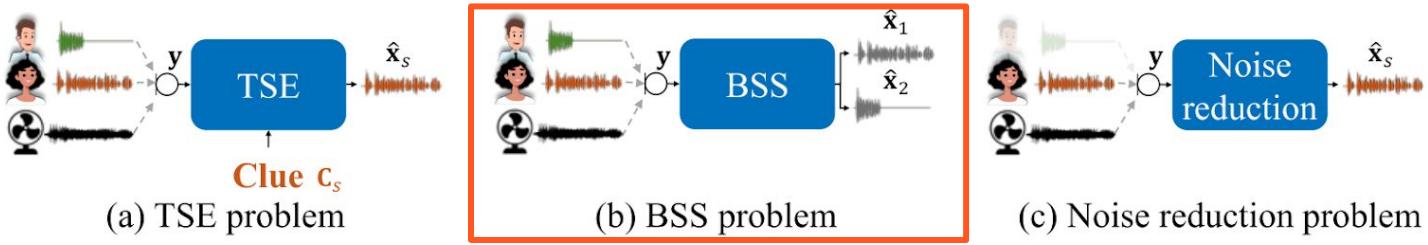


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Time-domain Audio Separation Network (TasNet)

given the discrete waveform of the mixture x(t):

- $x(t) = \sum_{i=1}^{C} s_i(t)$

We first segment the mixture and clean sources into K nonoverlapping vectors of length L samples, $x_k \in R^{1xL}$.

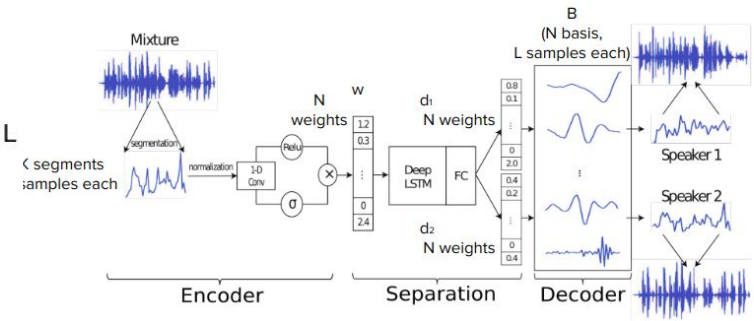


Fig. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

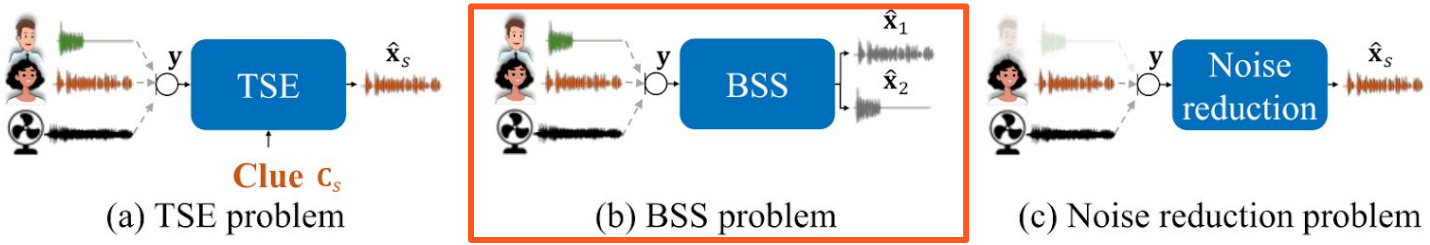


Fig. 2. Comparison of TSE with BSS and noise reduction

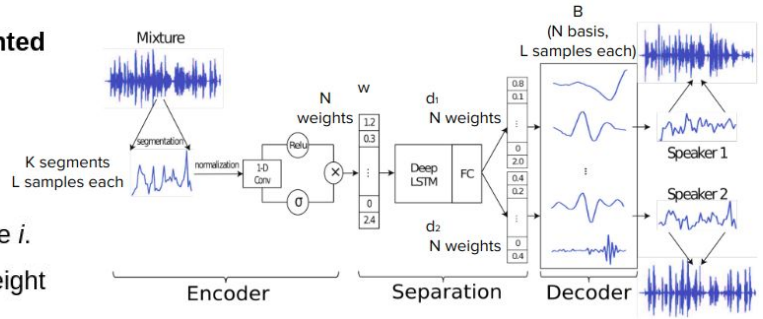
TAS-NET- Time-domain Audio Separation Network (TasNet)

Each segment of mixture and clean signals can be represented by a **nonnegative weighted sum of N basis signals** $B = [b_1, b_2, ..., b_N] \in R^{NxL}$, such that:

- $x_k = wB$
- $s_{i,k} = d_i B$

where $w \in R^{1xN}$ is the mixture weight vector, and $d_i \in R^{1xN}$ is the weight vector for the source *i*. Separating the sources in this representation is then reformulated as estimating the weight matrix of each source $d_i \in R^{1xN}$ given the mixture weight *w*, subject to:

$$w = \sum_{i=1}^{C} d_i$$



g. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and form the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture ights to reconstruct the sources. The source weights are then synthesized by the decoder.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

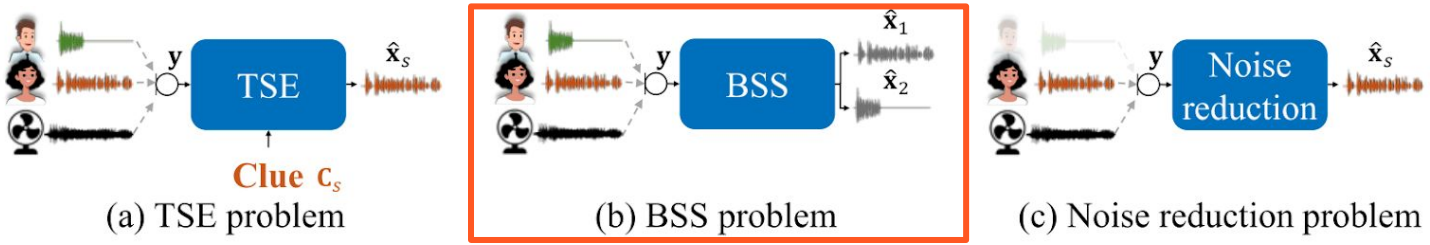


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Time-domain Audio Separation Network (TasNet)

Because all weights (w, d_i) are nonnegative, estimating the weight d_i of each source can be thought of as finding its corresponding mask-like vector, m_i, which is applied to the mixture weight, w, to recover d_i:

$d_i = m_i \odot w$

where $m_i \in R^{1xN}$ represents the **relative contribution** source i to the mixture weight *w*.

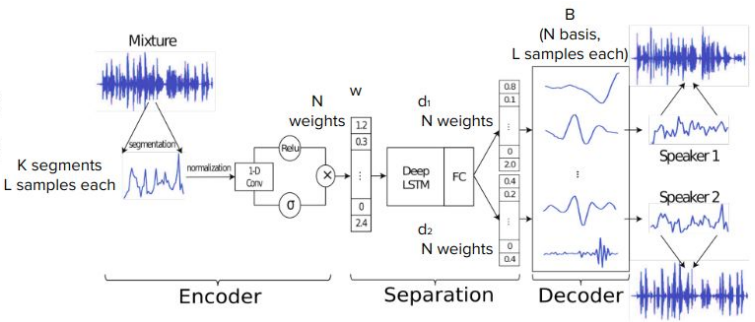


Fig. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

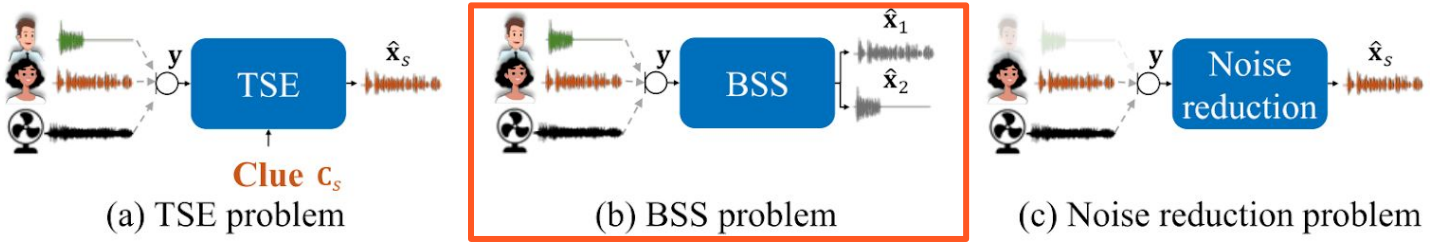


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Time-domain Audio Separation Network (TasNet)

- The **basis signals** (learned by the decoder) are **jointly optimized** with the other parameters of the separation network during training.
- The separation module has an LSTM layer that runs **over the K (temporal) segments**.

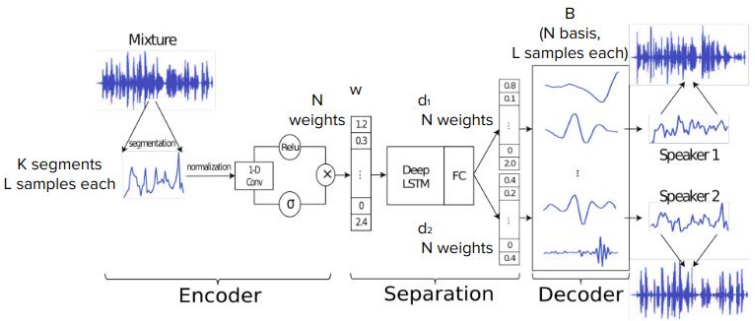


Fig. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

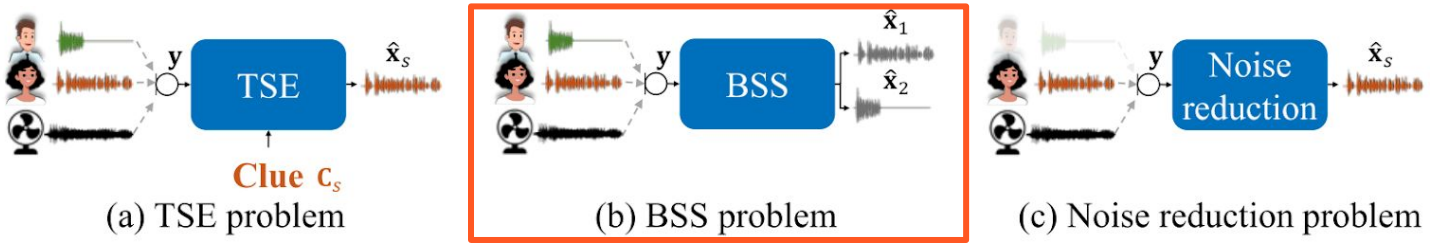


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Intuition

TAS-Net can be intuitively viewed either as:

- Encoder-decoder model, where encoder's outputs are being masked, and the masked representations are reconstructed using the decoder.
- ICA where the sources are not assumed to be independent, and there is no assumption regarding the non-Gaussianity of the sources.
- NMF where the basis and weights are learned jointly, and the basis are not forced to be non-negative (but the weights are).

This is something that connects the classic approach with the NN approach.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

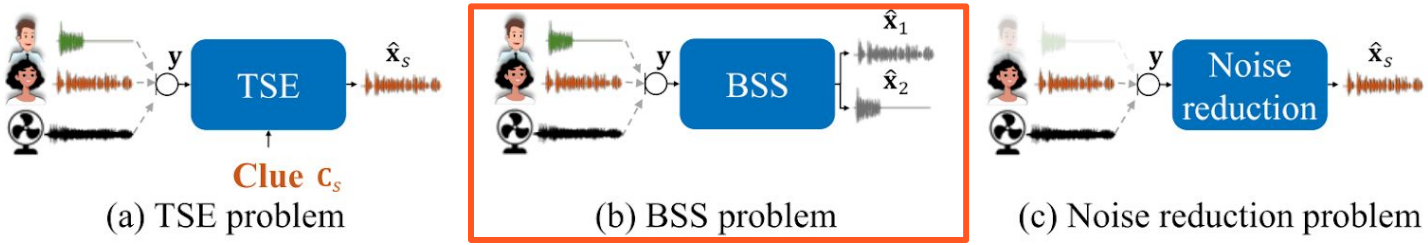


Fig. 2. Comparison of TSE with BSS and noise reduction

TAS-NET- Details

- Loss: SI-SNR with PIT.
- Dataset: WSJ0-2mix

Source 1, 2

Speech Enhancement- BSS- Methods

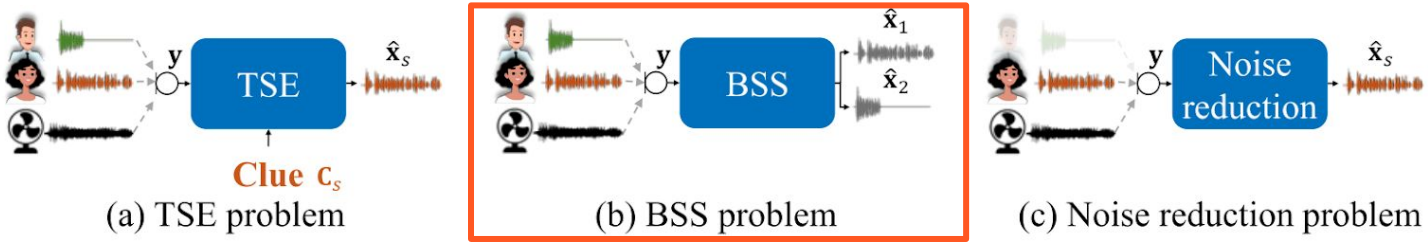


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - **Conv-TASNET**
  - **Dual Path RNN**
  - **GALR**
  - **SepFormer**
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

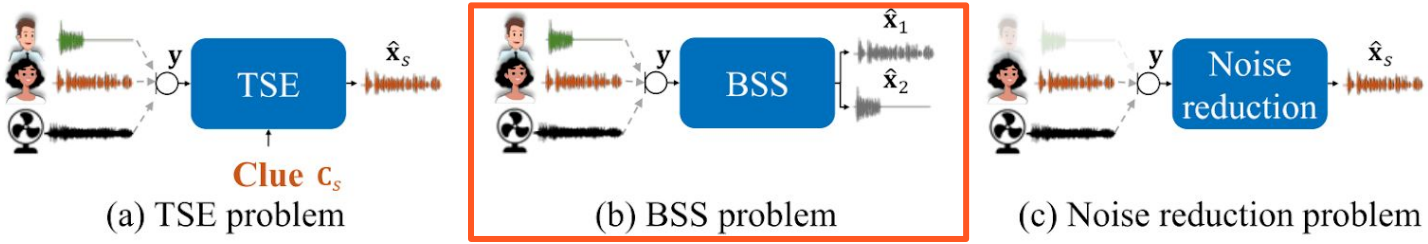
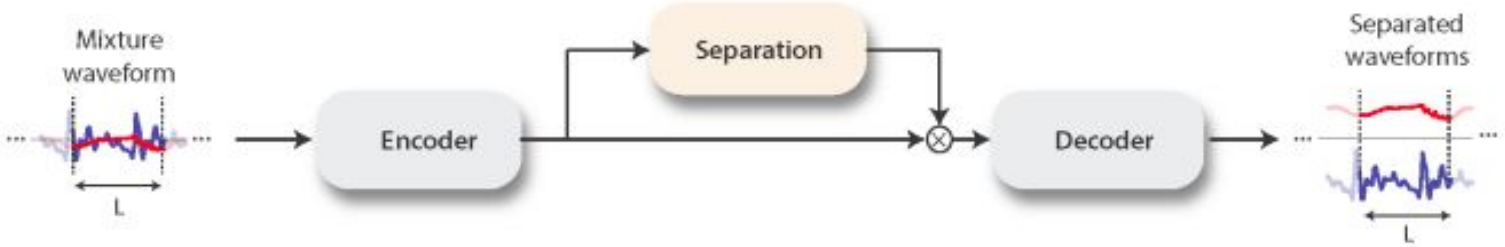


Fig. 2. Comparison of TSE with BSS and noise reduction

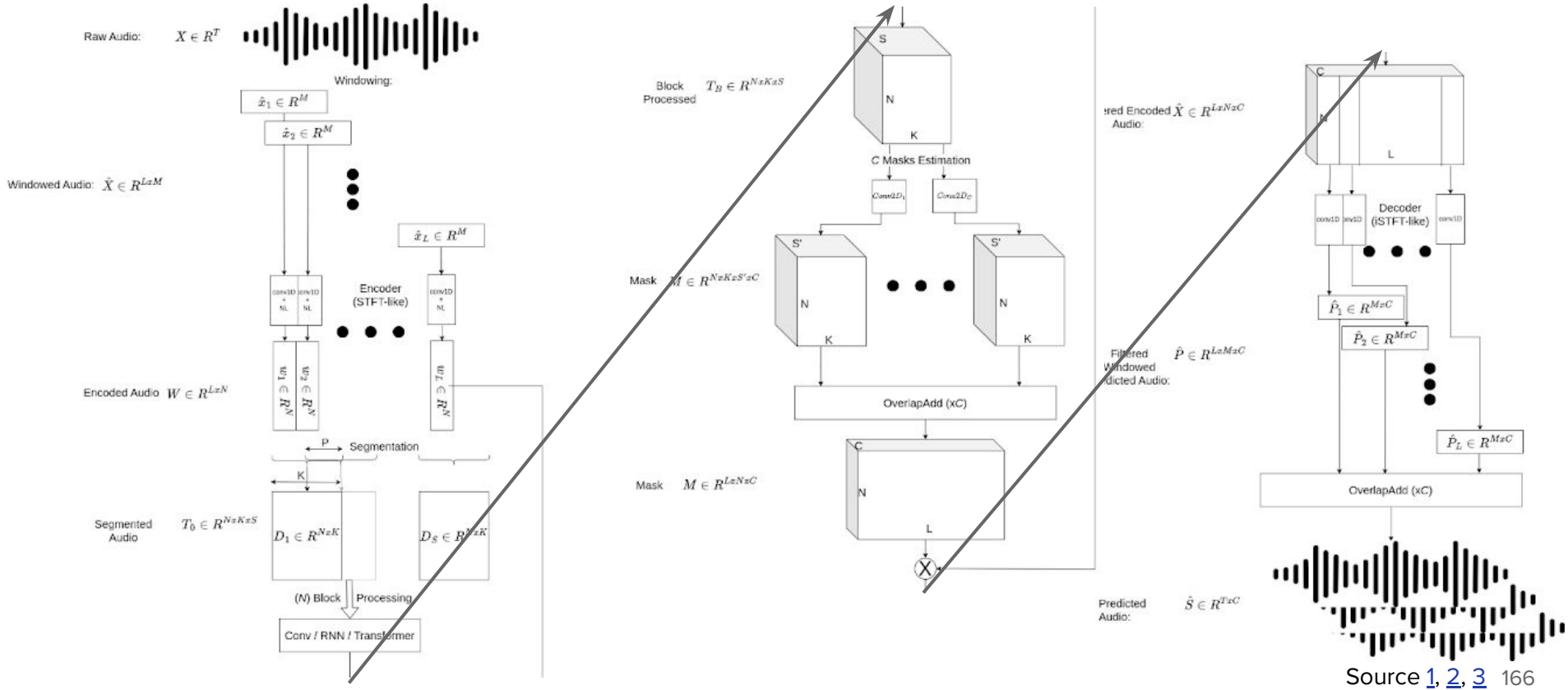
General Framework

- A general overview of Conv-TasNet, DPRNN, GALR, and Sepformer



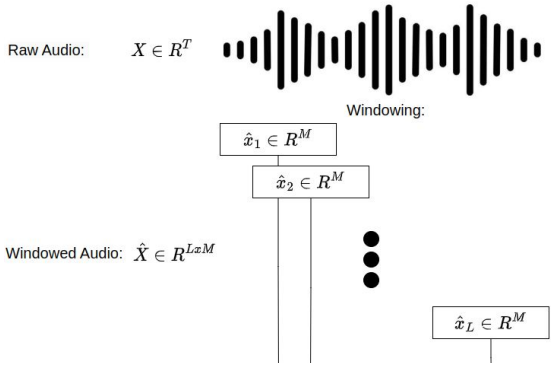
Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN



Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

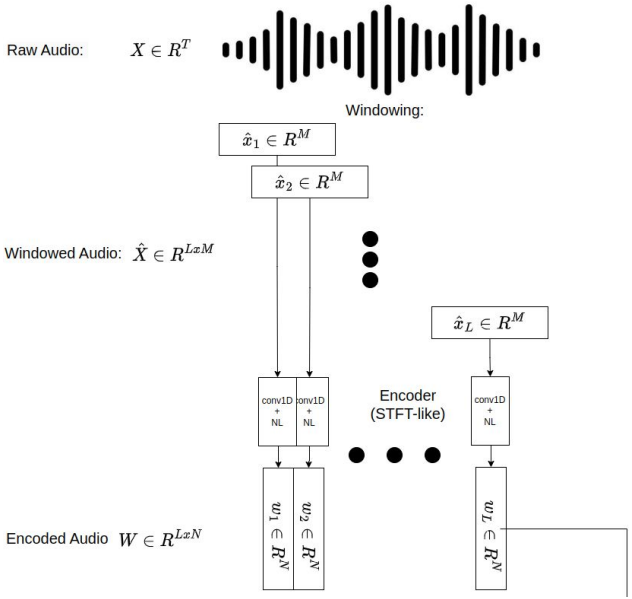


Encoder:

1. **Windowing**: similar to STFT, we apply windowing on the audio segments to get audio frames: In a TasNet-based separation system, an input mixture signal $X \in R^{MxL}$ is represented as *L* (half-overlapping) frames, denoted by $x_1,\ldots,x_L \in R^M$, where M denotes the window length.

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

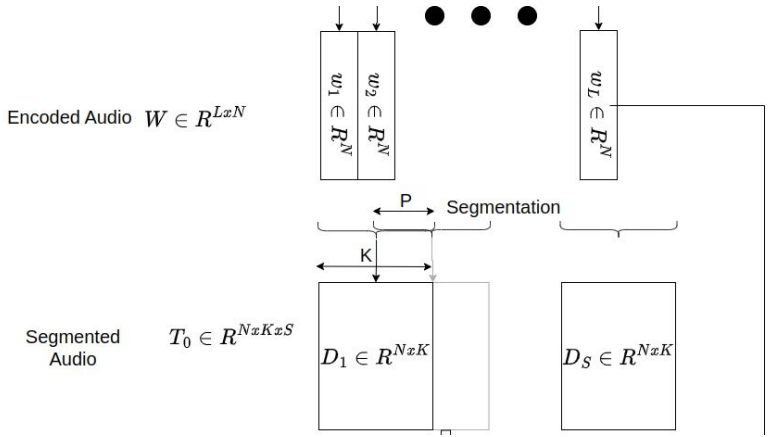


Encoder:

1. **Windowing**: similar to STFT, we apply windowing on the audio segments to get audio frames: In a TasNet-based separation system, an input mixture signal $X \in R^{MxL}$ is represented as *L* (half-overlapping) frames, denoted by $x_1,\ldots,x_L \in R^M$, where M denotes the window length.
2. **Encoding:** The encoder non-linearly transform each frame x_l into a *N*-dimensional feature vector $w_l \in R^N$ using a 1D gated convolutional layer: $w_l = ReLU(U * x_l) \in R^N$, where * denotes a 1D conv operator, and $U \in R^{NxM}$ contains *N* vectors (encoder basis functions) with length M each. This transform is an alternative to the STFT, and its dimensions are $W \in R^{NxL}$.

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

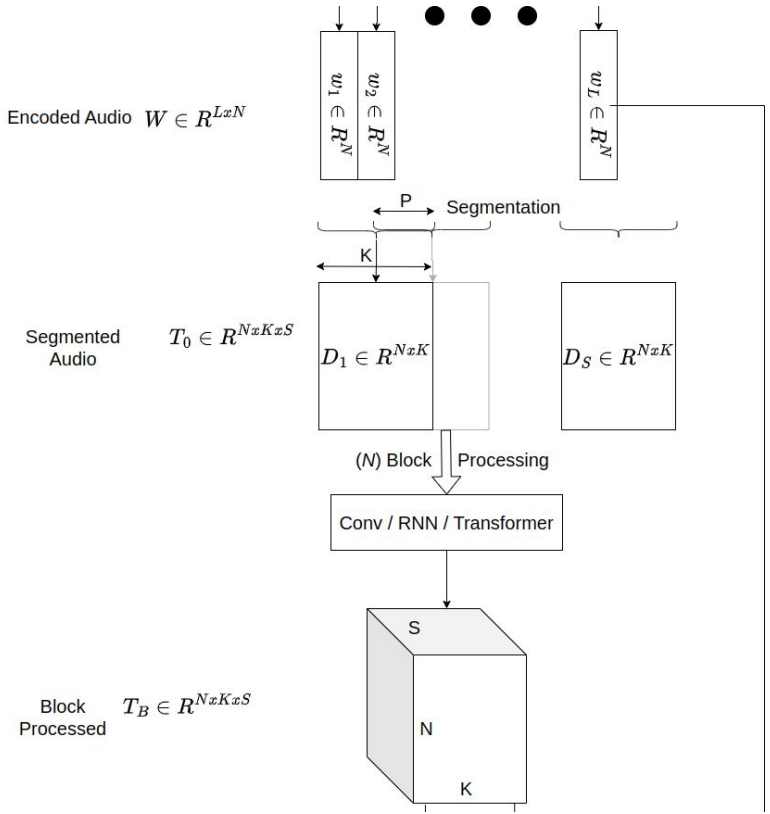


Separation Network:

1. **Segmentation:** Given an encoded signal input $W \in R^{NxL}$ where ***N* is the feature dimension** and ***L* is the number of time frames (steps)**, the segmentation stage splits W into chunks of **length *K*** and **hop size *P***. The first and last chunks are zero-padded so that every sample in W appears and only appears in K/P chunks, generating **S equal size chunks** $D_s \in R^{NxK}$, $s = 1,\dots,S$. All chunks are then concatenated together to form a 3-D tensor $T = [D_1,\dots,D_S] \in R^{NxKxS}$. Note that the **length of each chunk (K)** is a hyperparameter that affects the number of segments and can be used to **control the scale of the locality**.

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

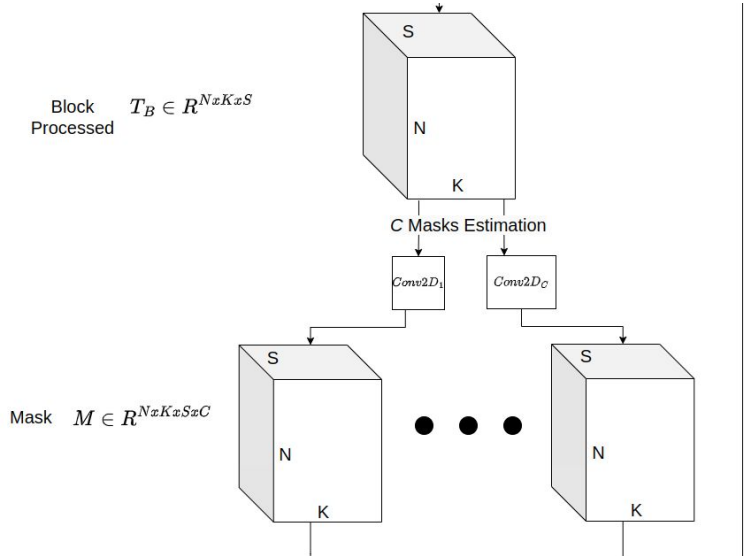


Separation Network:

1. **Segmentation:** Given an encoded signal input $W \in R^{NxL}$ where ***N* is the feature dimension** and ***L* is the number of time frames (steps)**, the segmentation stage splits W into chunks of **length *K*** and **hop size *P***. The first and last chunks are zero-padded so that every sample in W appears and only appears in K/P chunks, generating **S equal size chunks** $D_s \in R^{NxK}$, $s = 1,\dots,S$. All chunks are then concatenated together to form a 3-D tensor $T = [D_1,\dots, D_S] \in R^{NxKxS}$. Note that the **length of each chunk (K)** is a hyperparameter that affects the number of segments and can be used to **control the scale of the locality**.
2. **Block Processing:** The segmentation output T is then passed to the block processing module. Some consist of CNNs, RNNs, Transformers, or their permutation. We'll discuss it shortly when reviewing the papers. The output of this module, T_B, has the same dimension as the input $T_B \in R^{NxKxS}$

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

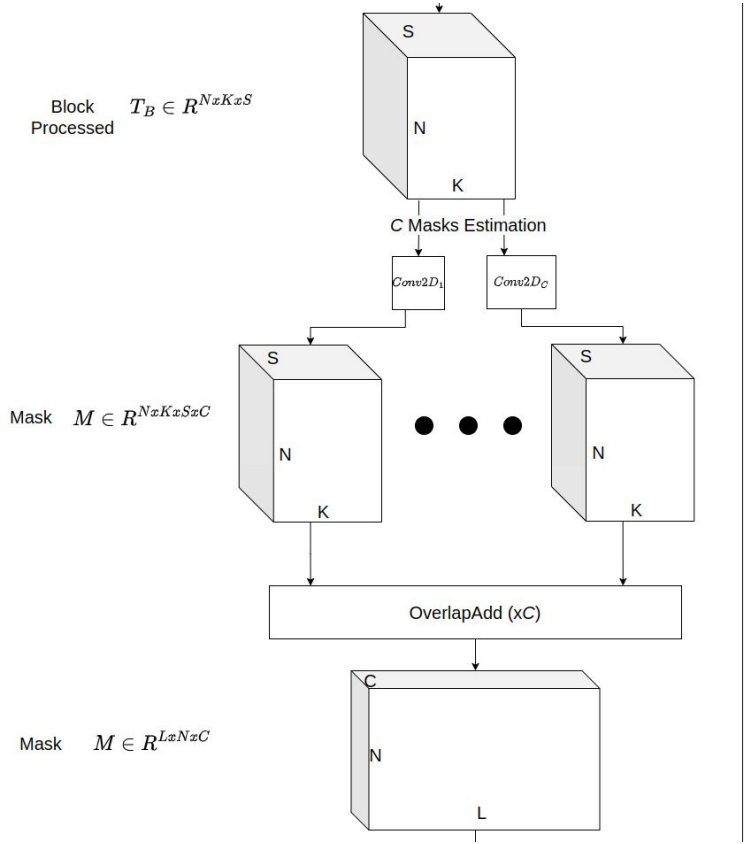


Separation Network:

1. **Segmentation:** Given an encoded signal input $W \in R^{NxL}$ where ***N* is the feature dimension** and ***L* is the number of time frames (steps)**, the segmentation stage splits W into chunks of **length *K*** and **hop size *P***. The first and last chunks are zero-padded so that every sample in W appears and only appears in K/P chunks, generating **S equal size chunks** $D_s \in R^{NxK}$, $s = 1,\dots,S$. All chunks are then concatenated together to form a 3-D tensor $T = [D_1,\dots, D_S] \in R^{NxKxS}$. Note that the **length of each chunk (K)** is a hyperparameter that affects the number of segments and can be used to **control the scale of the locality**.
2. **Block Processing:** The segmentation output T is then passed to the block processing module. Some consist of CNNs, RNNs, Transformers, or their permutation. We'll discuss it shortly when reviewing the papers. The output of this module, T_B, has the same dimension as the input $T_B \in R^{NxKxS}$
3. **Mask Estimation:** They apply a transformation (conv2d for example) on the block processed output T_B, and result in *C* masks $M_c \in R^{NxKxS}$, $c = 1,\dots,C$. They apply a different transformation per mask (otherwise they will get the same mask for all sources). This means that the model is constrained to output *C* signal (*C* is a property of the model).

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN



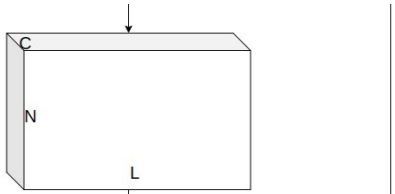
Separation Network:

1. **Segmentation:** Given an encoded signal input $W \in R^{NxL}$ where ***N* is the feature dimension** and ***L* is the number of time frames (steps)**, the segmentation stage splits W into chunks of **length *K*** and **hop size *P***. The first and last chunks are zero-padded so that every sample in W appears and only appears in K/P chunks, generating **S equal size chunks** $D_s \in R^{NxK}$, $s = 1,\dots,S$. All chunks are then concatenated together to form a 3-D tensor $T = [D_1,\dots, D_S] \in R^{NxKxS}$. Note that the **length of each chunk (K)** is a hyperparameter that affects the number of segments and can be used to **control the scale of the locality**.
2. **Block Processing:** The segmentation output T is then passed to the block processing module. Some consist of CNNs, RNNs, Transformers, or their permutation. We'll discuss it shortly when reviewing the papers. The output of this module, T_B, has the same dimension as the input $T_B \in R^{NxKxS}$
3. **Mask Estimation:** They apply a transformation (conv2d for example) on the block processed output T_B, and result in *C* masks $M_c \in R^{NxKxS}$, $c = 1,\dots,C$. They apply a different transformation per mask (otherwise they will get the same mask for all sources). This means that the model is constrained to output *C* signal (*C* is a property of the model).
4. **Overlap Add**: to get the final mask, they apply OverlapAdd operator on S chunks of the mask, which results in $Q \in R^{NxLXC}$, which can be viewed as *C* tensors, where each has the same dimensions as the encoded audio $W \in R^{NxL}$.

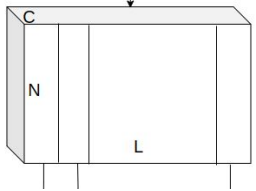
Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

Mask $M \in R^{LxNxC}$



Filtered Encoded Audio: $\hat{X} \in R^{LxNxC}$



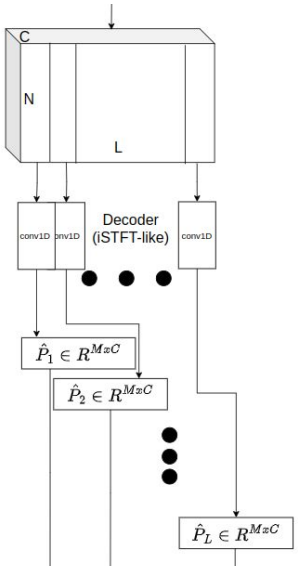
Decoder:

1. **Mask Multiplication:** Each of the masks multiplies W to form the filtered encoded audio

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

Filtered Encoded Audio: $\hat{X} \in R^{LxNxC}$



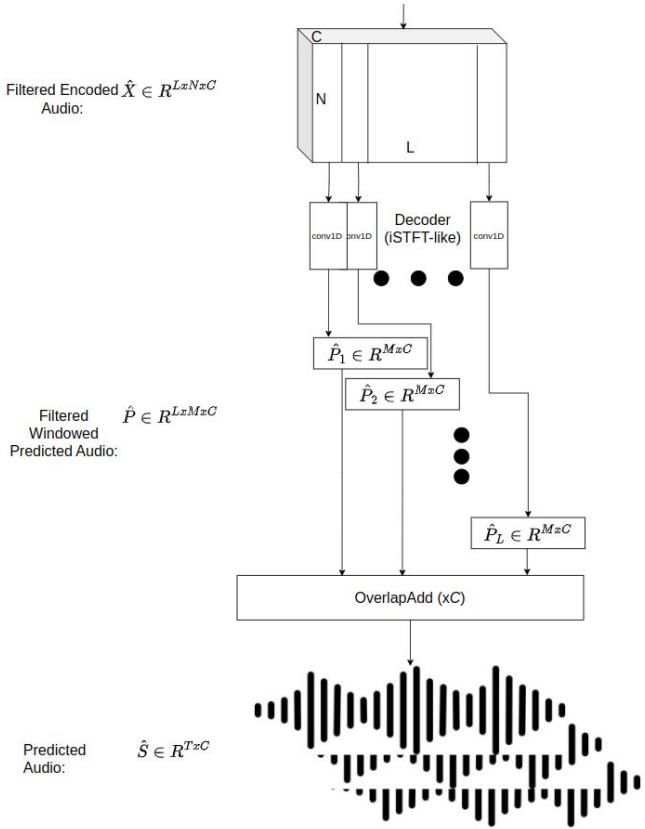
Filtered Windowed Predicted Audio: $\hat{P} \in R^{LxMxC}$

Decoder:

1. **Mask Multiplication:** Each of the masks multiplies W to form the filtered encoded audio
2. **Decoding:** The filtered encoded audio is transformed back to the audio domain using an inverse transform of the encoder (similar to iSTFT), to for $\hat{P_c} = B(W \odot Q_c) \in R^{LxM}$, $c = 1,\ldots,C$. Where $B \in R^{MxN}$ containing the basis signals with each column corresponding to a 1-D filter, and $\odot$ is the element-wise multiplication operator. This results the filtered predicted windowed audio $\hat{P} \in R^{LxMxC}$.

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN



Decoder:

1. **Mask Multiplication:** Each of the masks multiplies W to form the filtered encoded audio
2. **Decoding:** The filtered encoded audio is transformed back to the audio domain using an inverse transform of the encoder (similar to iSTFT), to for $\hat{P_c} = B(W \odot Q_c) \in R^{LxM}$, $c = 1,\ldots,C$. Where $B \in R^{MxN}$ containing the basis signals with each column corresponding to a 1-D filter, and $\odot$ is the element-wise multiplication operator. This results the filtered predicted windowed audio $\hat{P} \in R^{LxMxC}$.
3. **Overlap Add:** finally, OverlapAdd operator is applied for each of the C sources separately, to form C sources estimates, $\hat{S_c} \in R^T$, $c = 1,\ldots,C$

Source 1, 2, 3

Speech Enhancement- BSS- Methods- NN

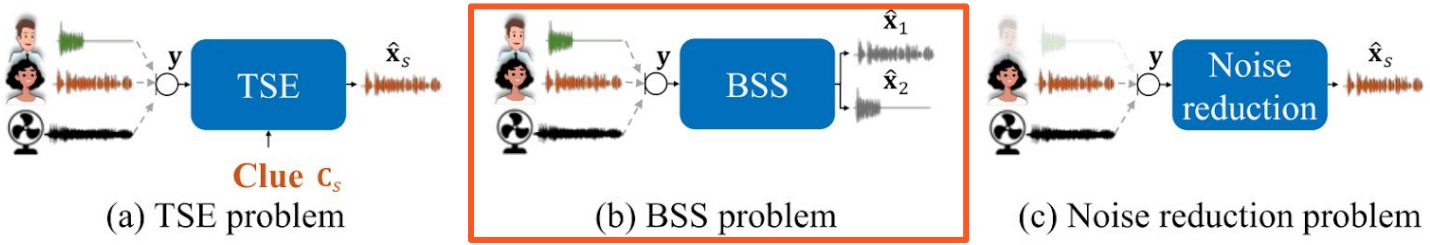


Fig. 2. Comparison of TSE with BSS and noise reduction

Important note

- As these models are relatively small in number of params, the intermediate tensors are large (thus has large memory consumption).
- Hence, it's hard to train these models using large (>2 samples depends on the HW) mini-batches, which results in long training time.
- If one tries to reduce the tensor dimensions, the model becomes not expressive enough.

Source 1, 2, 3

Speech Enhancement- BSS- Methods

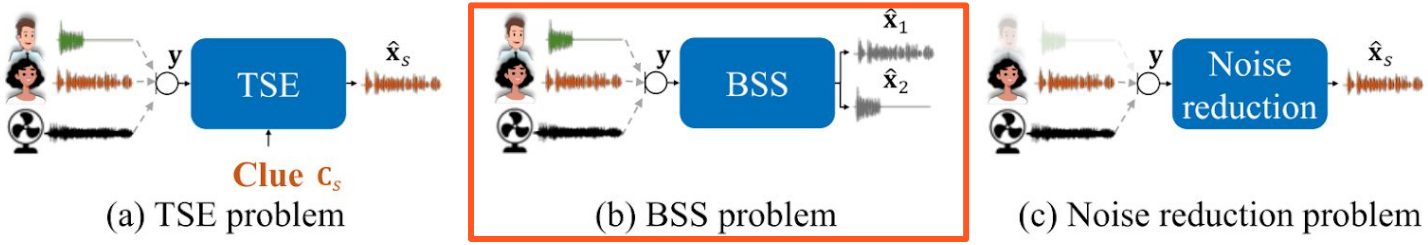


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - **Conv-TASNET**
  - Dual Path RNN
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

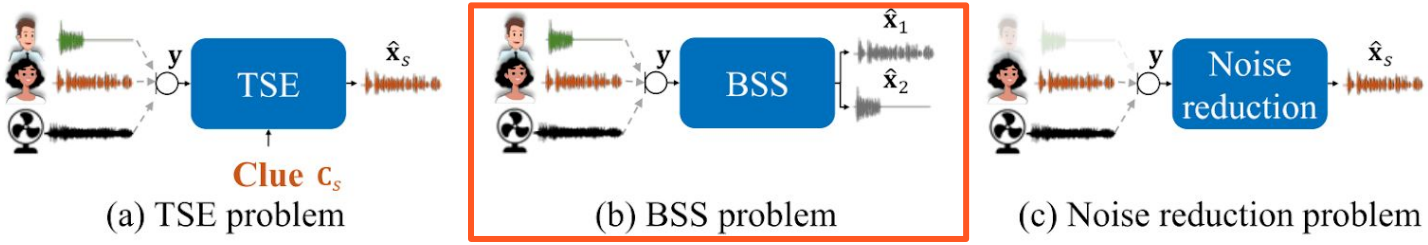


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET

The main drawback of the TAS-Net relies in the utilization of LSTM network in the separation module:

- Choosing a smaller kernel size (i.e. length of segments L) in the encoder increases the length of the encoder output K, which makes the training of the LSTMs unmanageable.
- The large number of parameters in deep LSTM network significantly increases its computational cost and limits its applicability to low resource/power platforms such as wearable hearing devices.
- Long temporal dependencies of LSTM networks often results in inconsistent separation accuracy, for example, when changing the starting point of the mixture.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

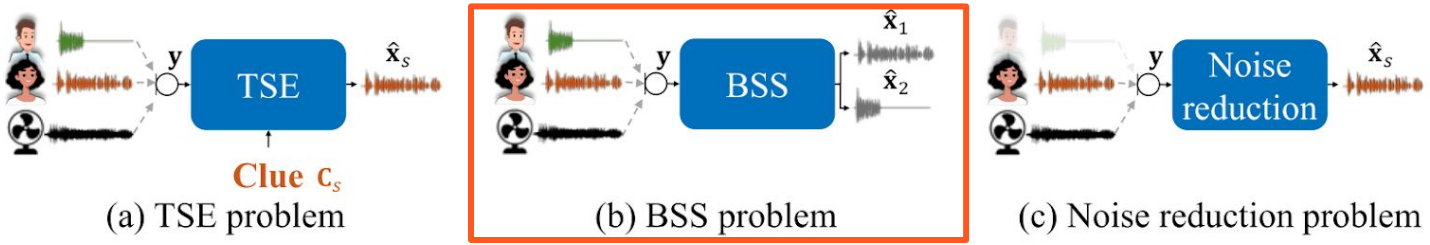


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET

Propose the fully-convolutional TasNet (Conv-TasNet) that uses only convolutional layers in all stages of processing.

- Overcomes the problem of the LSTMs
- CNNs can be optimized for edge computing

Source 1, 2

Speech Enhancement- BSS- Methods- NN

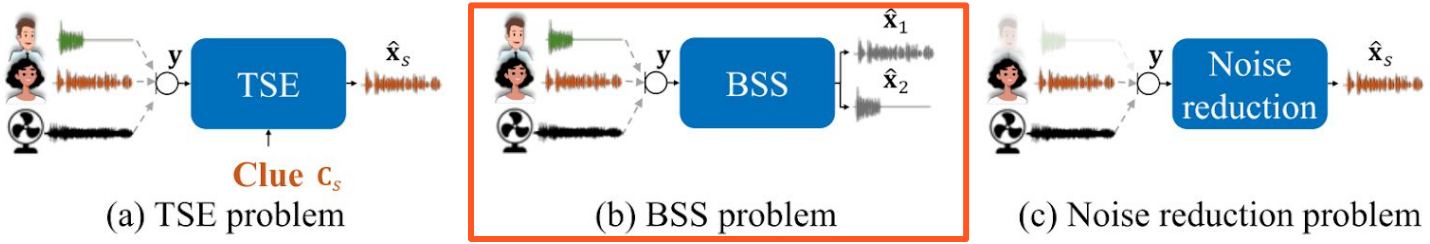


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET - Architecture

- **Encoder** (1-D conv) is used to transform short segments of the mixture waveform into their corresponding representations in an intermediate feature space.

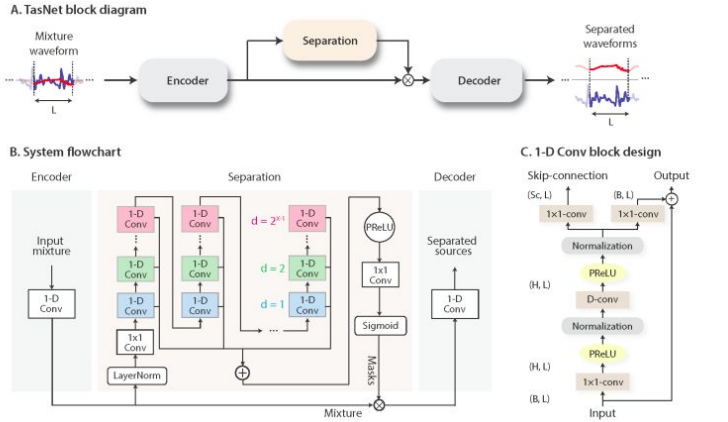


Fig. 1. (A): the block diagram of the TasNet system. An encoder maps a segment of the mixture waveform to a high-dimensional representation and a separation module calculates a multiplicative function (i.e., a mask) for each of the target sources. A decoder reconstructs the source waveforms from the masked features. (B): A flowchart of the proposed system. A 1-D convolutional autoencoder models the waveforms and a temporal convolutional network (TCN) separation module estimates the masks based on the encoder output. Different colors in the 1-D convolutional blocks in TCN denote different dilation factors. (C): The design of 1-D convolutional block. Each block consists of a 1×1-$conv$ operation followed by a depthwise convolution ($D-conv$) operation, with nonlinear activation function and normalization added between each two convolution operations. Two linear 1×1-$conv$ blocks serve as the residual path and the skip-connection path respectively.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

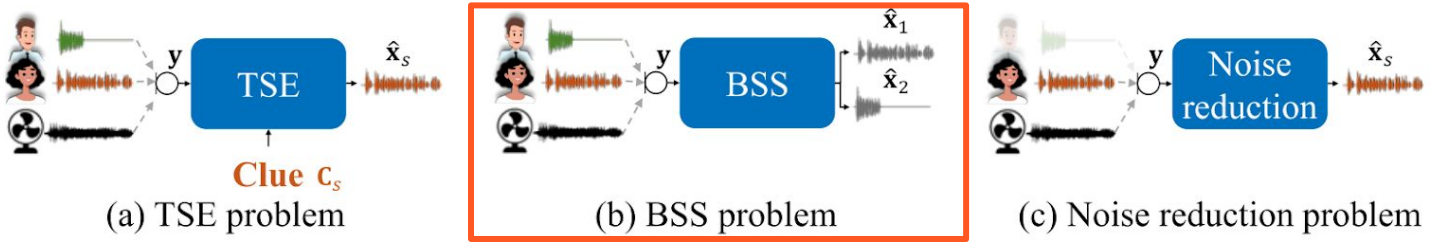


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET - Architecture

- **Separation:** The encoded representations are used to estimate a multiplicative function (mask) for each source at each time step.
  - Instead of using LSTMs (TASNET), Conv-TASNET uses dilated convolutions for long dependencies.
  - The output is passed to a point-wize convolution layer to estimate C mask vectors for the C target sources.

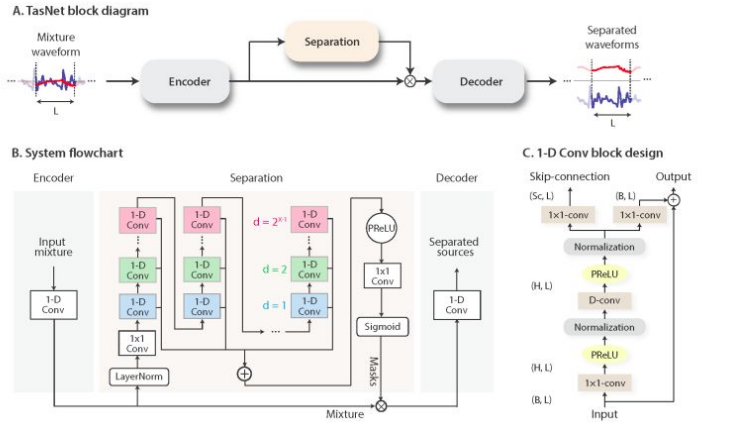


Fig. 1. (A): the block diagram of the TasNet system. An encoder maps a segment of the mixture waveform to a high-dimensional representation and a separation module calculates a multiplicative function (i.e., a mask) for each of the target sources. A decoder reconstructs the source waveforms from the masked features. (B): A flowchart of the proposed system. A 1-D convolutional autoencoder models the waveforms and a temporal convolutional network (TCN) separation module estimates the masks based on the encoder output. Different colors in the 1-D convolutional blocks in TCN denote different dilation factors. (C): The design of 1-D convolutional block. Each block consists of a 1×1-*conv* operation followed by a depthwise convolution (*D*−*conv*) operation, with nonlinear activation function and normalization added between each two convolution operations. Two linear 1×1−*conv* blocks serve as the residual path and the skip-connection path respectively.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

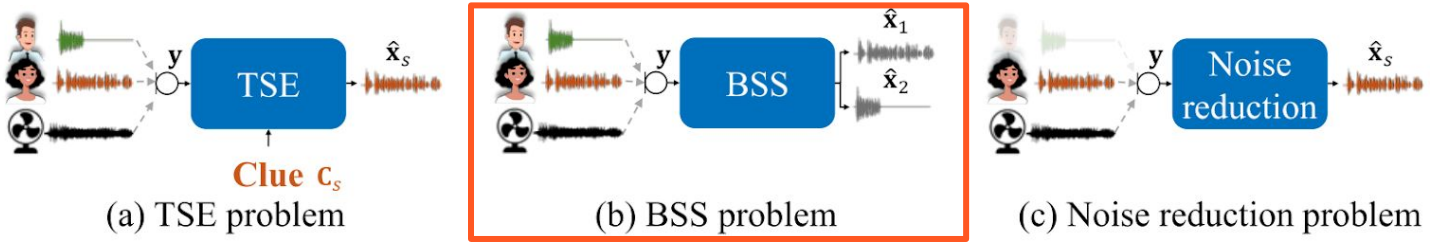


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET - Architecture

- **Decoder**- The source waveforms are then reconstructed by transforming the masked encoder features using a decoder module. Here it is a de-conv 1-D model.

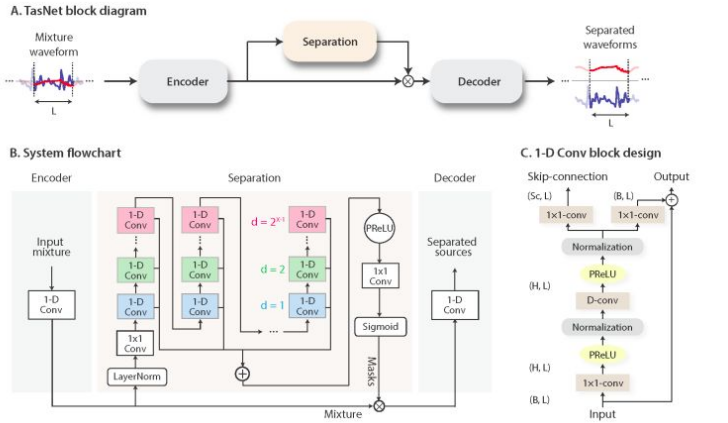


Fig. 1. (A): the block diagram of the TasNet system. An encoder maps a segment of the mixture waveform to a high-dimensional representation and a separation module calculates a multiplicative function (i.e., a mask) for each of the target sources. A decoder reconstructs the source waveforms from the masked features. (B): A flowchart of the proposed system. A 1-D convolutional autoencoder models the waveforms and a temporal convolutional network (TCN) separation module estimates the masks based on the encoder output. Different colors in the 1-D convolutional blocks in TCN denote different dilation factors. (C): The design of 1-D convolutional block. Each block consists of a 1×1-$conv$ operation followed by a depthwise convolution ($D-conv$) operation, with nonlinear activation function and normalization added between each two convolution operations. Two linear 1×1-$conv$ blocks serve as the residual path and the skip-connection path respectively.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

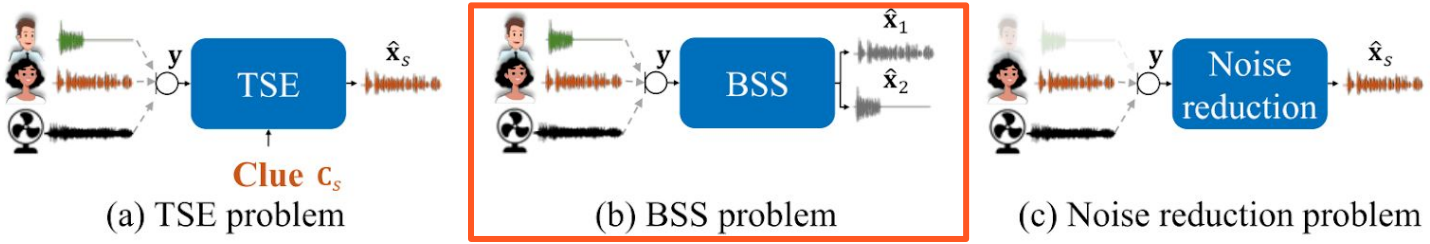


Fig. 2. Comparison of TSE with BSS and noise reduction

Conv TAS-NET - Results

- Conv-TasNet has a significantly smaller model size and a shorter minimum latency, making it a suitable solution for both offline and real-time speech separation applications.
- Had significant improvement compared to deep clustering!

Link for audio samples

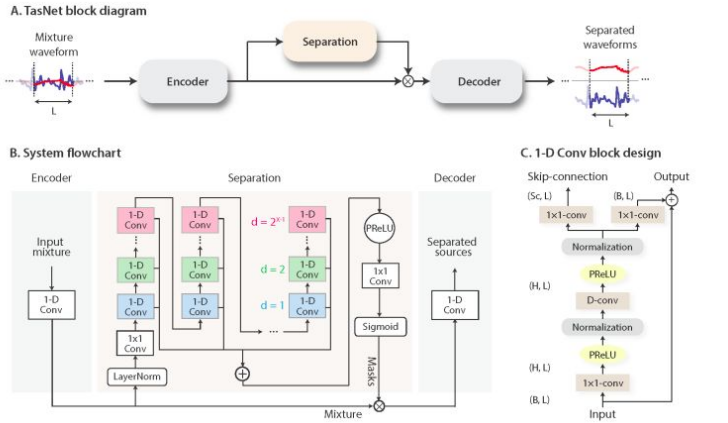


Fig. 1. (A): the block diagram of the TasNet system. An encoder maps a segment of the mixture waveform to a high-dimensional representation and a separation module calculates a multiplicative function (i.e., a mask) for each of the target sources. A decoder reconstructs the source waveforms from the masked features. (B): A flowchart of the proposed system. A 1-D convolutional autoencoder models the waveforms and a temporal convolutional network (TCN) separation module estimates the masks based on the encoder output. Different colors in the 1-D convolutional blocks in TCN denote different dilation factors. (C): The design of 1-D convolutional block. Each block consists of a 1×1-$conv$ operation followed by a depthwise convolution ($D-conv$) operation, with nonlinear activation function and normalization added between each two convolution operations. Two linear 1×1-$conv$ blocks serve as the residual path and the skip-connection path respectively.

Source 1, 2

Speech Enhancement- BSS- Methods

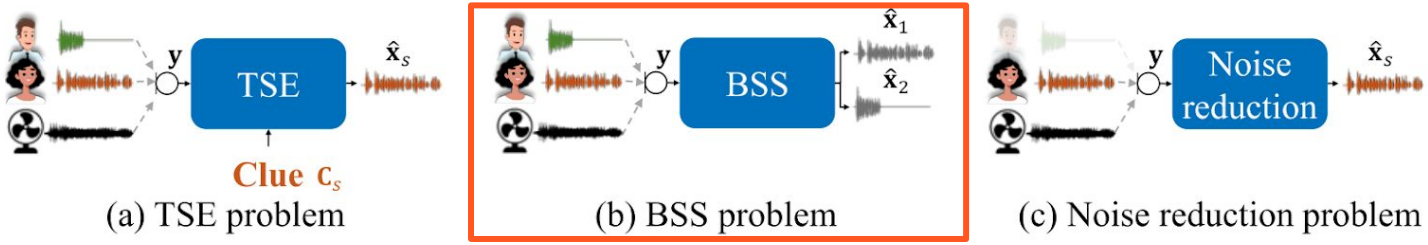


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - **Dual Path RNN**
  - GALR
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

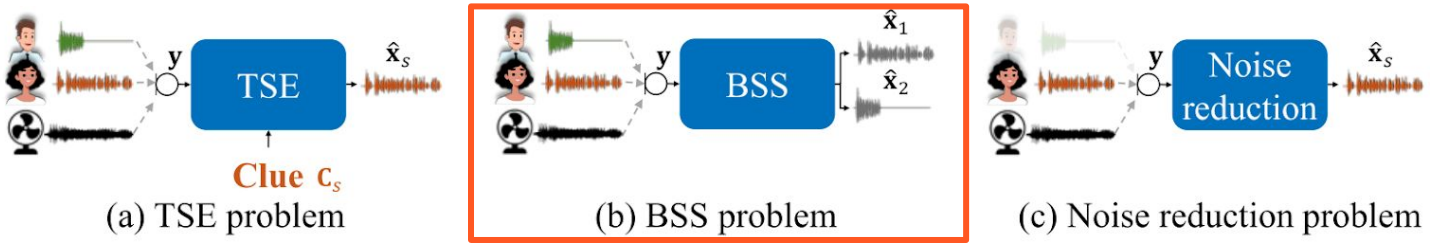


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN)

EFFICIENT LONG SEQUENCE MODELING FOR TIME-DOMAIN SINGLE-CHANNEL SPEECH SEPARATION

Source 1, 2

Speech Enhancement- BSS- Methods- NN

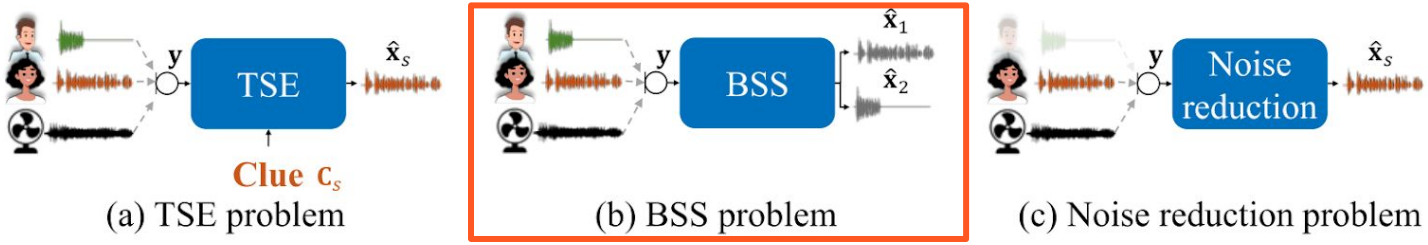


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN) - Challenges

- Recent studies in deep learning-based speech separation have proven the superiority of time-domain approaches to conventional time frequency-based methods.
- The time-domain separation systems often receive **extremely long sequences** (tens of thousands, or sometimes even more), which introduces challenges for modeling.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

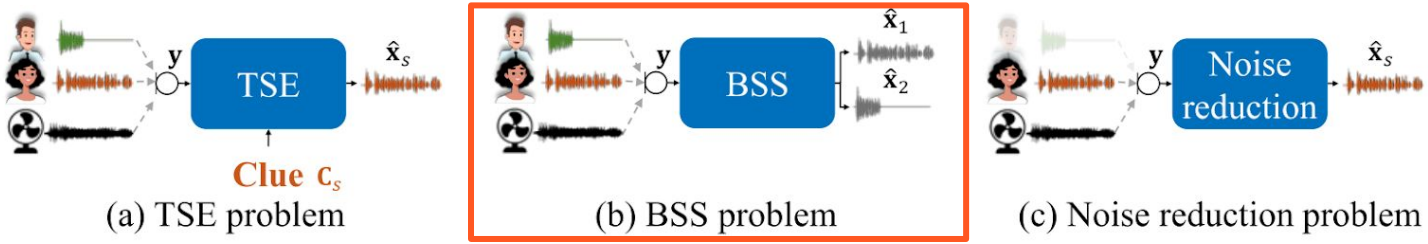


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN)- Solution

- Propose a simple network architecture, refer to as dual-path RNN (DPRNN), that organizes any kinds of RNN layers to model long sequential inputs in a very simple way.
- The intuition is to split the input sequence into shorter chunks and apply **DPRNN block consists of interleave two RNNs**, an **intra-chunk RNN and an inter-chunk RNN, for local and global modeling, respectively**.
    - The **intra-chunk** RNN first processes the **local chunks** independently,
    - The **inter-chunk** RNN aggregates the information from all the chunks to perform **utterance-level (global)** processing.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

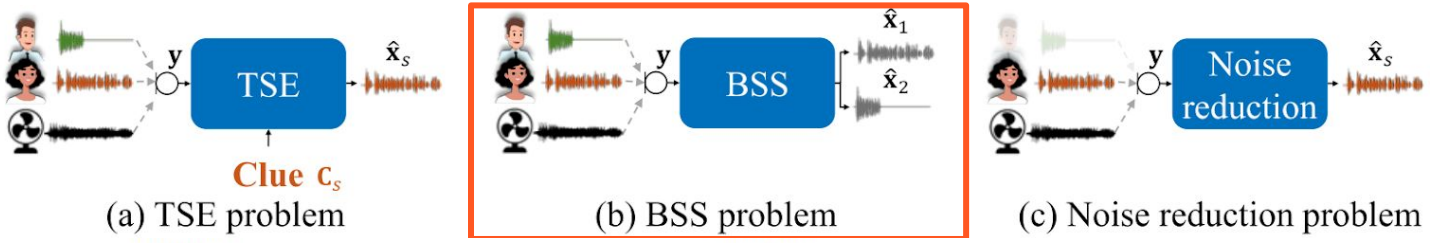


Fig. 2.

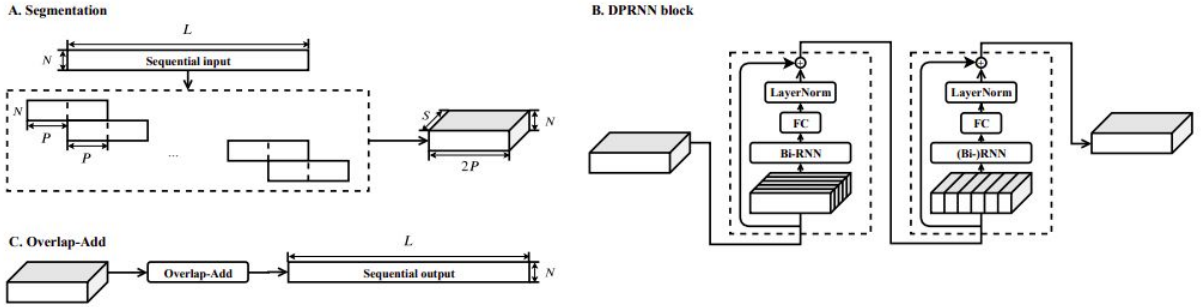


Fig. 1. System flowchart of dual-path RNN (DPRNN). (A) The segmentation stage splits a sequential input into chunks with or without overlaps and concatenates them to form a 3-D tensor. In our implementation, the overlap ratio is set to 50%. (B) Each DPRNN block consists of two RNNs that have recurrent connections in different dimensions. The *intra-chunk* bi-directional RNN is first applied to individual chunks in parallel to process local information. The *inter-chunk* RNN is then applied across the chunks to capture global dependency. Multiple blocks can be stacked to increase the total depth of the network. (C) The 3-D output of the last DPRNN block is converted back to a sequential output by performing overlap-add on the chunks.

Dual Path

- Propose … any kinds of …
- The intu… …sists **of interl… …l modelir…**
  - Th…
  - Th… **utterance-level (global)** processing.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

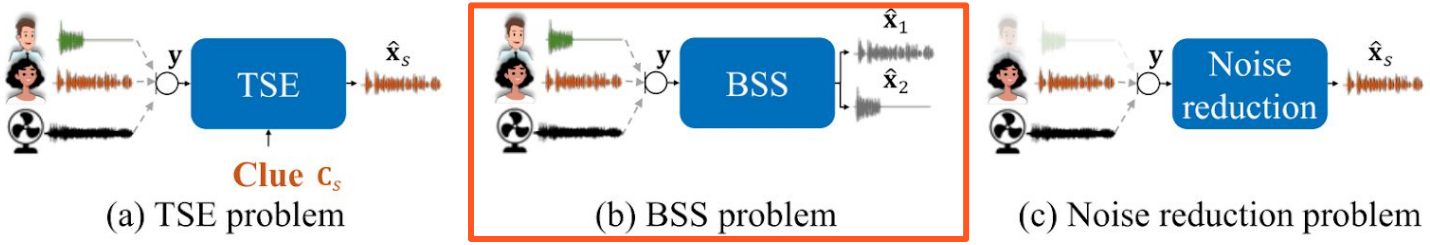


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN) - Architecture

- **Encoder** (1-D conv) is used to transform short segments of the mixture waveform into their corresponding representations in an intermediate feature space.

*Not specified in the paper

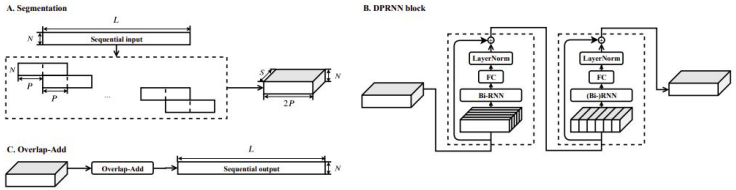


Fig. 1. System flowchart of dual-path RNN (DPRNN). (A) The segmentation stage splits a sequential input into chunks with or without overlaps and concatenates them to form a 3-D tensor. In our implementation, the overlap ratio is set to 50%. (B) Each DPRNN block consists of two RNNs that have recurrent connections in different dimensions. The *intra-chunk* bi-directional RNN is first applied to individual chunks in parallel to process local information. The *inter-chunk* RNN is then applied across the chunks to capture global dependency. Multiple blocks can be stacked to increase the total depth of the network. (C) The 3-D output of the last DPRNN block is converted back to a sequential output by performing overlap-add on the chunks.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

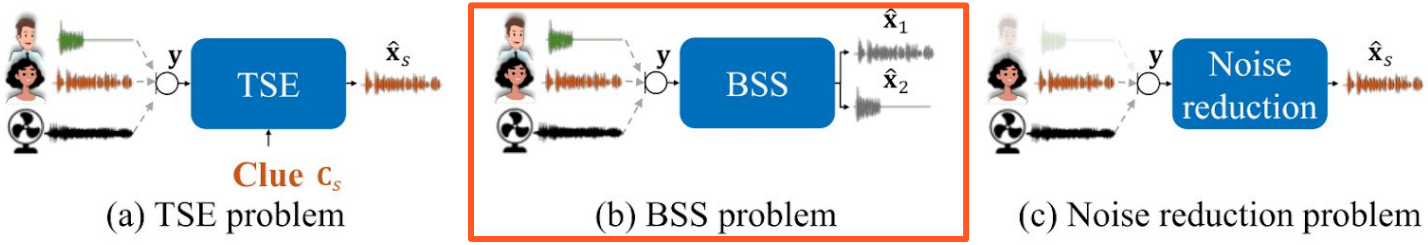


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN) - Architecture

- **Separation**: The segmentation (encoder's) output T is passed to the stack of B DPRNN blocks.
- Each block transforms an input 3-D tensor into another tensor with the same shape.
- As the intrachunk RNN is bi-directional, each time step contains the entire information of the chunk it belongs to, which allows the inter-chunk RNN to perform **fully sequence-level modeling**.

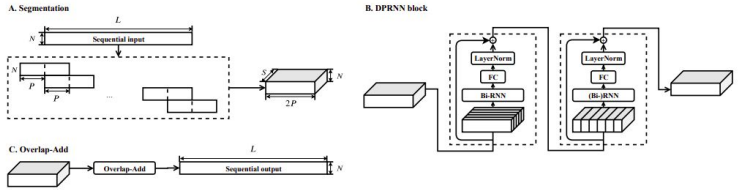


Fig. 1. System flowchart of dual-path RNN (DPRNN). (A) The segmentation stage splits a sequential input into chunks with or without overlaps and concatenates them to form a 3-D tensor. In our implementation, the overlap ratio is set to 50%. (B) Each DPRNN block consists of two RNNs that have recurrent connections in different dimensions. The *intra-chunk* bi-directional RNN is first applied to individual chunks in parallel to process local information. The *inter-chunk* RNN is then applied across the chunks to capture global dependency. Multiple blocks can be stacked to increase the total depth of the network. (C) The 3-D output of the last DPRNN block is converted back to a sequential output by performing overlap-add on the chunks.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

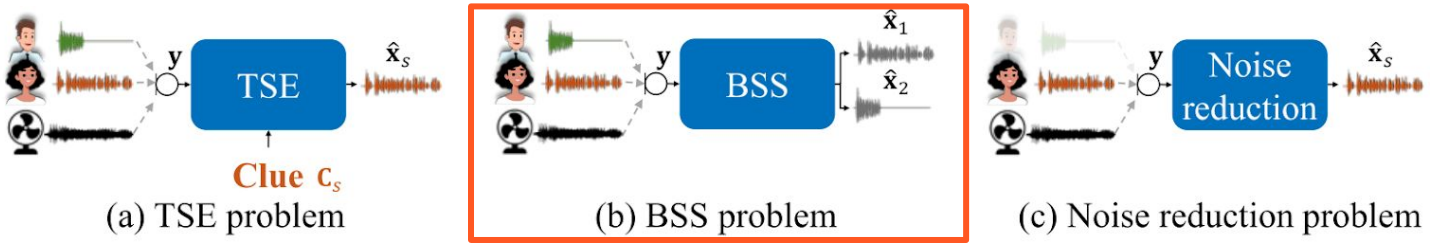


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN)- Architecture

- **Decoder**- The source waveforms are then reconstructed by transforming the masked encoder features using a decoder module. Here it is a de-conv 1-D model.

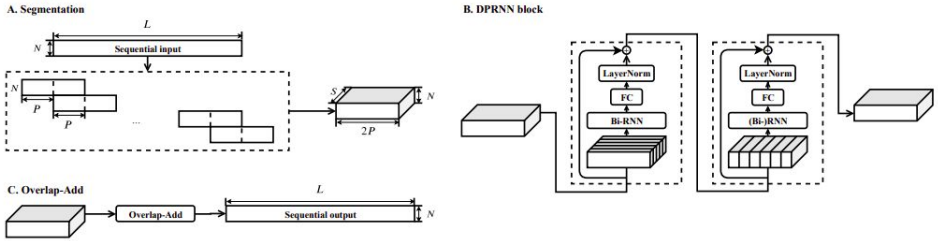


Fig. 1. System flowchart of dual-path RNN (DPRNN). (A) The segmentation stage splits a sequential input into chunks with or without overlaps and concatenates them to form a 3-D tensor. In our implementation, the overlap ratio is set to 50%. (B) Each DPRNN block consists of two RNNs that have recurrent connections in different dimensions. The *intra-chunk* bi-directional RNN is first applied to individual chunks in parallel to process local information. The *inter-chunk* RNN is then applied across the chunks to capture global dependency. Multiple blocks can be stacked to increase the total depth of the network. (C) The 3-D output of the last DPRNN block is converted back to a sequential output by performing overlap-add on the chunks.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

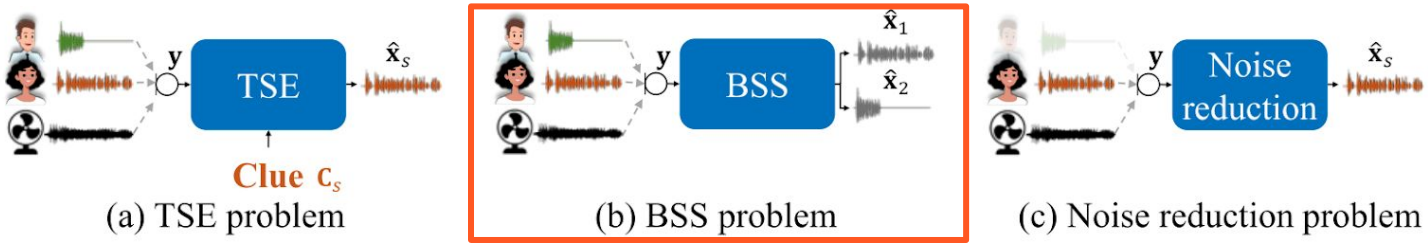


Fig. 2. Comparison of TSE with BSS and noise reduction

Dual Path RNN (DPRNN)- Architecture

- **Dataset:** WSJ0-2mix dataset
- **Objective:** All models are trained with utterance-level permutation invariant training (uPIT) to maximize scale-invariant SNR (SI-SNR).

Link to audio samples

Source 1, 2

Speech Enhancement- BSS- Methods

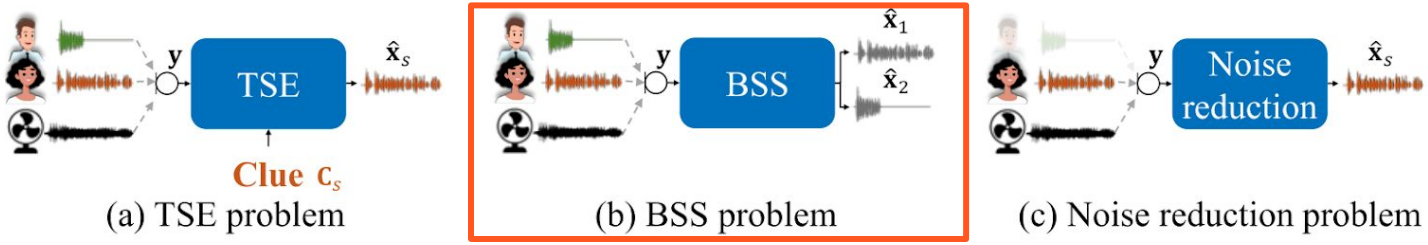


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - **GALR**
  - SepFormer
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

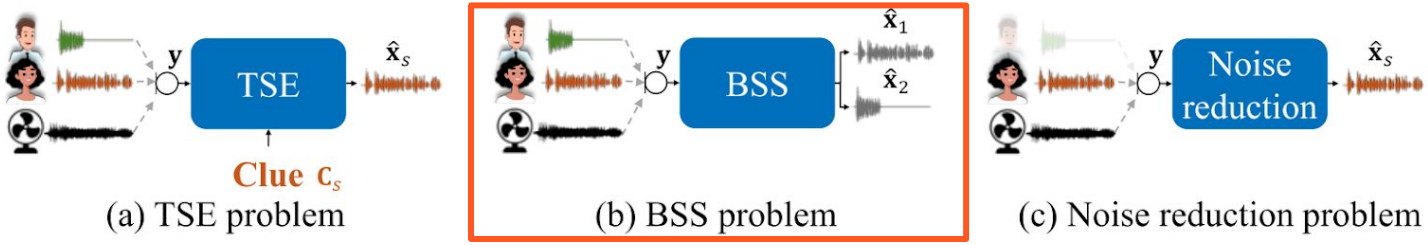


Fig. 2. Comparison of TSE with BSS and noise reduction

GALR- Architecture

EFFECTIVE LOW-COST TIME-DOMAIN AUDIO SEPARATION USING **G**LOBALLY **A**TTENTIVE **L**OCALLY **R**ECURRENT NETWORKS

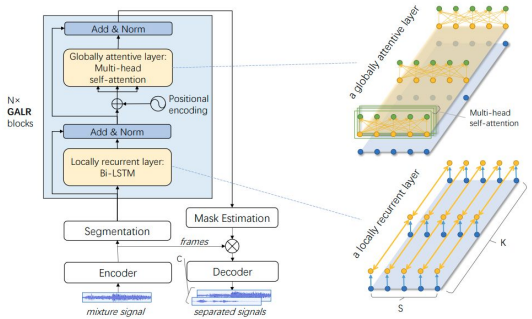


Fig. 2: Left: the overall architecture of our GALR network. Right: detailed illustration about how the intra- and inter-segment sequences are processed in the locally recurrent layer (lower right) and the globally attentive layer (upper right) inside each GALR block, respectively.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

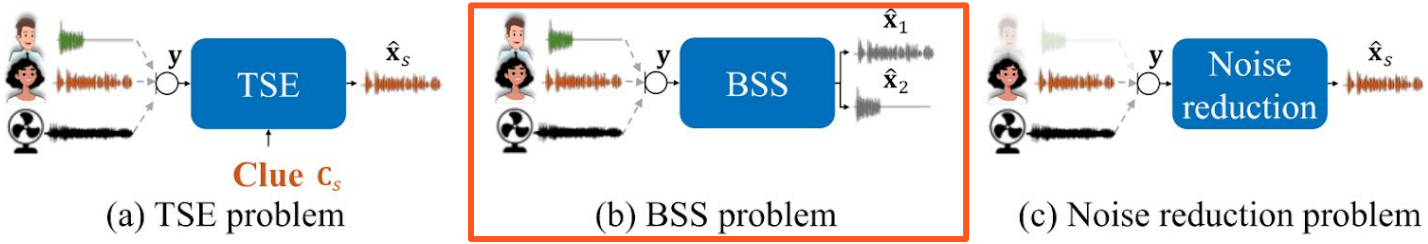


Fig. 2. Comparison of TSE with BSS and noise reduction

GALR- Architecture

- The **Separation module** combines both RNN and attention mechanisms.
  - The **RNN** is used **within** segments
  - The **attention** mechanism is used **across** segments.

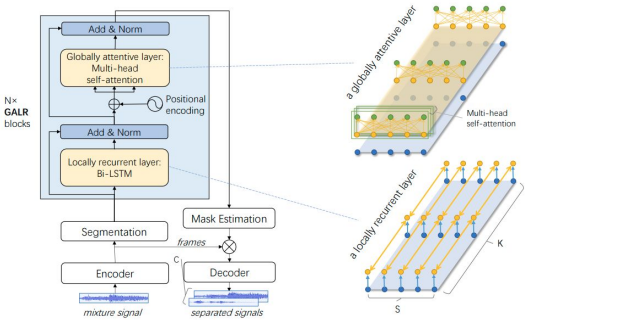


Fig. 2: Left: the overall architecture of our GALR network. Right: detailed illustration about how the intra- and inter-segment sequences are processed in the locally recurrent layer (lower right) and the globally attentive layer (upper right) inside each GALR block, respectively.

Source 1, 2

Speech Enhancement- BSS- Methods

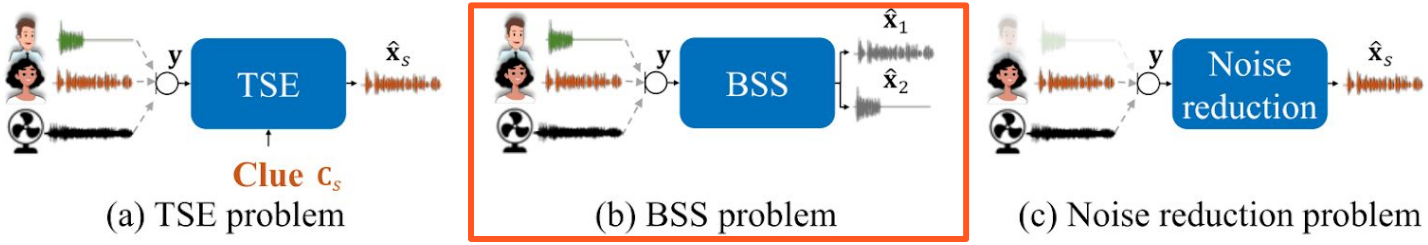


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- **NN-based:**
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - **SepFormer**
- Other:
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

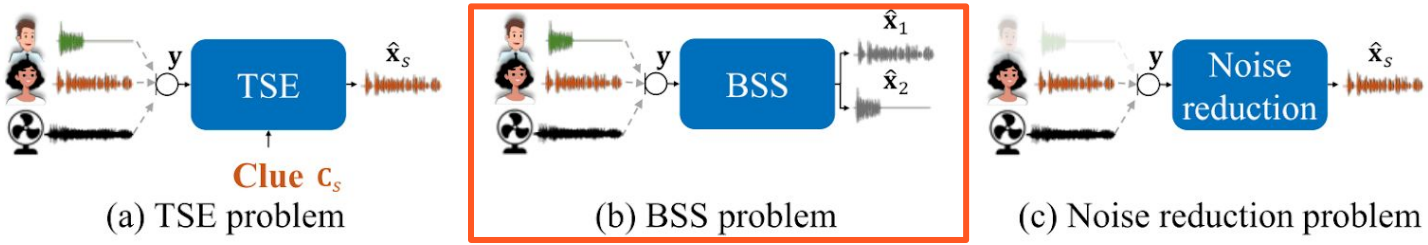


Fig. 2. Comparison of TSE with BSS and noise reduction

SepFormer

ATTENTION IS ALL YOU NEED IN SPEECH SEPARATION

Source 1, 2

Speech Enhancement- BSS- Methods- NN

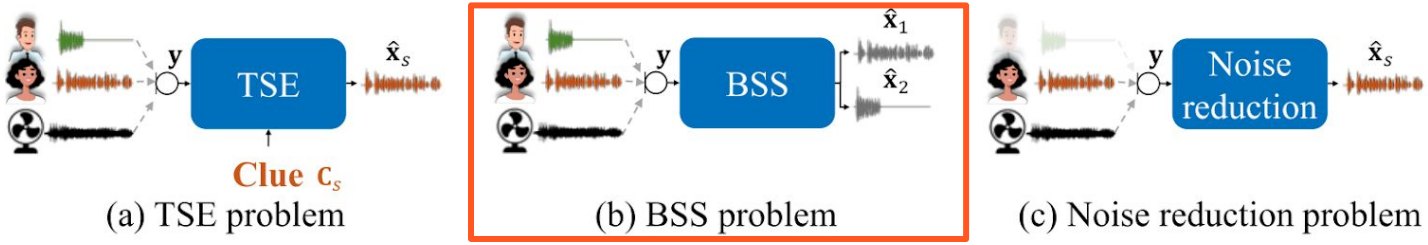


Fig. 2. Comparison of TSE with BSS and noise reduction

SepFormer

- Can be seen as a continuation to the line of work of DPRNN and GALR-
  - The **separation module** is an RNN-free Transformer-based architecture
  - Transformers learn both short and long-term dependencies.

Source 1, 2

Speech Enhancement- BSS- Methods

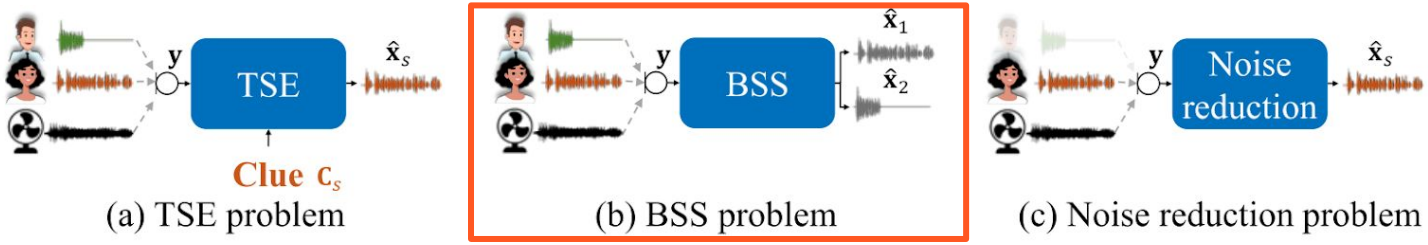


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods

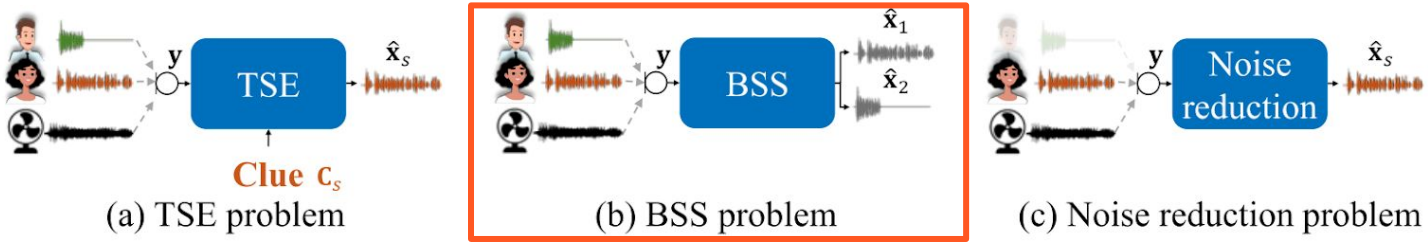


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - **Speaker embeddings**
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

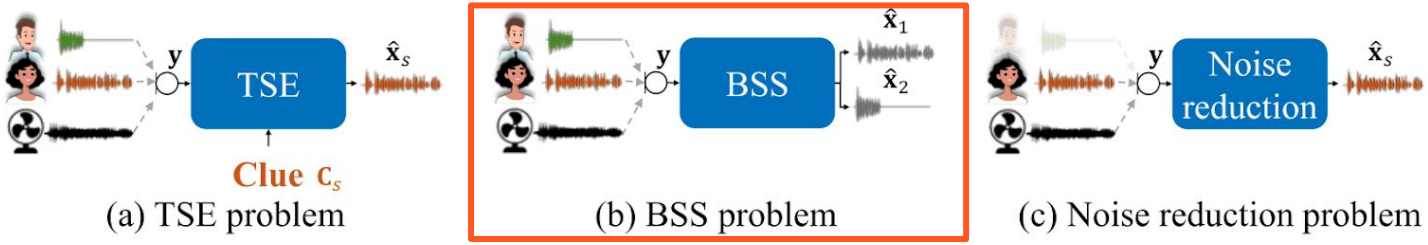


Fig. 2. Comparison of TSE with BSS and noise reduction

Incorporating Speaker Embedding- Wavesplit

Wavesplit: End-to-End Speech Separation by Speaker Clustering

- A single model that **learns jointly the speaker embeddings and separation.**
- From a single mixture, the model infers a representation for each source and then estimates each source signal given the inferred representations.
  - The model is trained to jointly perform both tasks from the raw waveform.
- Can be seen of some kind of generalization of the DeepClustering paper.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

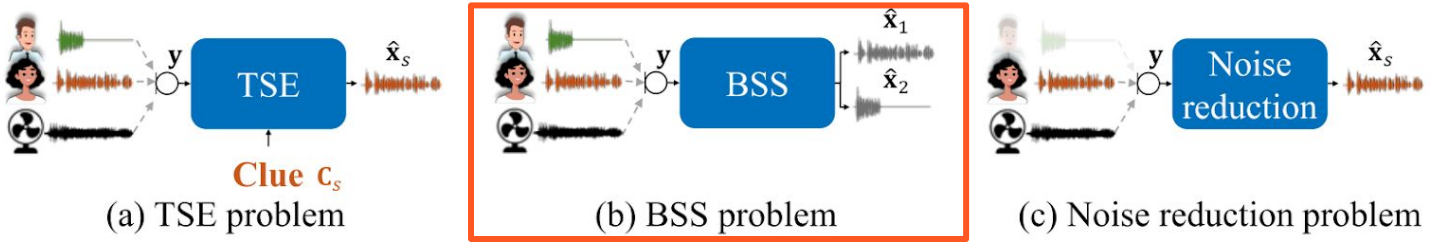


Fig. 2. Comparison of TSE with BSS and noise reduction

Incorporating Speaker Embedding- Wavesplit

Wavesplit: End-t

- A single m
- From a sin mates each sourc
  - The r
- Can be see

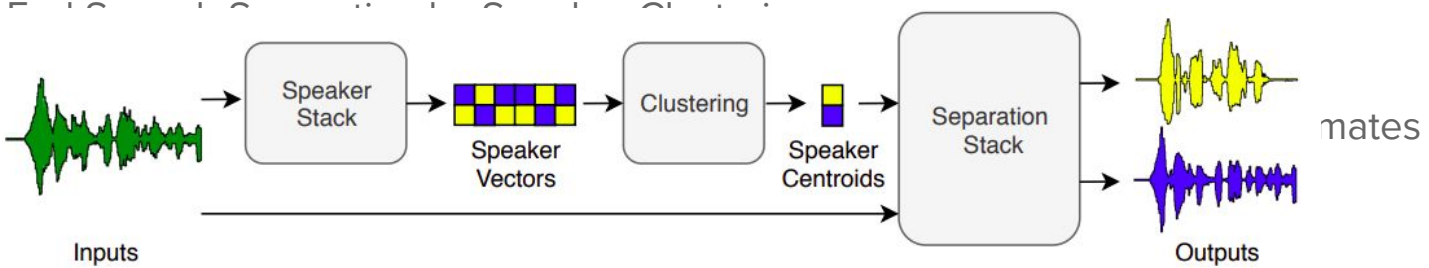


Fig. 1. Wavesplit for 2-speaker separation. The speaker stack extracts speaker vectors at each timestep. The vectors are clustered and aggregated into speaker centroids. The separation stack ingests the centroids and the input mixture to output two clean channels.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

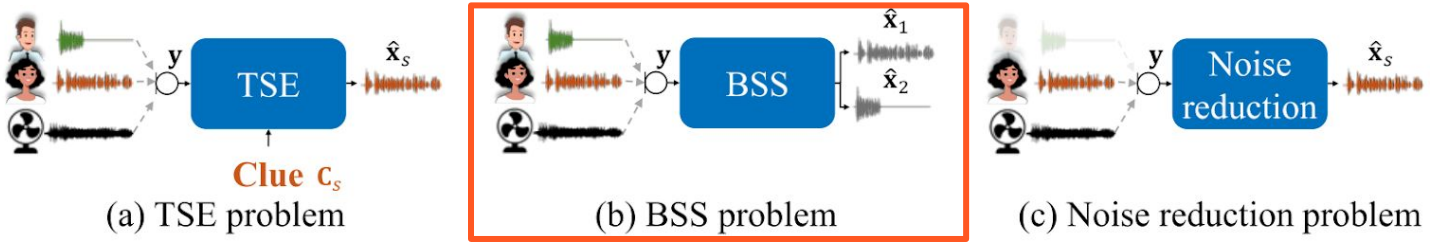
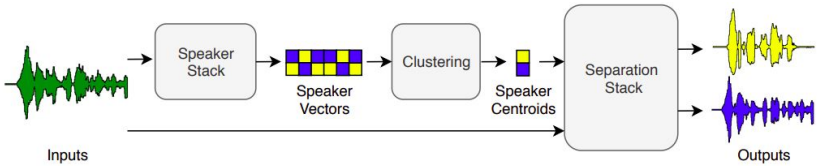


Fig. 2. Comparison of TSE with BSS and noise reduction



Incorporating Speaker Embedding- Wavesplit

Wavesplit combines two convolutional subnetworks: speaker stack and separation stack

- **Speaker stack** - maps the mixture to a set of vectors representing the recorded speakers
- **Separation stack**- consumes both the mixture and the set of speaker representations from the speaker stack. It produces a multi-channel audio output with separated speech from each speaker.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

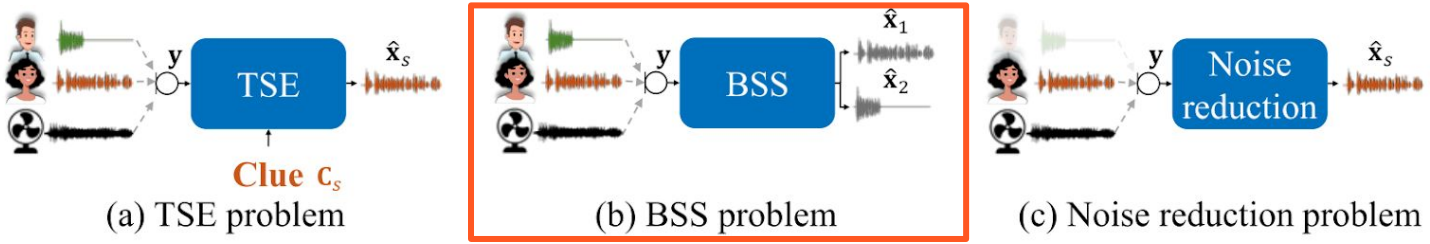
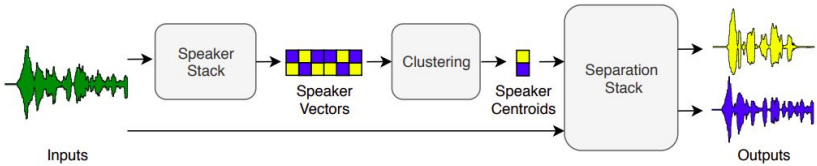


Fig. 2. Comparison of TSE with BSS and noise reduction



Incorporating Speaker Embedding- Wavesplit

Separation stack:

- consumes both the mixture and the set of speaker representations from the speaker stack. It produces a multi-channel audio output with separated speech from each speaker.
- Resembles previous architectures conditioned on pre-trained speaker vectors (such as Voice Filter- will be discussed in TSE)

Source 1, 2

Speech Enhancement- BSS- Methods- NN

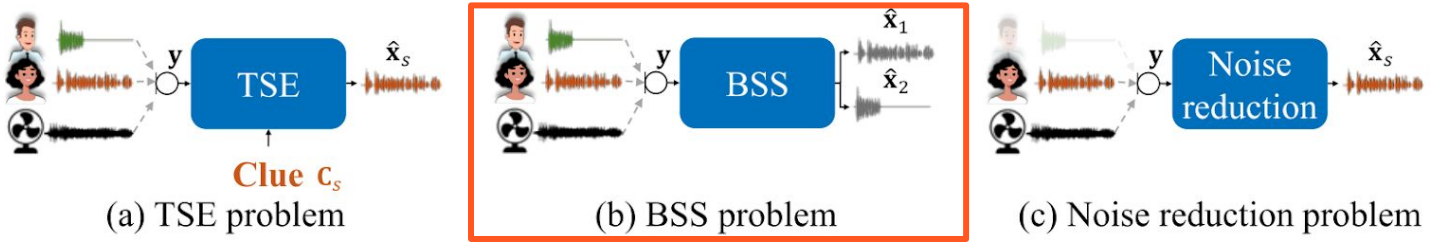
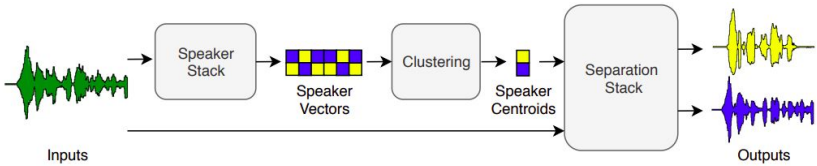


Fig. 2. Comparison of TSE with BSS and noise reduction



Incorporating Speaker Embedding- Wavesplit

Speaker stack: core contribution

- Produces *N* speaker representations of dimension *d* at each time step.
  - *N represents the **maximum number** of **simultaneous speakers** targeted by the system*
  - *It is important to note that it is **not required** to keep the **speakers' order consistently** across a sequence*
    - *E.g. a given speaker Bob could be represented by the first vector at time t and by the second vector at a different time t'*

Source 1, 2

Speech Enhancement- BSS- Methods- NN

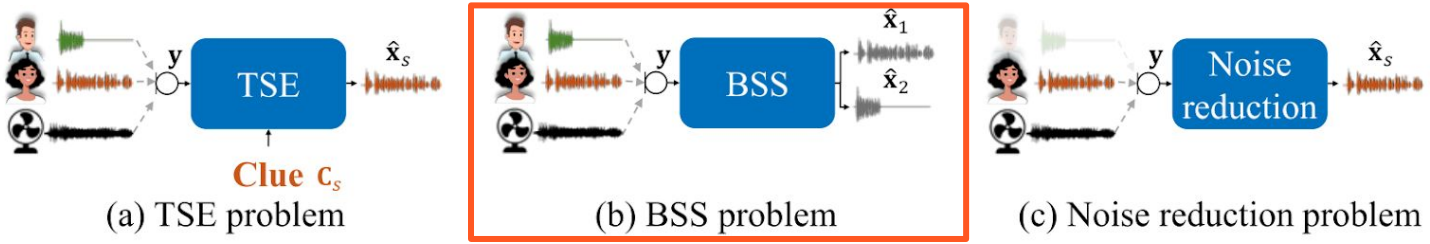
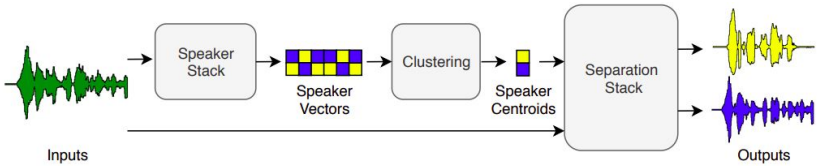


Fig. 2. Comparison of TSE with BSS and noise reduction



Incorporating Speaker Embedding- Wavesplit

Speaker stack: core contribution

- Produces *N* speaker representations of dimension *d* at each time step.
- Aggregation over the entire sequence using clustering.
  - Groups all vectors by speaker and outputs *N* summary vectors (cluster centroids) for the whole sequence.
  - K-means clustering performs this aggregation at inference and returns the centroids of the *N* identified cluster

Source 1, 2

Speech Enhancement- BSS- Methods- NN

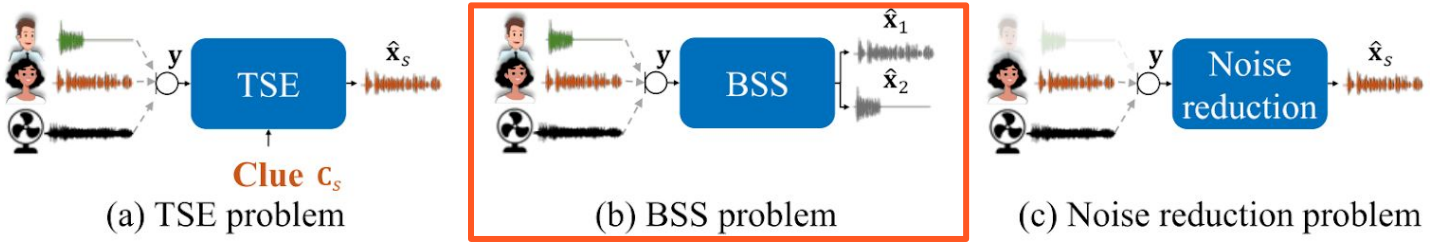
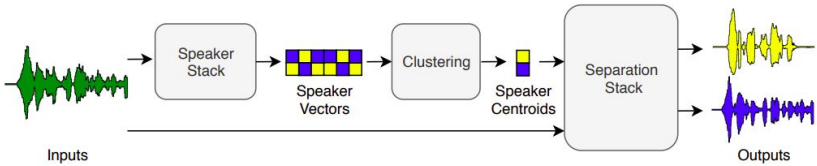


Fig. 2. Comparison of TSE with BSS and noise reduction



Incorporating Speaker Embedding- Wavesplit

- Training objective:
  - Contrastive (hinge) loss for the speaker recognition,
  - SI-SDR with PIT for the separation task.
  - Thus it encourages identifying instantaneous speaker representations such that:
    - These representations can be grouped into individual speaker clusters
    - The cluster centroids provide a long-term speaker representation for the reconstruction of individual speaker signals

Link to audio samples

Source 1, 2

Speech Enhancement- BSS- Methods

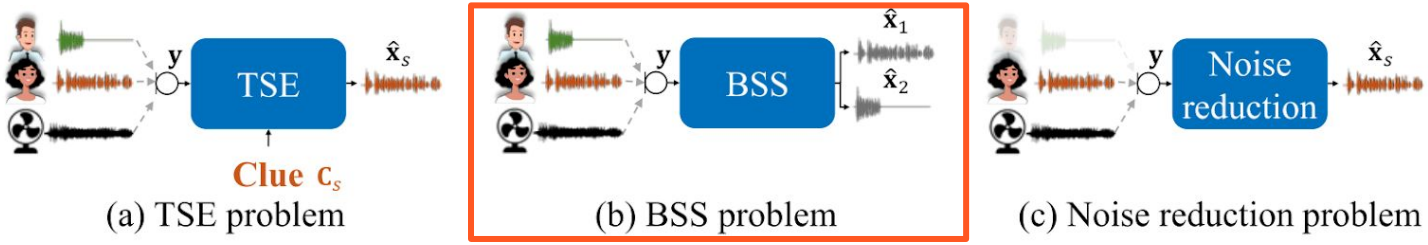


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - Speaker embeddings
  - **Estimating #speakers**
  - Iterative methods
  - GenAI
  - TF-GridNet

Image Source 208

Speech Enhancement- BSS- Methods- NN

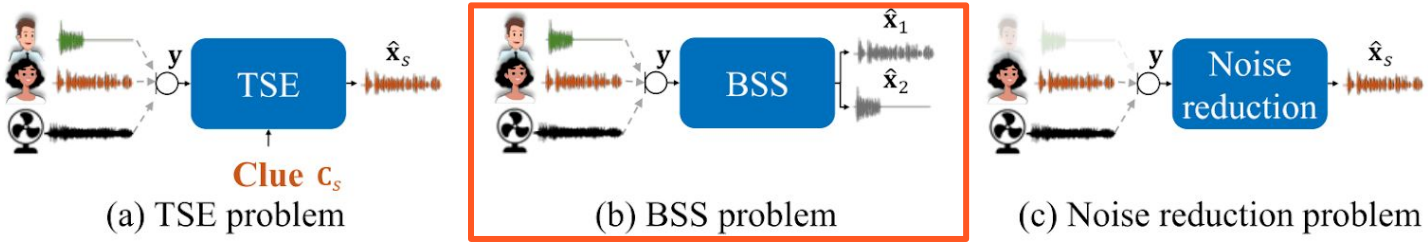


Fig. 2. Comparison of TSE with BSS and noise reduction

Estimating the number of speakers

Voice Separation with an Unknown Number of Multiple Speakers

- Employ gated neural networks that are trained to separate the voices at multiple processing steps, while **maintaining the speaker in each output channel fixed**.
- **A different model** is trained for **every number of possible speakers**, and the model with the largest number of speakers is **employed to select** the **actual number of speakers in a given sample.**
- Employ a speaker embeddings model (trains separately) for identity loss.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

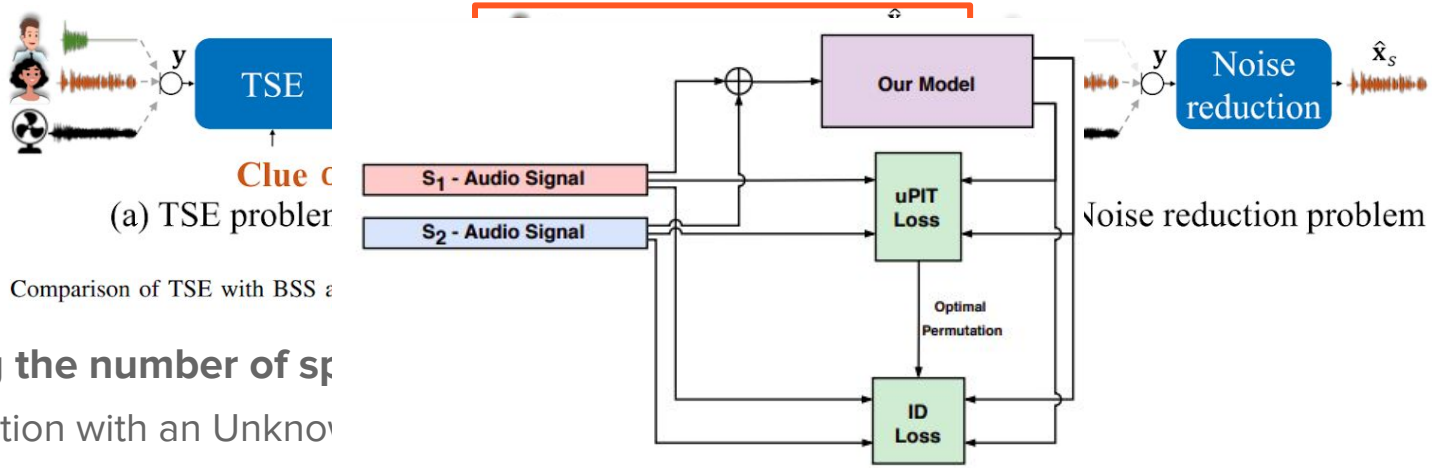


Fig. 2. Comparison of TSE with BSS a

Figure 3. The training losses used in our method, shown for the case of $C = 2$ speakers. The mixed signal x combines the two input voices s_1 and s_2. Our model then separates to create two output channels $\hat{s}_1$ and $\hat{s}_2$. The permutation invariant SI-SNR loss computes the SI-SNR between the ground truth channels and the output channels, obtained at the channel permutation π that minimizes the loss. The identity loss is then applied to the matching channels, after they have been ordered by π.

Estimating the number of sp

Voice Separation with an Unkno

- Employ gated neural netwo at multiple processing steps, while **maintaining the spea**
- **A different model** is traine nd the model with the largest number of speakers **of speakers in a given sample.**
- Employ a speaker embedd ss.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

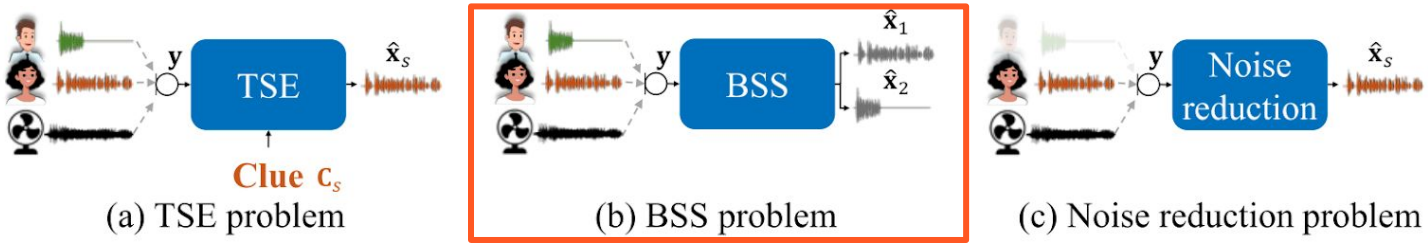


Fig. 2. Comparison of TSE with BSS and noise reduction

Estimating the number of speakers - Identity loss

- The outputs are given in a permutation invariant fashion
- Hence, voices can switch between output-channels, especially during transient silence episodes.
- To tackle this, they propose a new loss that is based on a voice representation network that is trained on the same training set.
- The embedding obtained by this network is then used to compare the output voice to the voice of the output channel.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

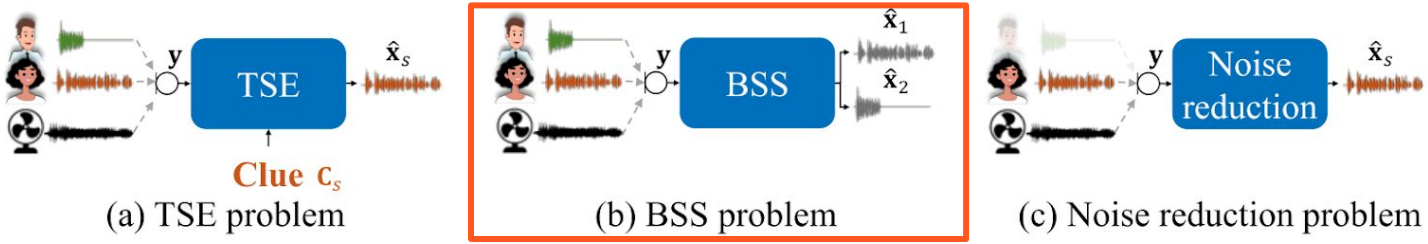


Fig. 2. Comparison of TSE with BSS and noise reduction

Estimating the number of speakers - Training details

- Datasets:
  - WSJ0-2mix and WSJ0-3mix datasets
  - Further expand the WSJ-mix dataset to four and five speakers
    - introduce WSJ0-4mix and WSJ0-5mix datasets- applied the same procedure as in WSJ0-2mix

Link to audio samples

Source 1, 2

Speech Enhancement- BSS- Methods

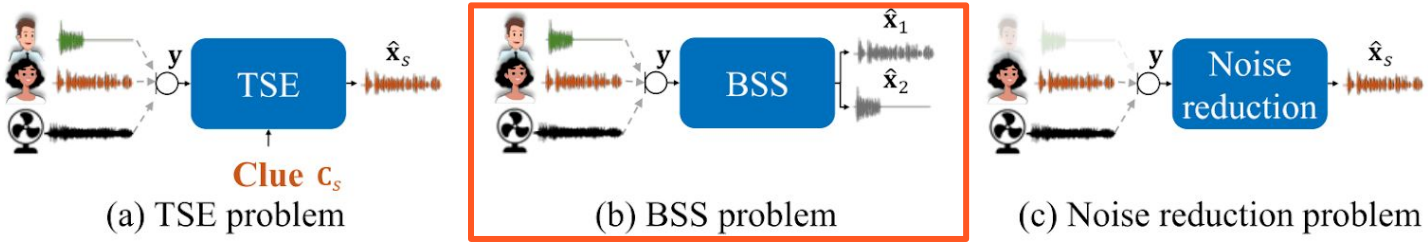


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - Speaker embeddings
  - Estimating #speakers
  - **Iterative methods**
  - GenAI
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

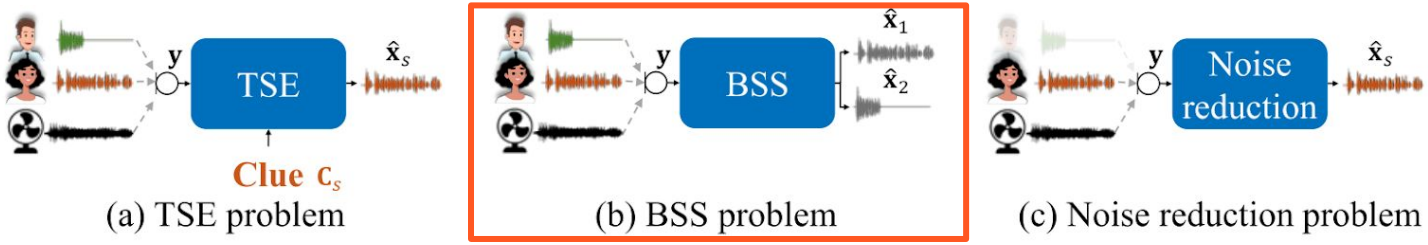


Fig. 2. Comparison of TSE with BSS and noise reduction

Iterative methods - SepIt

SepIt: Approaching a Single Channel Speech Separation Bound

- Sepit iteratively improves the different speakers' estimation
- Consists of sequential processing of the estimated signals, where each iteration contains a replica of the basic model.
  - An iterative encoder-mask-decoder procedure is applied for refining the separated signals.

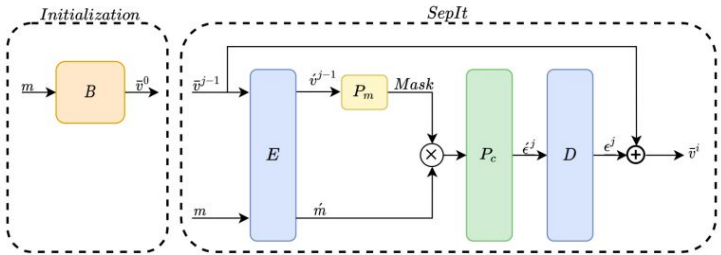


Figure 1: *The SepIt block architecture.*

Source 1, 2

Speech Enhancement- BSS- Methods

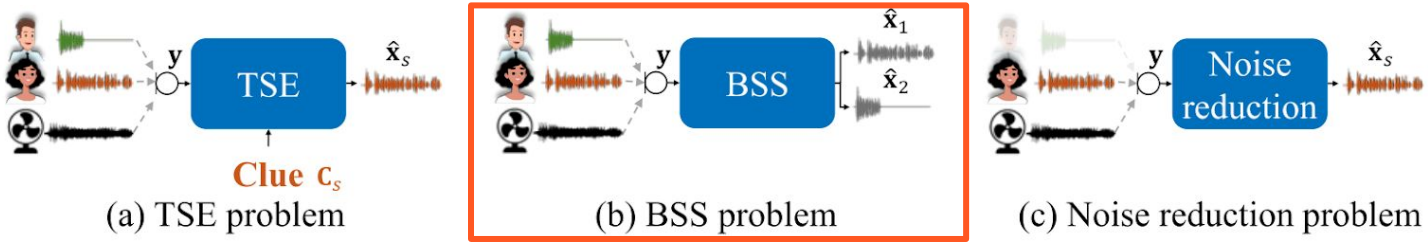


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - **GenAI**
  - TF-GridNet

Image Source

Speech Enhancement- BSS- Methods- NN

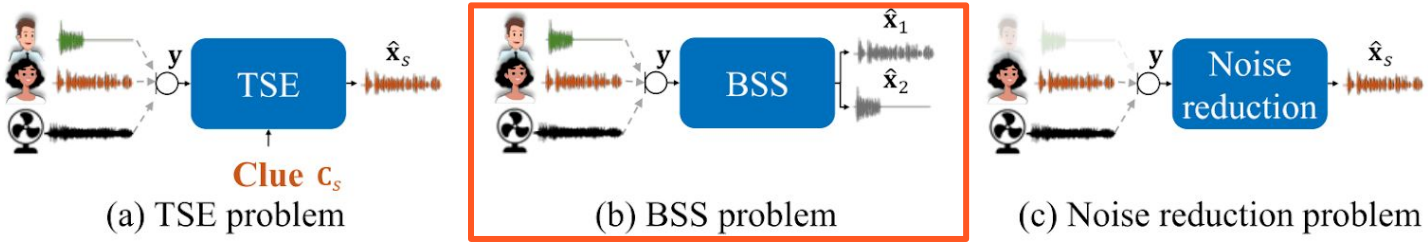


Fig. 2. Comparison of TSE with BSS and noise reduction

GenAI: Separate And Diffuse

Separate And Diffuse: Using a Pretrained Diffusion Model for Improving Source Separation

1 Introduction

Here is a simple algorithm for improving source separation. Given a test mixture m of C speakers, (1) apply a deep neural architecture B to m and obtain multiple approximated sources $\bar{v}_d^i$ for $i = 1...C$, (2) apply a generative diffusion model GM using each of the approximations obtaining $\bar{v}_g^i$, and (3) apply a shallow convolutional neural network F to $\bar{v}_d^i$ and $\bar{v}_g^i$ to obtain mixing weights $[\alpha_i, \beta_i] = F(\bar{v}_d^i, \bar{v}_g^i)$ and combine the two approximations linearly in the frequency domain to obtain the output $\bar{v}^i$.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

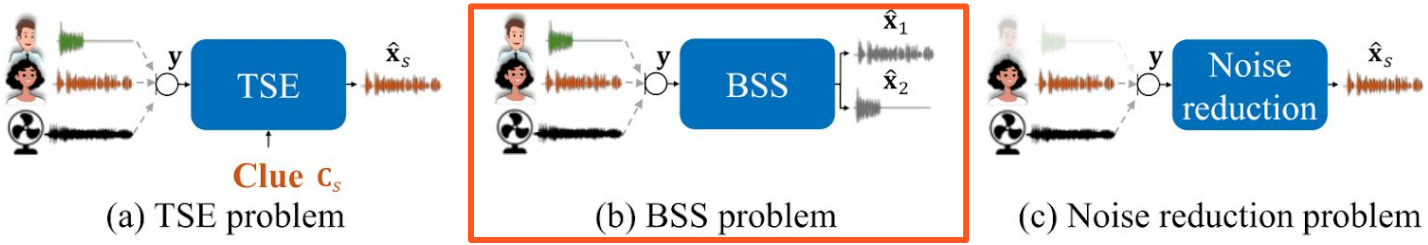


Fig. 2. Comparison of TSE with BSS and noise reduction

GenAI: Separate And Diffuse

- Train only network F out of the three networks (B, GM, F)
- The other networks are taken, as is, from published models.
  - Specifically, B is either Gated-LSTM or SepFormer
  - GM is the DiffWave network
- Achieve promising results on separation of 10 and 20 speakers.

Source 1, 2

Speech Enhancement- BSS- Methods

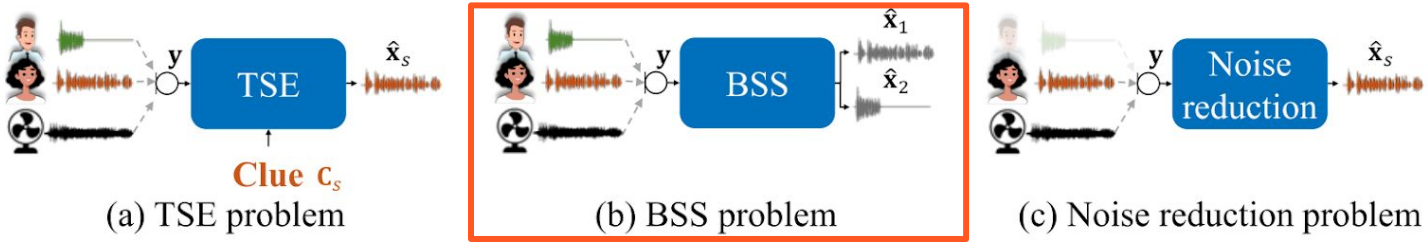


Fig. 2. Comparison of TSE with BSS and noise reduction

- Traditional:
  - Single mic: Non-Negative Matrix Factorization (NMF)
  - Multi mic: Independent component analysis (ICA)
- NN-based:
  - Deep Clustering
  - Permutation Invariant loss
  - TAS-NET
  - Conv-TASNET
  - Dual Path RNN
  - GALR
  - SepFormer
- **Other:**
  - Speaker embeddings
  - Estimating #speakers
  - Iterative methods
  - GenAI
  - **TF-GridNet**

Image Source

Speech Enhancement- BSS- Methods- NN

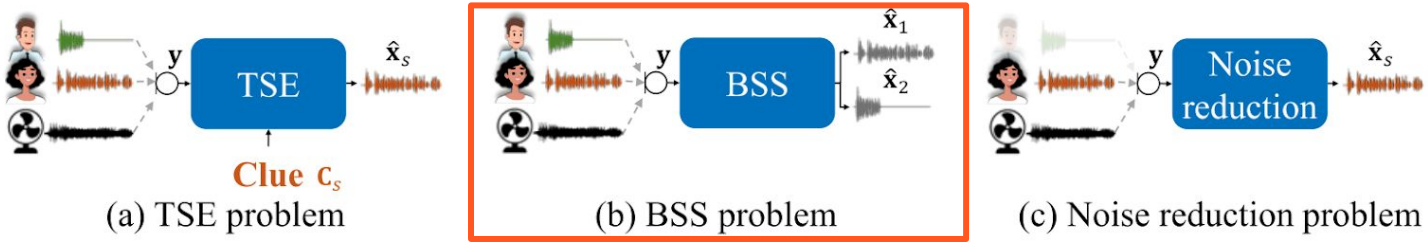


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet:

- TF-GridNet is a novel multi-path deep neural network (DNN) operating in the **time-frequency (T-F) domain**, for **monaural** talker-independent speaker separation in **anechoic conditions**.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

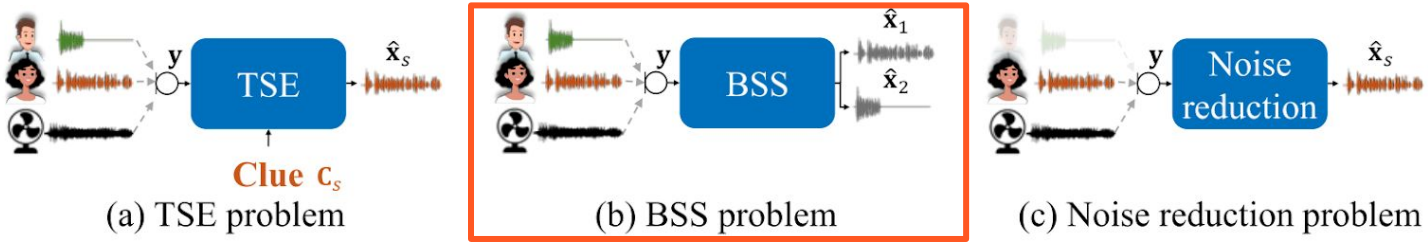


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet:

- TF-GridNet is a novel multi-path deep neural network (DNN) operating in the **time-frequency (T-F) domain**, for **monaural** talker-independent speaker separation in **anechoic conditions**.
- The model **stacks several multi-path blocks**,

Source 1, 2

Speech Enhancement- BSS- Methods- NN

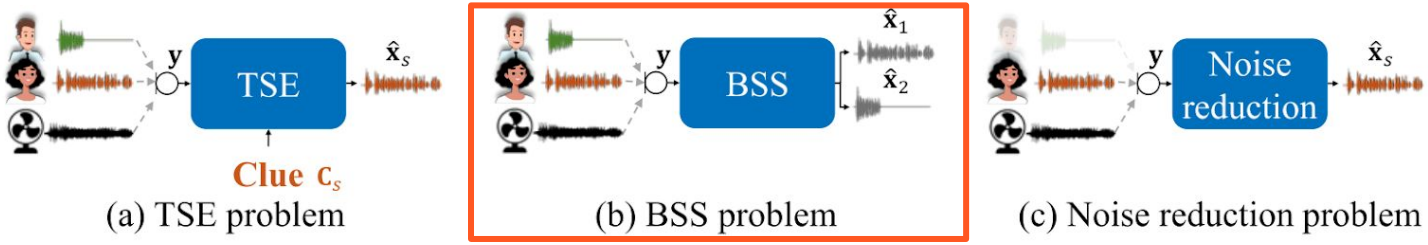


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet:

- TF-GridNet is a novel multi-path deep neural network (DNN) operating in the **time-frequency (T-F) domain**, for **monaural** talker-independent speaker separation in **anechoic conditions**.
- The model **stacks several multi-path blocks**,
- Each multi-path block consists of three modules to leverage **local and global spectro-temporal** (hence the term TF-GridNet) information for separation:
  - Intraframe spectral module
  - Sub-band temporal module
  - Full-band self-attention module.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

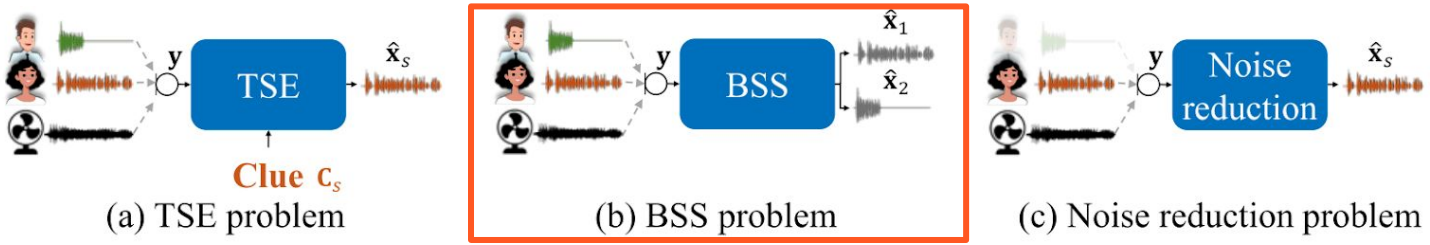


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet:

- The model is trained to perform **complex spectral mapping**:
  - Real and imaginary (RI) components of the input mixture are stacked as input features
  - Model predicts target RI components (mapping, not masking!).

Source 1, 2

Speech Enhancement- BSS- Methods- NN

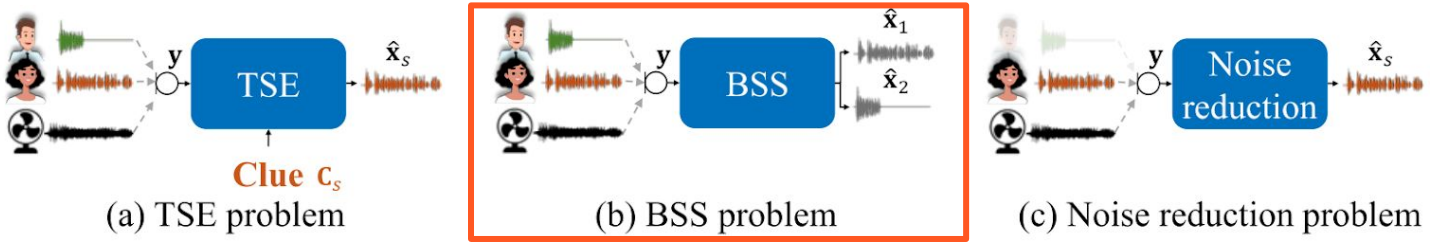


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet:

- The model is trained to per
  - Real and imaginary (R stacked as input features
  - Model predicts target).

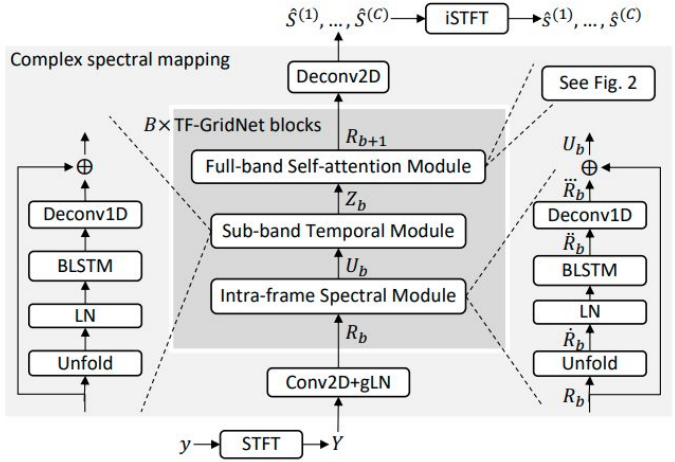


Fig. 1: Overview of proposed system.

Source 1, 2

Speech Enhancement- BSS- Methods- NN

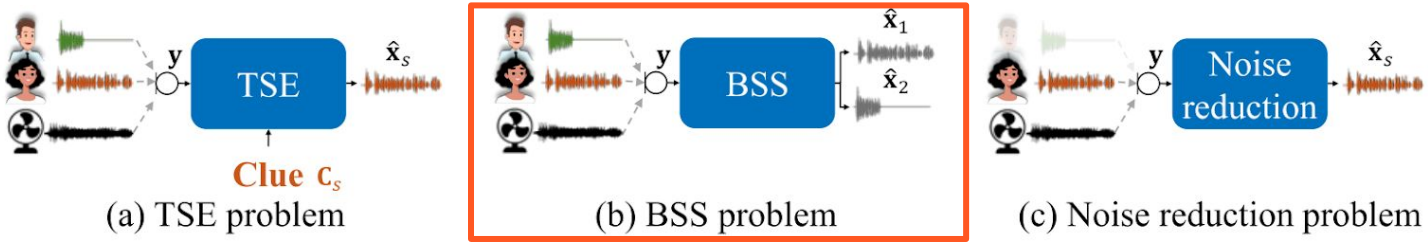


Fig. 2. Comparison of TSE with BSS and noise reduction

TF-GridNet: Results

- Obtain 23.4 dB SI-SDR improvement (SI-SDRi) on the WSJ0-2mix dataset, outperforming the previous best by a large margin.

Source 1, 2

Speech Enhancement- TSE

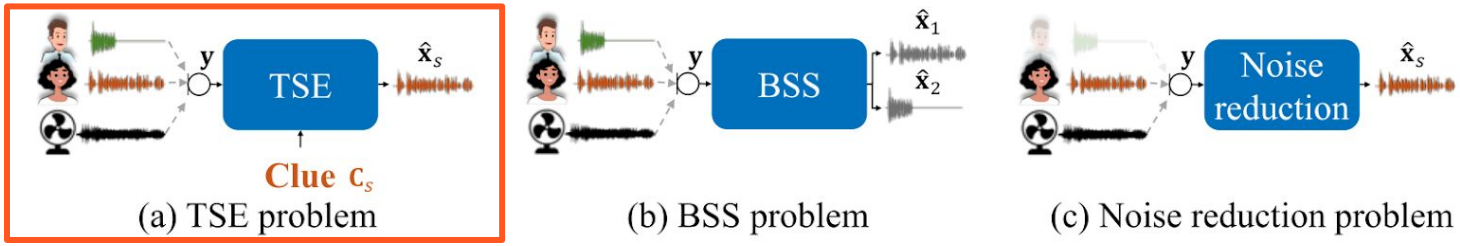


Fig. 2. Comparison of TSE with BSS and noise reduction

Roughly speaking, there are three main tasks in this domain:

- Noise reduction.
- Blind source separation.
- **Target speech enhancement.**

Follow this great paper

Image Source 225

Speech Enhancement- TSE- Overview

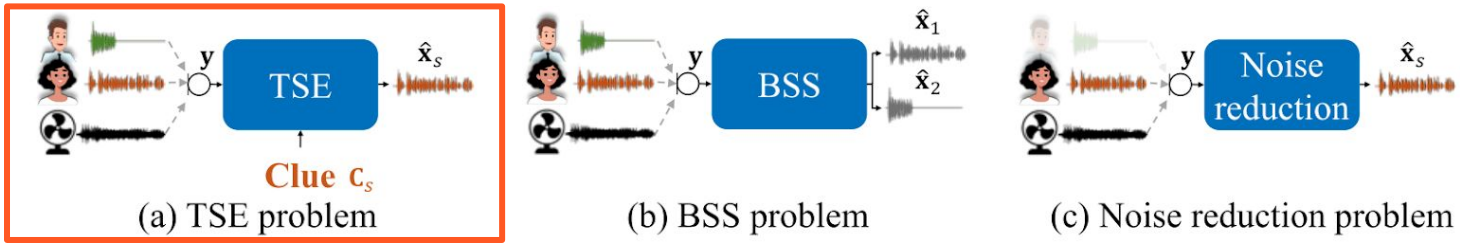


Fig. 2. Comparison of TSE with BSS and noise reduction

Overview

- Humans can listen to a target speaker even in challenging acoustic conditions that have noise, reverberation, and interfering speakers.
- Solving the “cocktail-party problem” has been the holy grail for decades.
- The interfering speakers form an ‘informative masking’ as their voices have similar characteristics/overlap the target speakers’ frequencies.

Source 1

Speech Enhancement- TSE- Overview

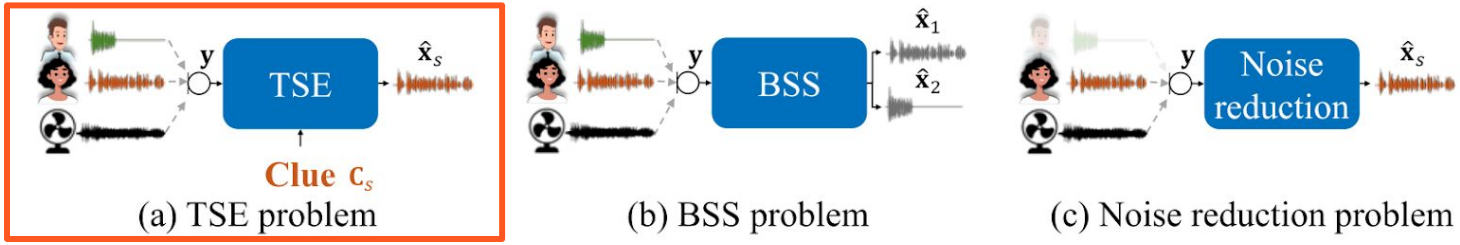


Fig. 2. Comparison of TSE with BSS and noise reduction

Overview

- many studies have identified essential cues exploited by humans to attend to a target speaker in a speech mixture:
  - Spatial
  - Spectral
  - Visual
  - Semantic
  - etc.

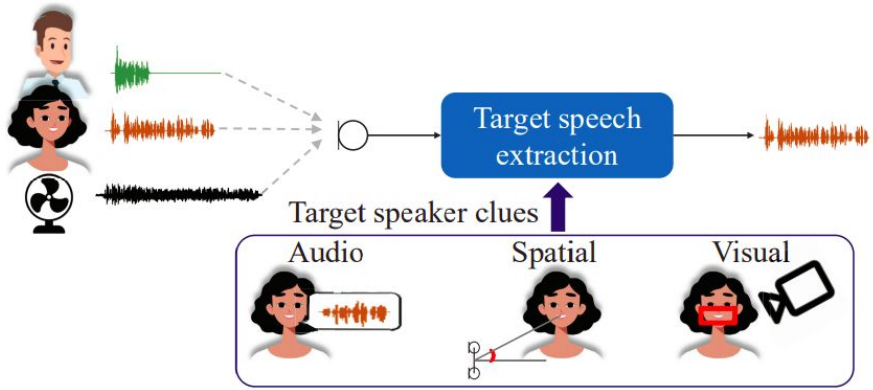


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- Overview

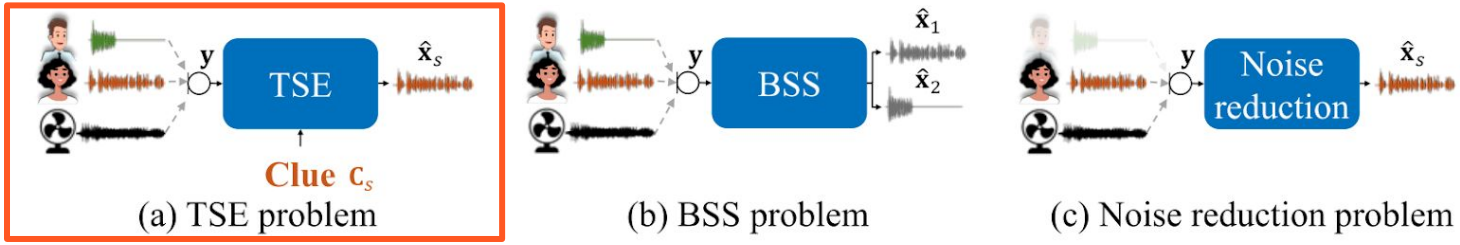


Fig. 2. Comparison of TSE with BSS and noise reduction

Overview

- Unlike BSS, TSE focuses on the target speaker's speech by exploiting clues without assuming knowledge of the number of speakers in the mixture and avoids global permutation ambiguity.
- It thus offers a practical alternative to noise reduction or BSS when the use case requires focusing on a desired speaker's voice.

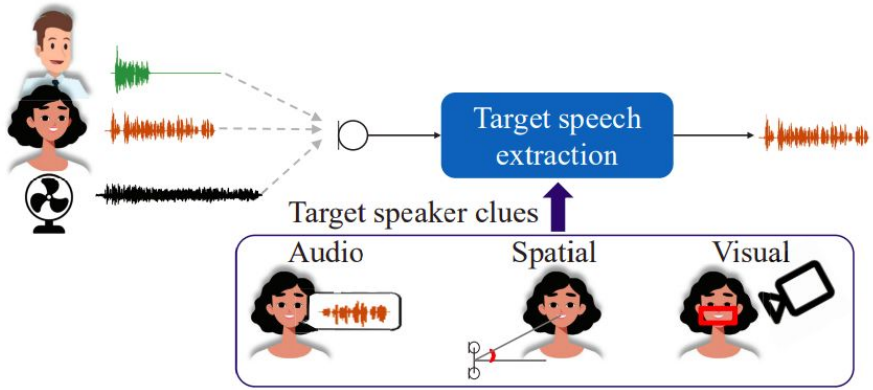


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- Problem Formulation

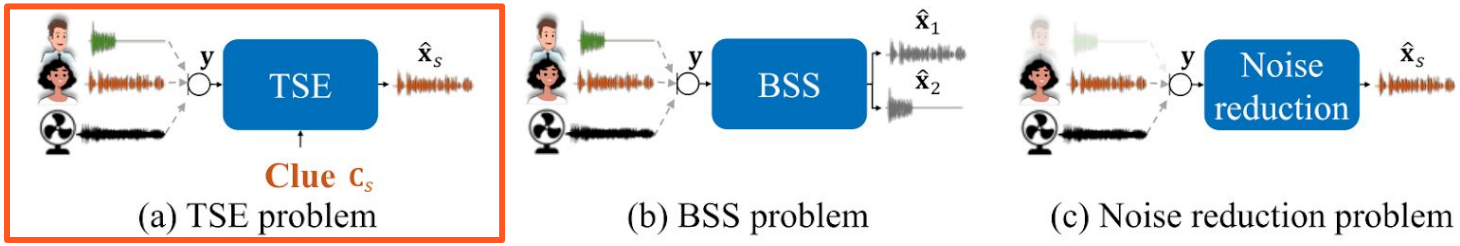


Fig. 2. Comparison of TSE with BSS and noise reduction

We can express the mixture signal recorded at a microphone as

$$\mathbf{y}^m = \mathbf{x}_s^m + \sum_{k \neq s} \mathbf{x}_k^m + \mathbf{v}^m,$$

Where:

1. $y^m \in R^T$ is the time-domain signal of the mixture (acquired audio)
2. $x_s^m \in R^T$ is the target speech
3. $x_k^m \in R^T$ is the interference speech (non-target/s speech)
4. $v_m \in R^T$ is the noise signal
5. Variable *T* represents the duration (number of samples) of the signals, *m* is the index of the microphone in an array of microphones, *s* represents the index of the target speaker and *k* is the index for the other speech sources.

Source 1

Speech Enhancement- TSE- Problem Formulation

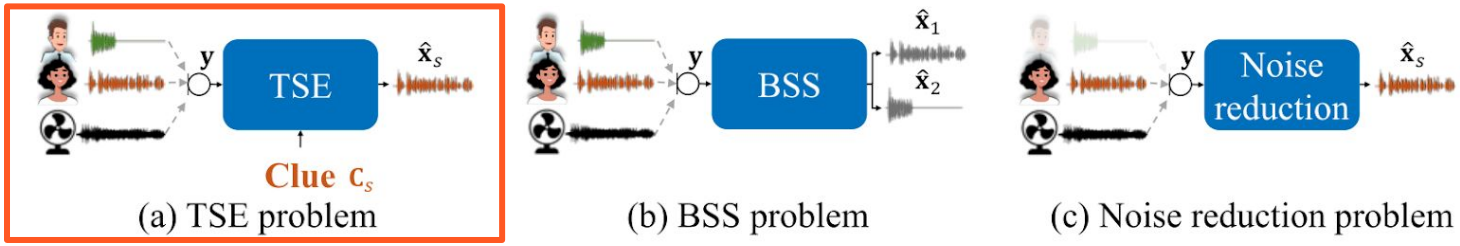


Fig. 2. Comparison of TSE with BSS and noise reduction

Target Speech Enhancement (TSE) estimates the target speech, given a clue, C_s, as

$$\hat{x}_s = TSE(y, C_s; \theta_{TSE}),$$

where $\hat{x}_s$ is the estimate of the target speech, $TSE(\cdot; \theta_{TSE})$ represents a TSE system with parameters θ_{TSE} .

Source 1

Speech Enhancement- TSE- Clue Types

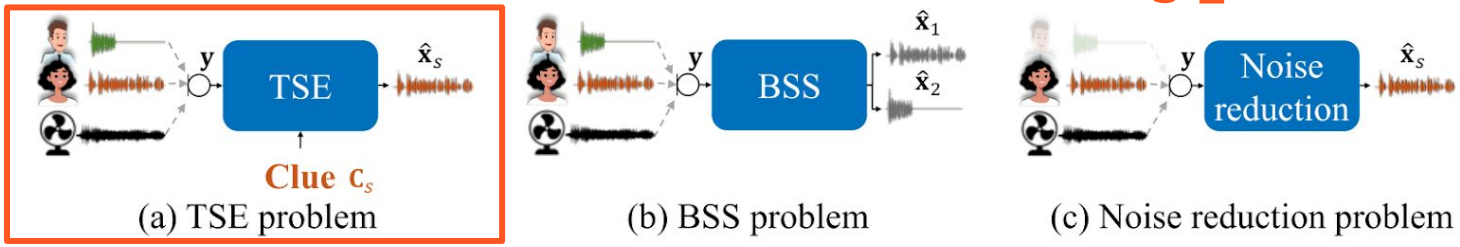


Fig. 2. Comparison of TSE with BSS and noise reduction

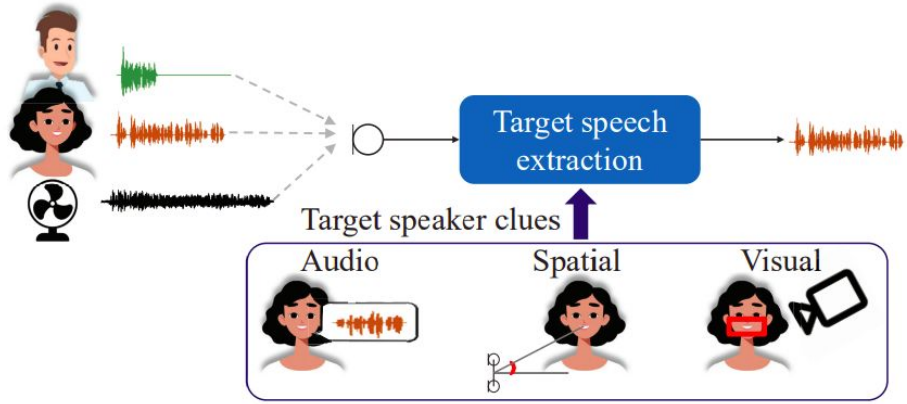


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- Clue Types

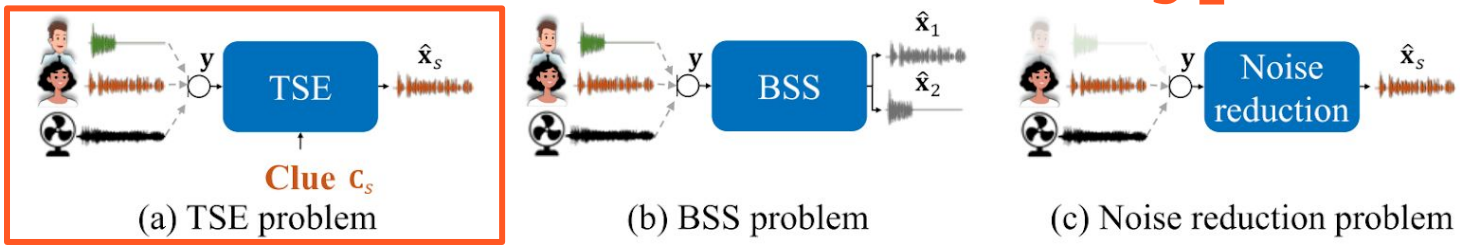


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Audio Clues:**
  - Consists of a recording of a speech signal of the target speaker.
  - Audio clues are perhaps the most easy to capture, however, the performance may be limited compared to other clues.

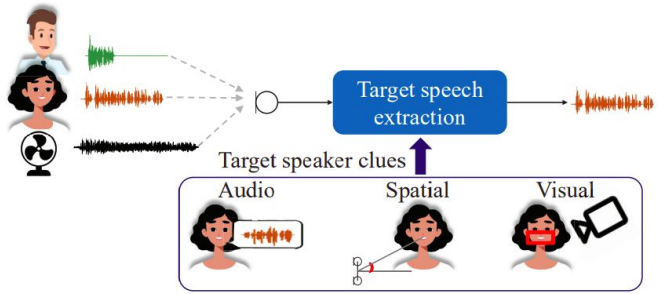


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- Clue Types

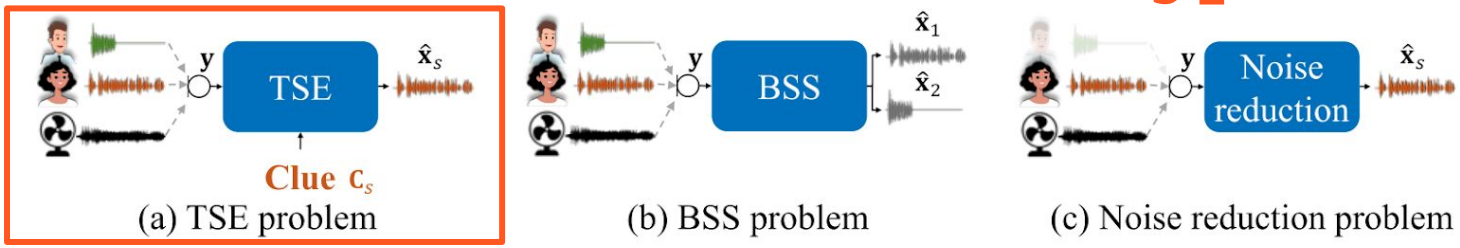


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Audio Clues:**
- **Visual Clues:**
  - Consists of a video of the target speaker talking (lip area).
  - Visual clues are typically synchronized with audio signals that are processed
  - Useful mainly when speakers in the recording have similar voices.

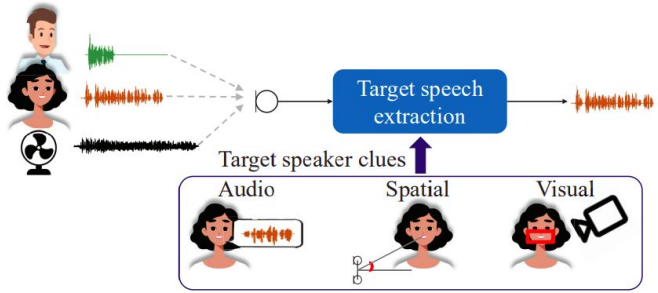


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- Clue Types

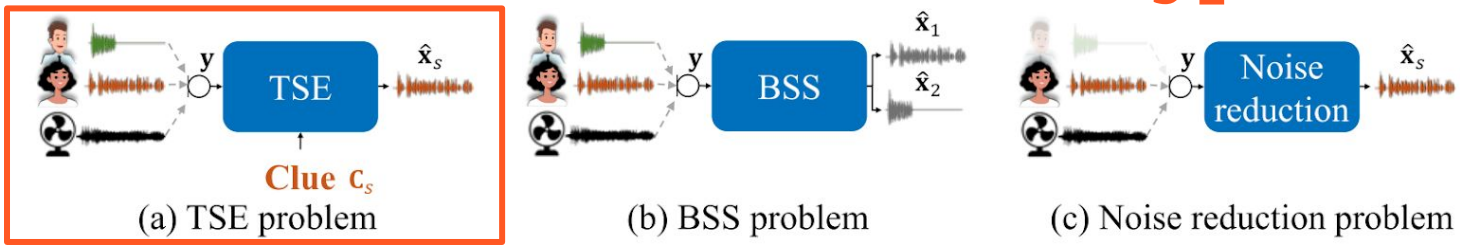


Fig. 2. Comparison of TSE with BSS and noise reduction

- **Audio Clues:**
- **Visual Clues:**
- **Spatial Clues:**
  - Refers to the target speaker's location, e.g., the angle from the recording devices.
  - Applicable only when a recording from a microphone array.
  - Can identify the target speaker reliably

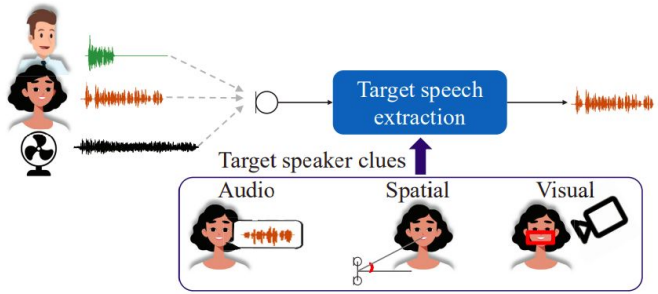


Fig. 1. TSE problem and examples of clues

Source 1

Speech Enhancement- TSE- General Framework

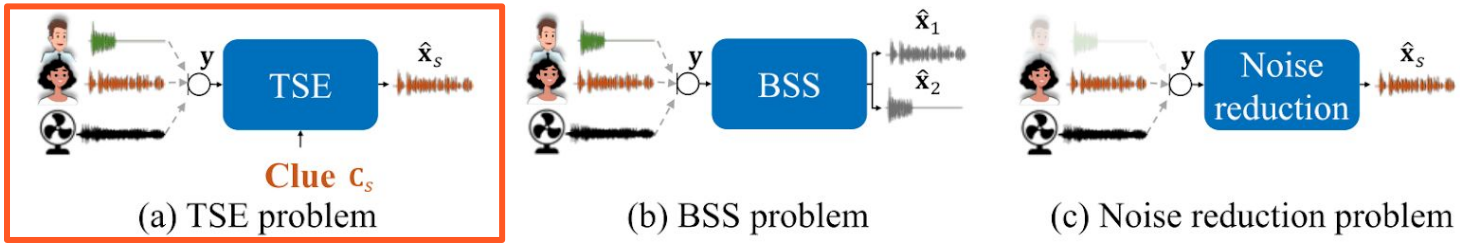


Fig. 2. Comparison of TSE with BSS and noise reduction

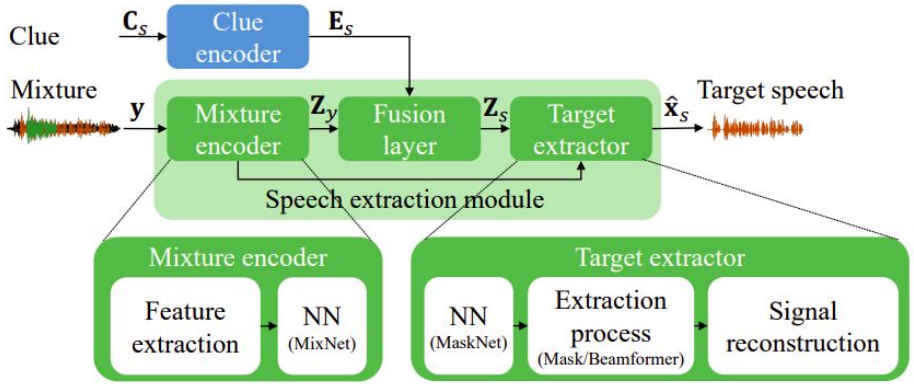


Fig. 3. General framework for neural TSE

A neural TSE system consists of an NN that estimates the target speech conditioned on a clue.

Source 1

Speech Enhancement- TSE- General Framework

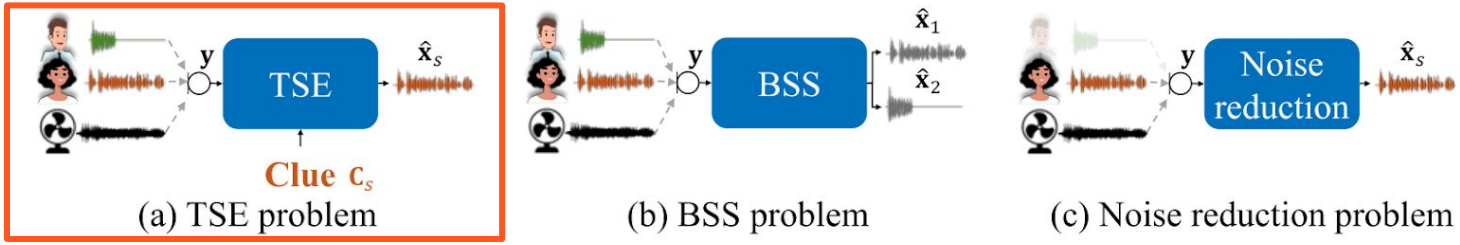
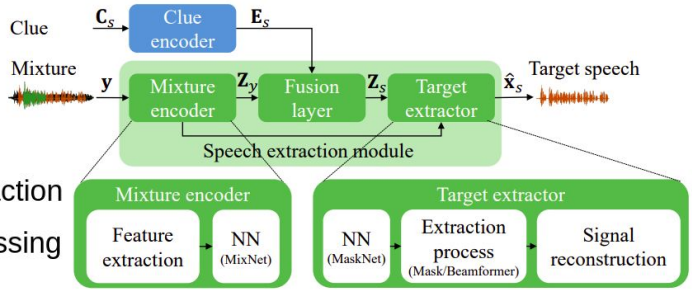


Fig. 2. Comparison of TSE with BSS and noise reduction

Clue Encoder



The **clue encoder** extracts from the clue C_s information that allows the speech extraction module to identify and extract the target speech in the mixture. We can express the processing as

$$E_s = ClueEncoder(C_s; \theta_{Clue}),$$

where $ClueEncoder(\cdot; \theta_{Clue})$ represents a Clue Encoder with parameters θ_{Clue}, and E_s are the clue embeddings.

Source 1

Speech Enhancement- TSE- General Framework

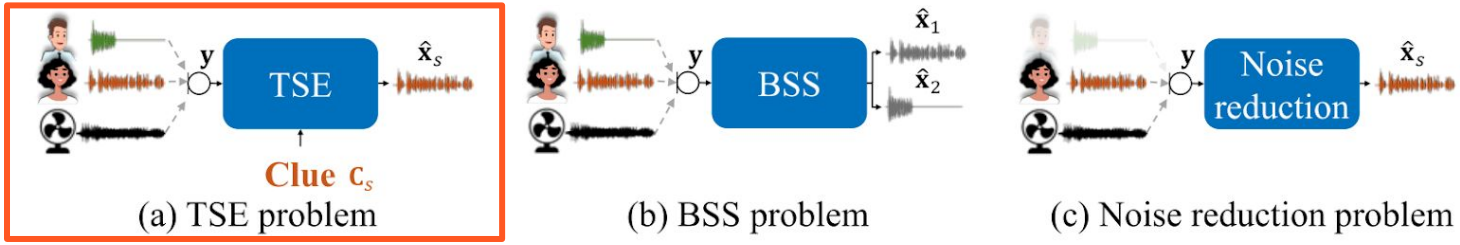
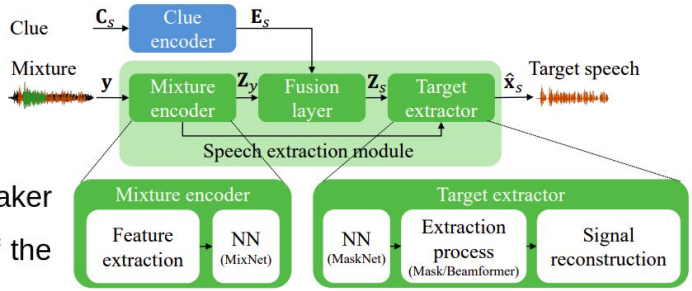


Fig. 2. Comparison of TSE with BSS and noise reduction

Clue Encoder

The dimensionality of the clue depends on the modality/clue type:

- An enrollment utterance will be processed to form $E_s = E_s^{(a)} \in R^{D_{Emb}}$, a speaker embedding vector of dimension D_{Emb} that **represents** the **voice characteristics** of the target speaker.
- Visual clues, $E_s = E_s^{(v)} \in R^{D_{Emb} xN}$ can be a sequence of the embeddings of length *N*, representing, e.g., the lip movements of the target speaker.
    - For audio clue-based TSE systems we can repeat the speaker embedding vector for each time frame to form a sequence of embeddings.



Source 1

Speech Enhancement- TSE- General Framework

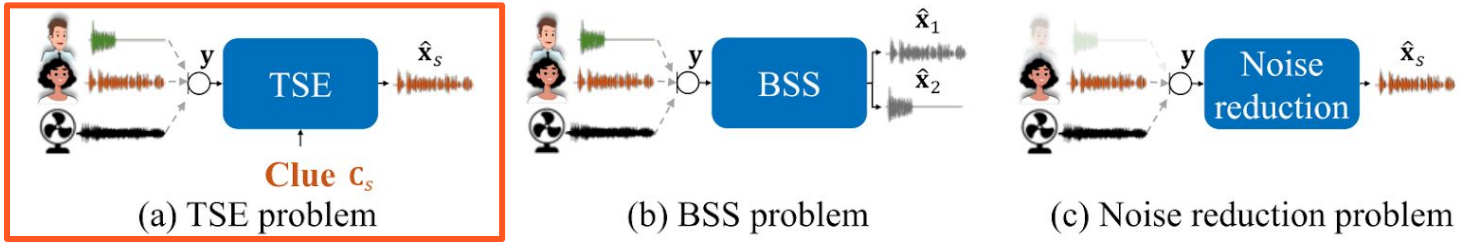
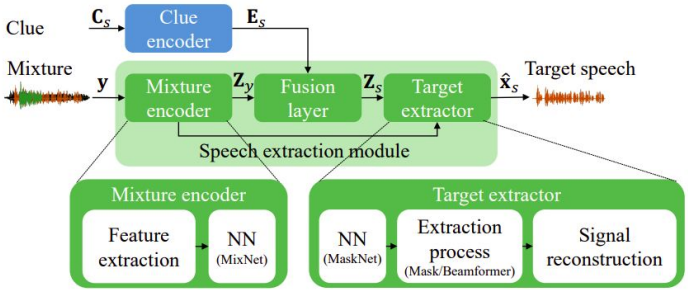


Fig. 2. Comparison of TSE with BSS and noise reduction

Speech extraction module

- The speech extraction module estimates the target speech from the mixture, given the target speaker embeddings.
- The same configuration can be used independently of the type of clue.
- The process can be decomposed into three main parts:
  - Mixture encoder
  - Fusion layer
  - Target extractor



Source 1

Speech Enhancement- TSE- General Framework

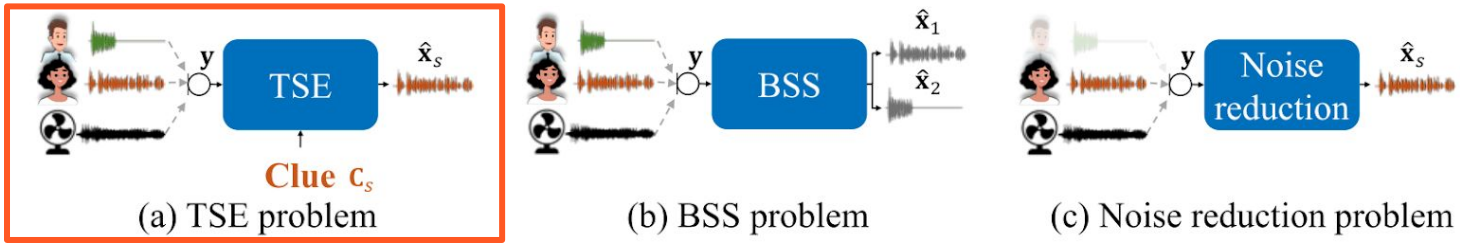
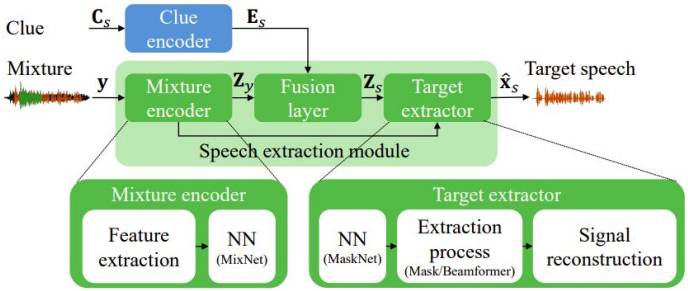


Fig. 2. Comparison of TSE with BSS and noise reduction



Speech extraction module

Mixture encoder:

- extract features (STFT or the-like as in DPRNN)
- Applies few layers of a NN, termed MixNet.
- It outputs the encoded representation of the mixture Zy, which is (at this point) **agnostic** of the target.

$$Z_y = MixEncoder(y; \theta_{Mix})$$

Source 1

Speech Enhancement- TSE- General Framework

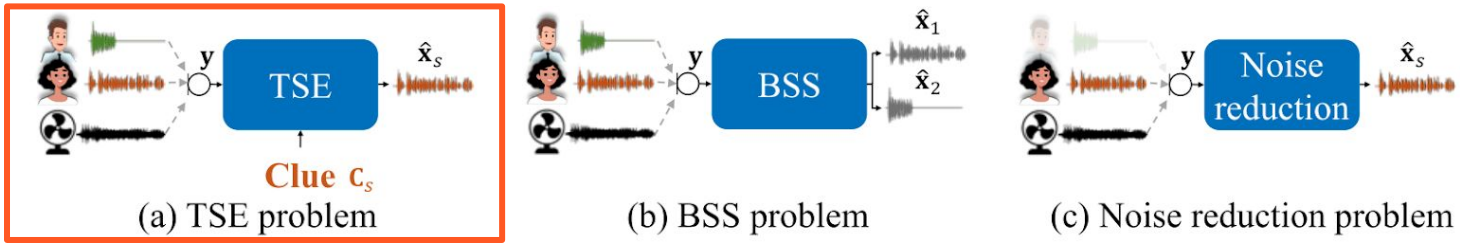
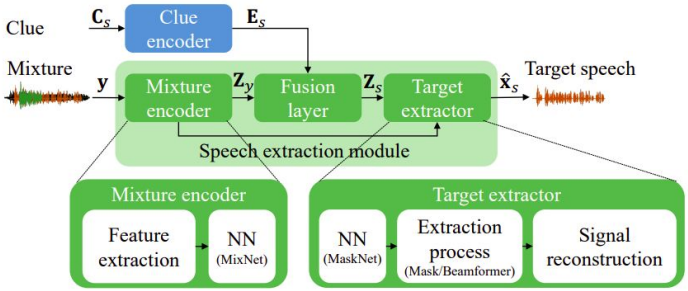


Fig. 2. Comparison of TSE with BSS and noise reduction

Speech extraction module



Fusion / Adaptation layer:

- Allows conditioning of the process on the clue.
- It combines Zy with the clue embeddings Es.
- Fusion can be done using several ways: concatenation, addition, etc.

$$Z_s = Fusion(Z_y, E_s; \theta_{Fusion})$$

Source 1

Speech Enhancement- TSE- General Framework

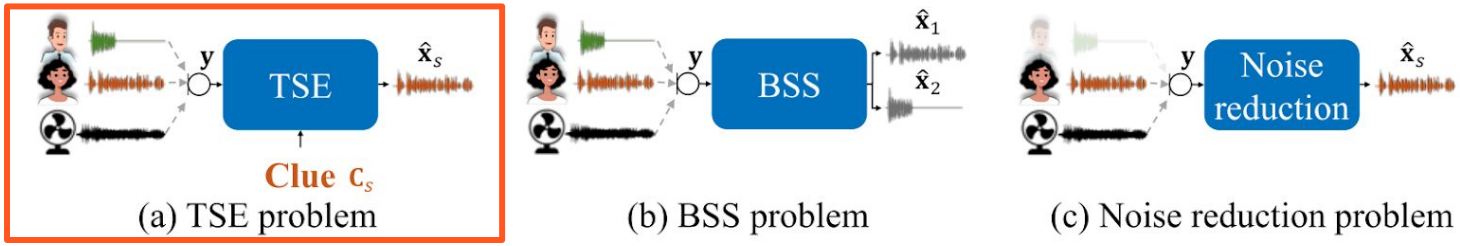
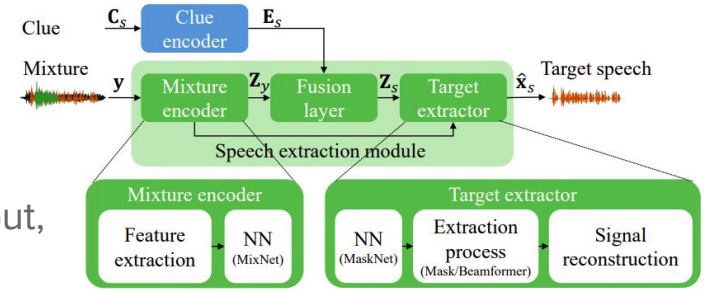


Fig. 2. Comparison of TSE with BSS and noise reduction

Speech extraction module



Target Extraction:

- Estimates the target signal based on the fusion layer's output, and the input signal y
- Can be done using:
  - Masking (elemwize hadamard multiplication)
  - Mapping - regression

$$Z_s = TgtExtractor(Z_s, y; \theta_{TgtExtractor})$$

Source 1

Speech Enhancement- TSE- General Framework

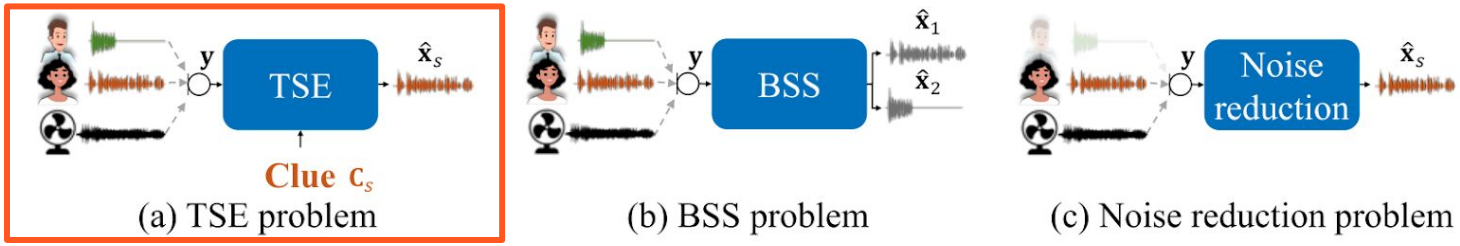


Fig. 2. Comparison of TSE with BSS and noise reduction

Speech extraction module

Target Ex

- Esti
  and
- Can
  - Mapping - regression

Fig. 4. Example of time-frequency mask for speech extraction: Time-frequency mask shows spectrogram regions where target source is dominant. By applying this mask to the mixture, we obtain an extracted speech signal that estimates the target speech.

$$Z_s = TgtExtractor(Z_s, y; \theta_{TgtExtractor})$$

Source 1

Speech Enhancement- TSE- General Framework

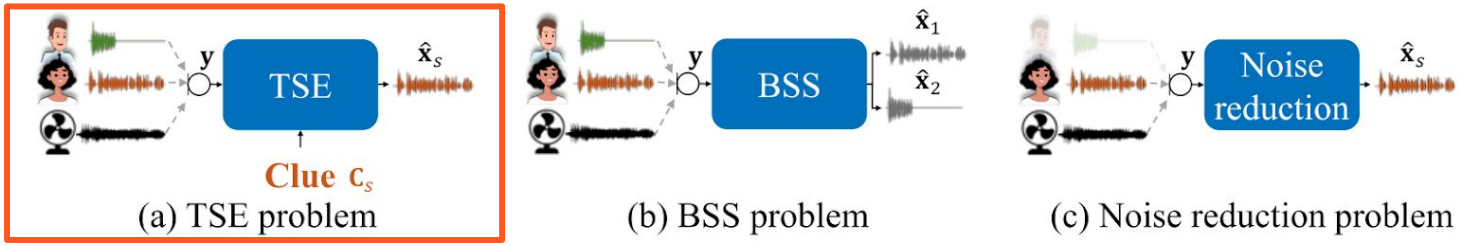
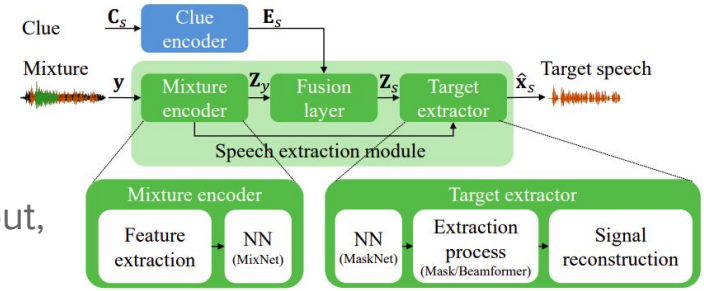


Fig. 2. Comparison of TSE with BSS and noise reduction

Speech extraction module



Target Extraction:

- Estimates the target signal based on the fusion layer's output, and the input signal y
- Can be done using:
  - Masking (elemwize hadamard multipication)
  - Mapping - regression
- Reconstruction: inverse operation of the feature extraction of the mixture encoder (e.g., iSTFT or transposed conv in DPRNN)

$$M_s = MaskNet(Z_s; \theta_{Mask})$$

$$\hat{X}_s = M_s \odot Y$$

$$\hat{x}_s = Reconstruct(\hat{X}_s; \theta_{Reconstruct})$$

Source 1

Speech Enhancement- TSE- Training

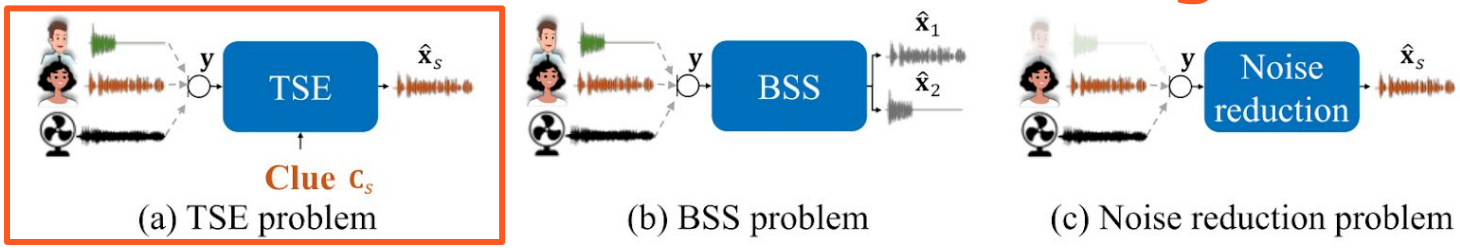


Fig. 2. Comparison of TSE with BSS and noise reduction

We want to optimize the model with parameters $\theta_{TSE} = \{\theta_{Mix}, \theta_{Clue}, \theta_{Fusion}, \theta_{TgtExtractor}\}$

Source 1

Speech Enhancement- TSE- Training

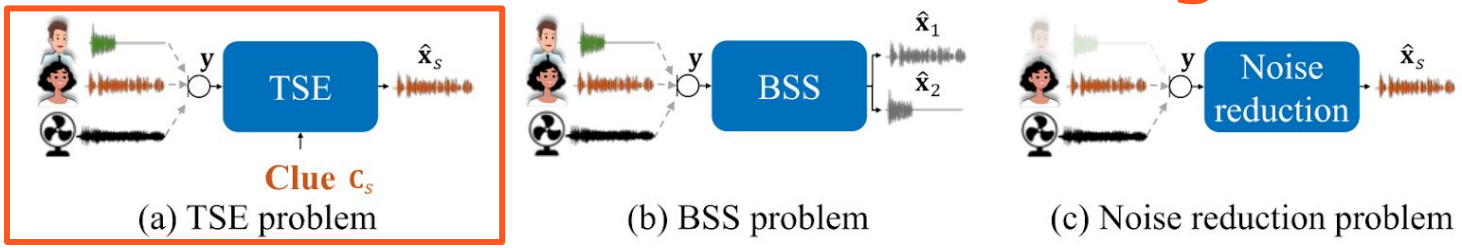


Fig. 2. Comparison of TSE with BSS and noise reduction

Data generation pipeline:

- Select random audio segments from the target speaker, the interference speaker, and the background noise.
- Clues:
  - Audiotory clue: select another speech signal from the target speaker (enrollment)
  - Visual clue: select the video frames associated with the target speech.
- Mix the audios (can add reverb and modify the gains to determine desired SNR).

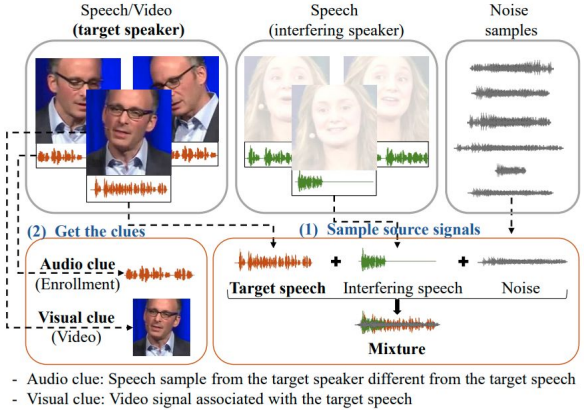


Fig. 5. Example of generating simulation data for training or testing: This example assumes videos are available so that audio and visual clues can be generated. No video is needed for audio clue-based TSE. For visual clue-based TSE, we do not necessarily need multiple videos from the same speaker.

Source 1

Speech Enhancement- TSE- Training

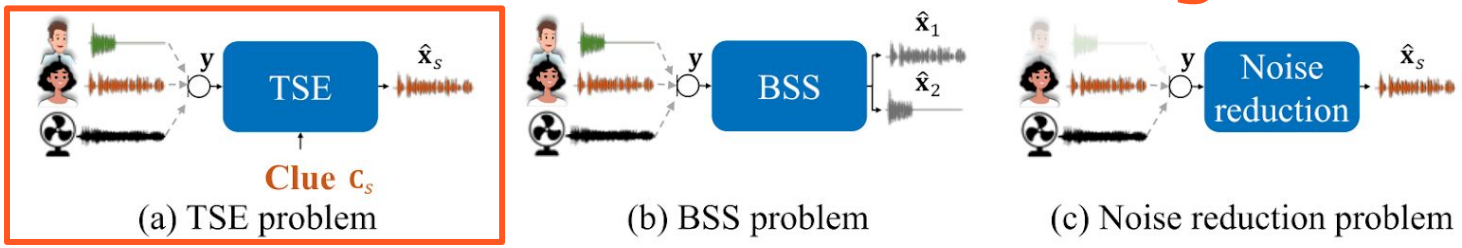


Fig. 2. Comparison of TSE with BSS and noise reduction

Loss

- Similar to BSS
- Measures how close is the estimated target speech is to the target source signal (ground truth).

Source 1

Speech Enhancement- TSE- Training

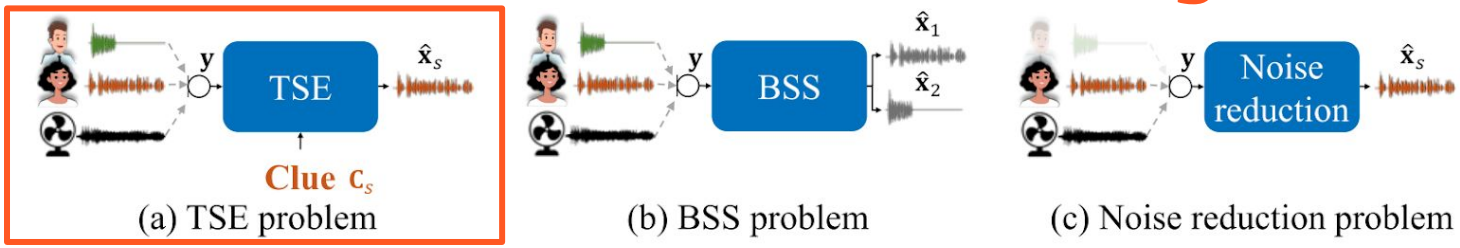


Fig. 2. Comparison of TSE with BSS and noise reduction

Loss

- Similar to BSS
- Measures how close is the estimated target speech is to the target source signal (ground truth).

Clue encoder

- Can be trained jointly/separately with the separation task
- Many times will be trained separately due to computational/bandwidth constraints.

Source 1

Speech Enhancement- TSE- Audio Clues

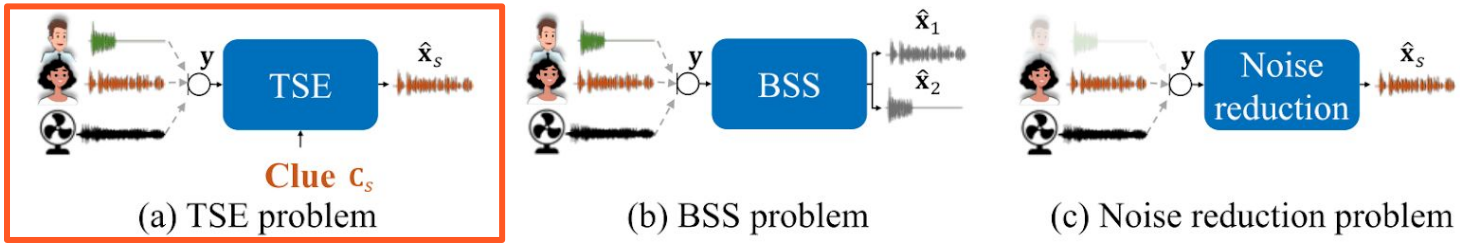


Fig. 2. Comparison of TSE with BSS and noise reduction

- An audio clue is an enrollment utterance spoken by the target speaker from which we derive the voice characteristics.
  - embeddings extracted from a speaker verification/recognition models are often used.
  - There are some open-source models.
- The clue encoder is usually used to extract a single vector that summarizes the entire enrollment utterance.

Source 1

Speech Enhancement- TSE- Audio Clues

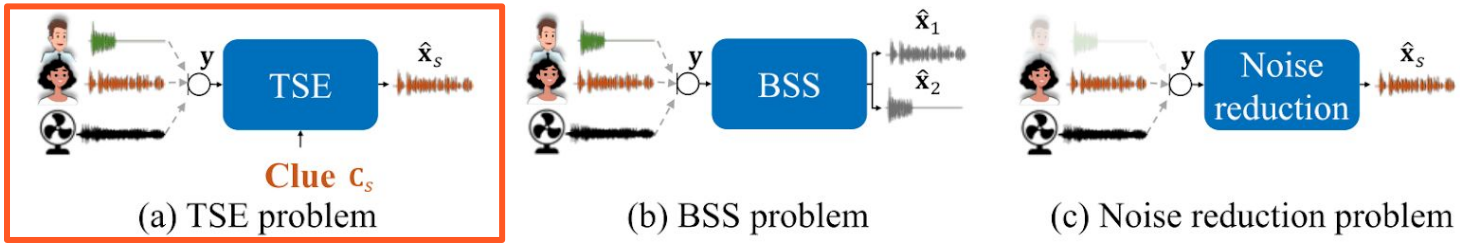


Fig. 2. Comparison of TSE with BSS and noise reduction

Voice Filter

- speaker embeddings: d-vector
- Compute the soft-max prediction (between 0 to 1) based on the noisy input audio and the speaker embeddings

Links to samples: 1, 2

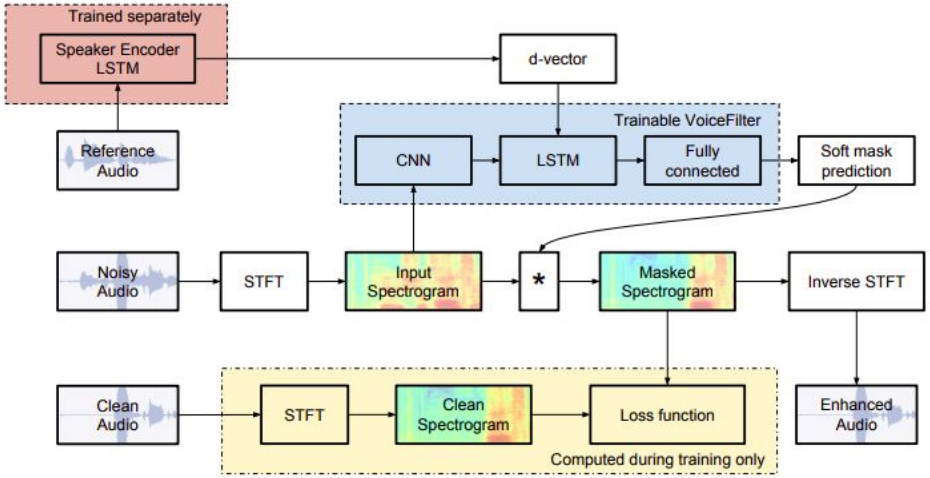


Figure 1: *System architecture.*

Source 1, 2

Speech Enhancement- TSE- Visual Clues

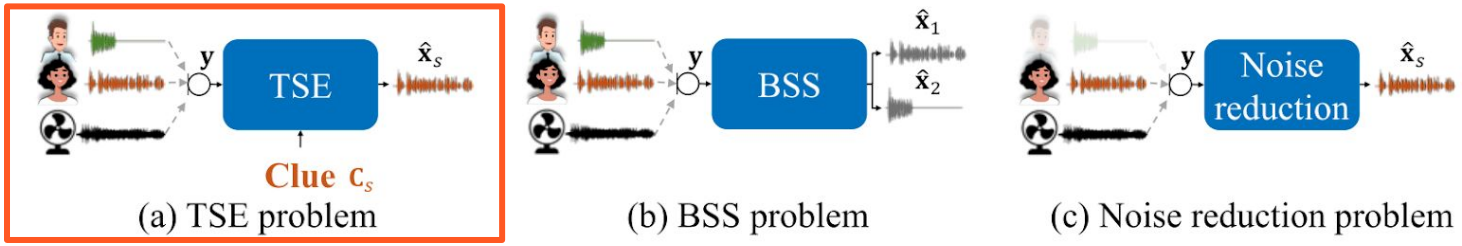


Fig. 2. Comparison of TSE with BSS and noise reduction

- Motivated by psycho-acoustic studies that revealed that humans read lips to understand speech.
- Assumes that a video camera captures the face of the target speaker.
- Can capture whether the target speaker is speaking or not, or even more refined information about the phoneme being uttered.

Source 1

Speech Enhancement- TSE- Visual Clues

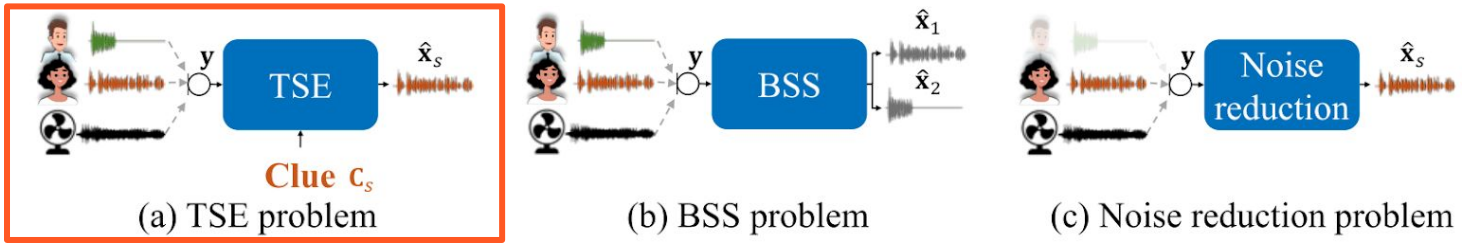


Fig. 2. Comparison of TSE with BSS and noise reduction

Pros:

- Captures information from different modality
- Can better handle mixtures of speakers with similar voices (e.g., same-gender mixtures) than audio clue-based systems
- The users may not need to pre-enroll their voice.

Cons:

- Dependents on having an additional hardware (camera).
- Depends on the illumination, occlusion that may occur, and angle of the target speaker.
- Depends on the syncing of the audio and video

Source 1

Speech Enhancement- TSE- Visual Clues

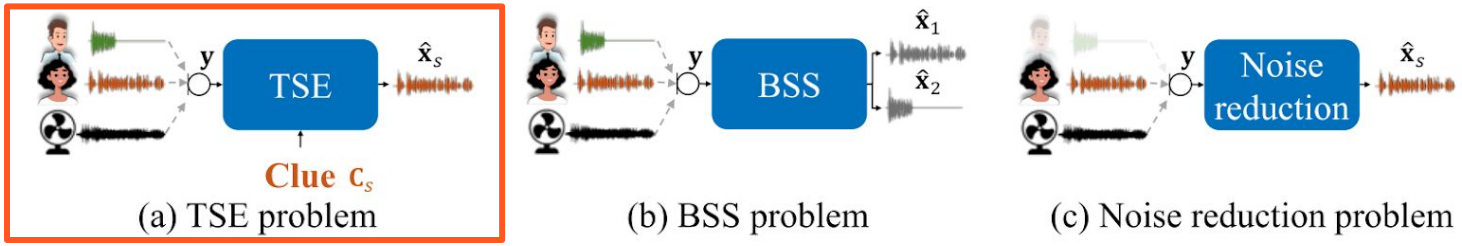


Fig. 2. Comparison of TSE with BSS and noise reduction

Methods

- Three works released concurrently: Zisserman, Efros and Efrat.
- All tried to isolate individual (target) speakers from multi-talker simultaneous speech using visual clues.
- Proposed a deep audio-visual speech enhancement network that is able to separate a speaker's voice given lip regions in the corresponding video, by predicting both the magnitude and the phase of the target signal.
- The method is applicable to speakers unheard and unseen during training, and for unconstrained environments.

Source 1

Speech Enhancement- TSE- Visual Clues

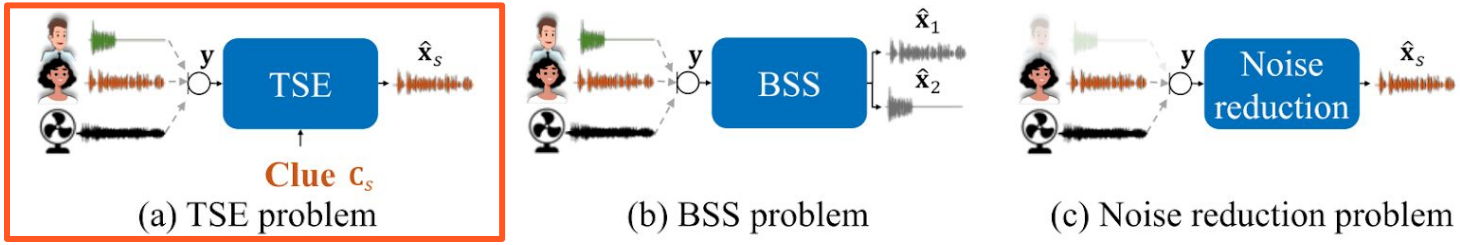


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.- high level overview

- the model consists of 2 sub-networks:
  - Magnitude
  - Phase

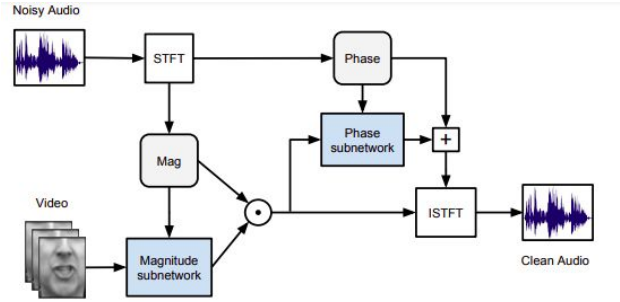
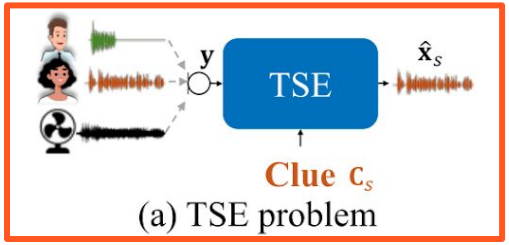


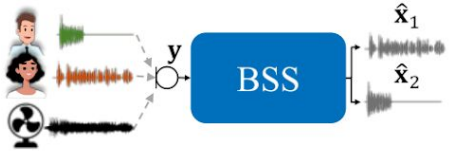
Figure 1: *Audio-visual enhancement architecture overview. It consists of two modules: a magnitude sub-network and a phase sub-network. The first sub-network receives the magnitude spectrograms of the noisy signal and the speaker video as inputs and outputs a soft mask. We then multiply the input magnitudes element-wise with the mask to produce a filtered magnitude spectrogram. The magnitude prediction, along with the phase spectrogram obtained from the noisy signal are then fed into the second sub-network, which produces a phase residual. The residual is added to the noisy phase, producing the enhanced phase spectrograms. Finally the enhanced magnitude and phase spectra are transformed back to the time domain, yielding the enhanced signal.*

Source 1, 2

Speech Enhancement- TSE- Visual Clues



(a) TSE problem



(b) BSS pro

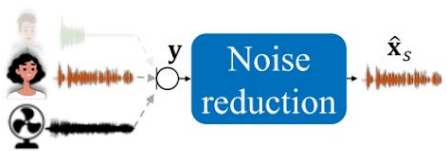


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.

- **lips sig extraction:** The video (seq. of face crops) is passed through a resnet (Lip-Reading model- multiclass (500 classes) on LRW) that outputs a 512 vector representation per frame.

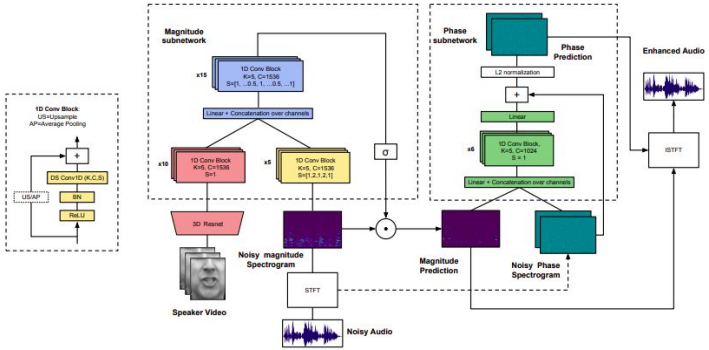
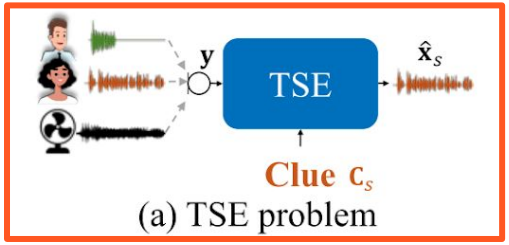


Figure 2: *Audio-visual enhancement network. **BN**: Batch Normalization, **C**: number of channels; **K**: kernel width; **S**: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

Source 1, 2

Speech Enhancement- TSE- Visual Clues



(a) TSE problem

(b) BSS pro

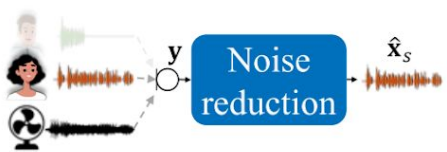


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.

- **lips sig extraction**
- **Processing:** The magnitude components of the mixed audio's STFT along with the feature vectors of the video are passed through two separate networks

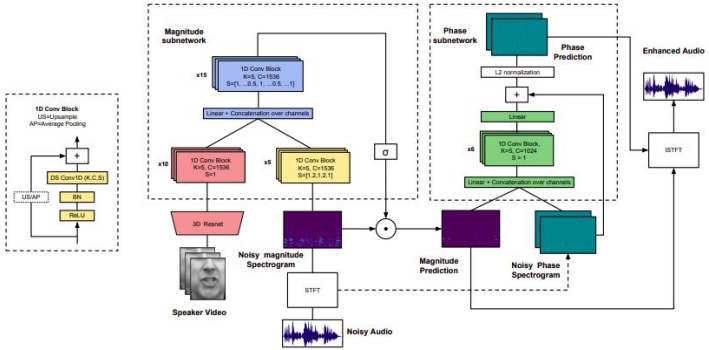
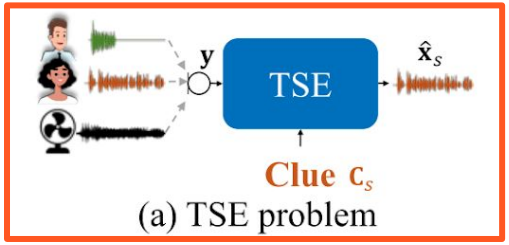


Figure 2: *Audio-visual enhancement network.* ***BN****: Batch Normalization,* ***C****: number of channels;* ***K****: kernel width;* ***S****: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

Source 1, 2

Speech Enhancement- TSE- Visual Clues



(b) BSS pro

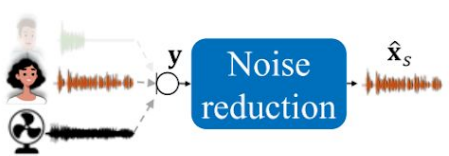


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.

- **lips sig extraction**
- **Processing**
- **Fusion:** The outputs are fused
  - The fused output goes through additional layers - that outputs a spectrogram mask.

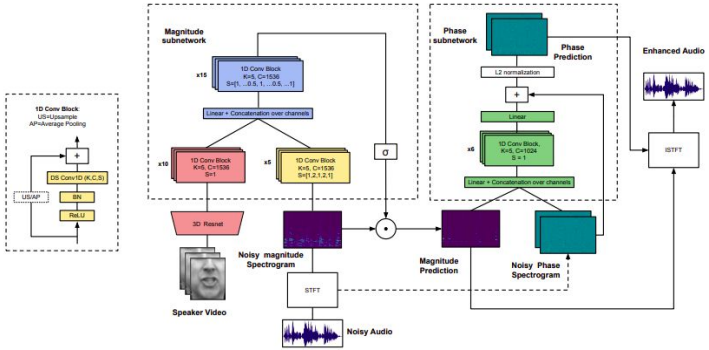


Figure 2: *Audio-visual enhancement network. **BN**: Batch Normalization, **C**: number of channels; **K**: kernel width; **S**: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

Source 1, 2

Speech Enhancement- TSE- Visual Clues

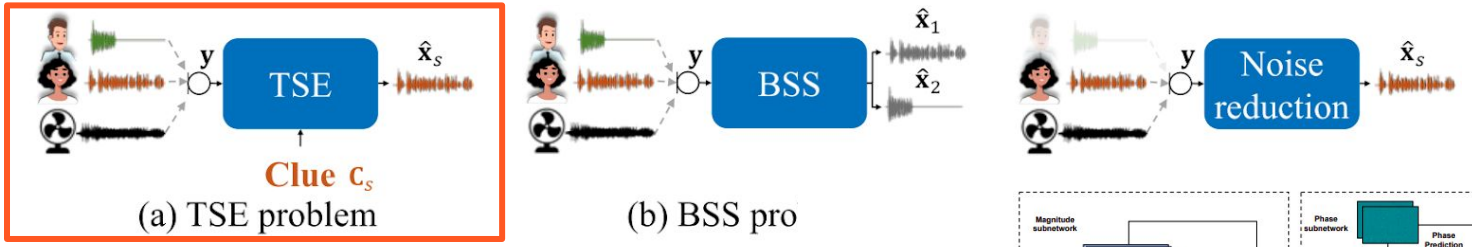


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.

- **lips sig extraction**
- **Processing**
- **Fusion**
- **Mag Mask:** The mask multiplies (element wise) the mixed STFT-magnitude to form the clean mag.

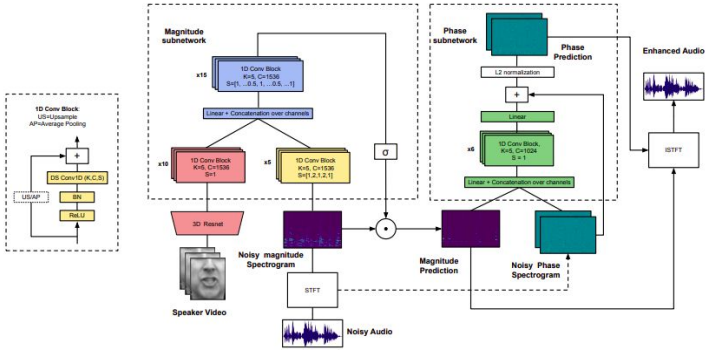


Figure 2: *Audio-visual enhancement network. **BN**: Batch Normalization, **C**: number of channels; **K**: kernel width; **S**: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

Source 1, 2

Speech Enhancement- TSE- Visual Clues

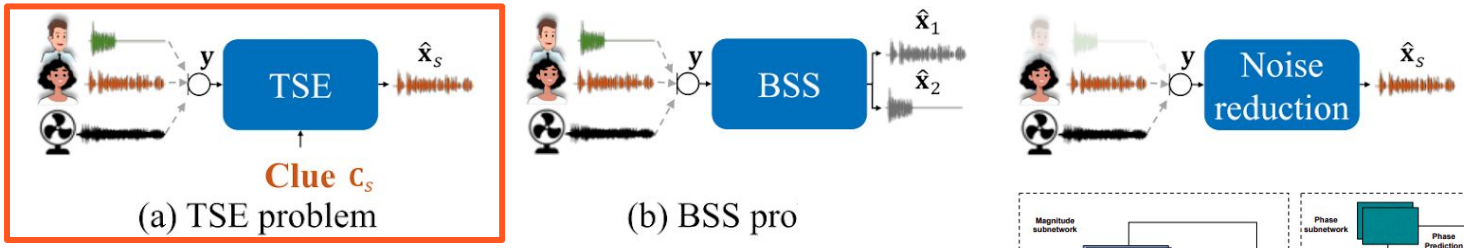


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et.al

- **lips sig extraction**
- **Processing**
- **Fusion**
- **Mag Mask**
- **Phase Residual:** The result is passed to another network which receives as input also the STFT-phase of the original sound and outputs a "phase residual" which is then added to the original STFT-phase signal.

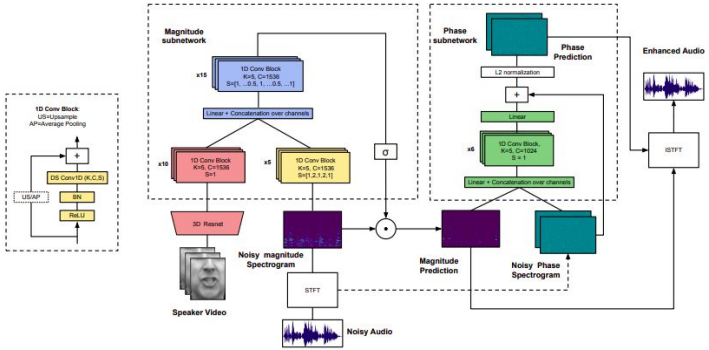
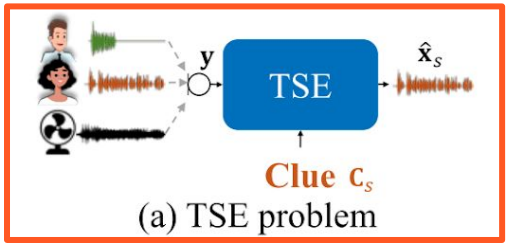


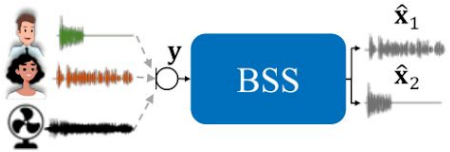
Figure 2: *Audio-visual enhancement network. **BN**: Batch Normalization, **C**: number of channels; **K**: kernel width; **S**: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

Source 1, 2

Speech Enhancement- TSE- Visual Clues



(a) TSE problem



(b) BSS pro

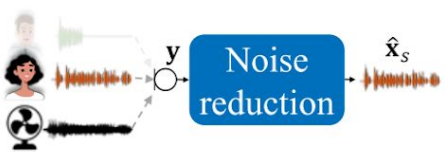


Fig. 2. Comparison of TSE with BSS and noise reduction

Zisserman et al.- Loss + results

L1 loss between the predicted magnitude spectrogram and the ground truth

$$\mathcal{L} = ||\hat{M} - M^*||_1 - \lambda \frac{1}{TF} \sum_{t,f} M^*_{tf} < \hat{\Phi}_{tf}, \Phi^*_{tf} >$$

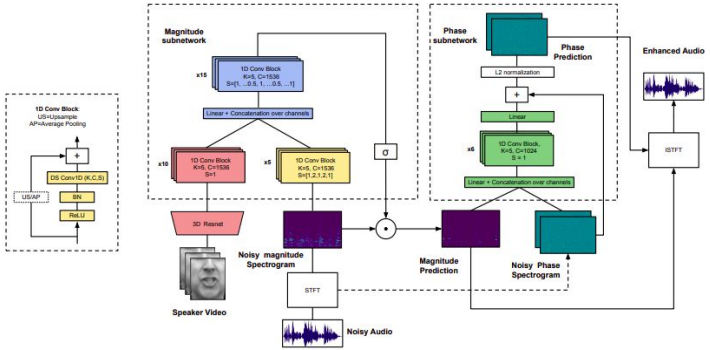


Figure 2: *Audio-visual enhancement network. **BN**: Batch Normalization, **C**: number of channels; **K**: kernel width; **S**: strides – fractional ones denote transposed convolutions. The network consists of a magnitude and a phase sub-network. The basic building unit is the temporal convolutional block with pre-activation [37] shown on the left. Identity skip connections are added after every convolution layer (and speed up training). All convolutional layers have 1536 channels in the magnitude sub-network and 1024 in the phase subnetwork. Depth-wise separable convolution layers [38] are used, which consist of a separate convolution along the time dimension for every channel, followed by a position-wise projection onto the new channel dimensions (equivalent to a convolution with kernel width 1).*

[Link to audio samples]

Source 1, 2

Speech Enhancement- TSE- Aud-Vis Clues

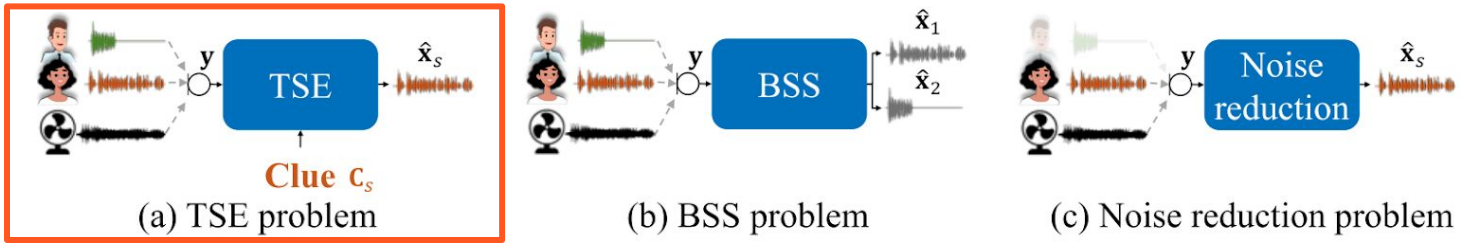


Fig. 2. Comparison of TSE with BSS and noise reduction

Audio-Visual Clues

An audio-visual model for separating a single speaker from a mixture of sounds such as other speakers and background noise.

- Moreover, the model enables to output the speaker even when the visual cues are temporarily absent due to occlusion

(a) *Training samples* **(b)** *Test samples*

Figure 3: *Example frames of occluded videos used during training end evaluation.*

Source 1, 2

Speech Enhancement- TSE- Aud-Vis Clues

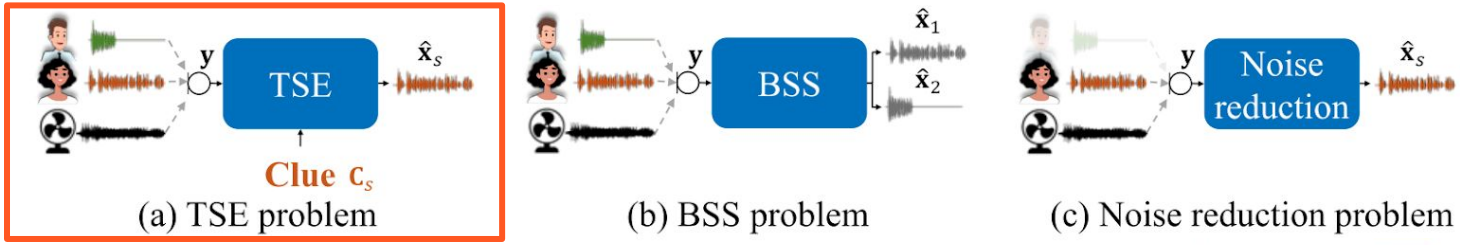


Fig. 2. Comparison of TSE with BSS and noise reduction

Audio-Visual Clues

An audio-visual model for separating a single speaker from a mixture of sounds such as other speakers and background noise.

- Moreover, the model enables to output the speaker even when the visual cues are temporarily absent due to occlusion
- The model separates a speaker's voice by **conditioning on both the speaker's lip movements and/or a audio fingerprint**

Link to audio samples

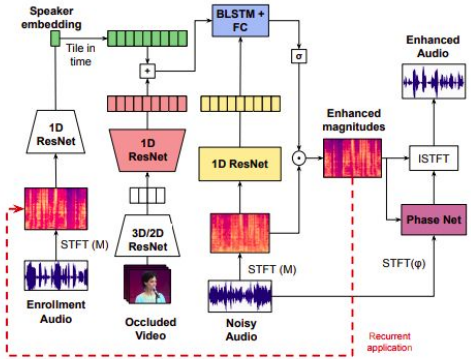


Figure 2: *The architecture of the audio-visual speech enhancement network: There are 2 audio streams. The one processes the incoming noisy audio, while the other takes as input an enrollment audio sample and creates a speaker embedding that captures the speaker's voice characteristics. A visual stream extracts frame-wise representations from the input video. The visual, speaker and audio embeddings are combined and fed into the BLSTM which outputs a multiplicative mask that filters the noisy spectrograms. When no enrollment audio is provided, the enhanced magnitudes (created by a video-only pass) can be used as the input to the speaker embedding network.*

Source 1, 2

Speech Enhancement- TSE- Spatial Clues

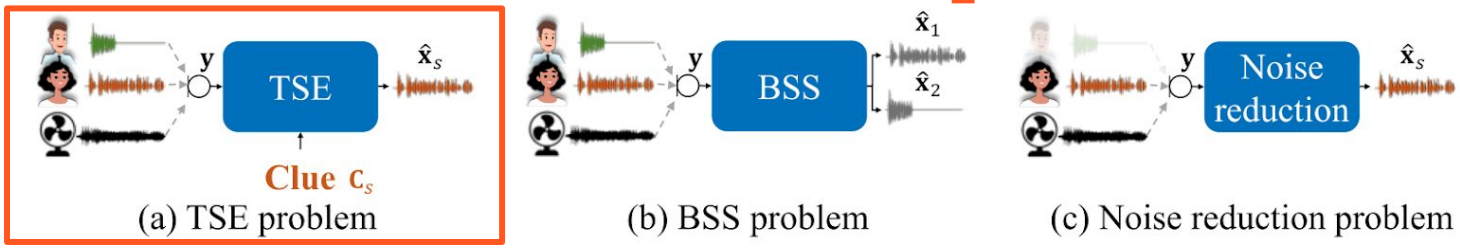
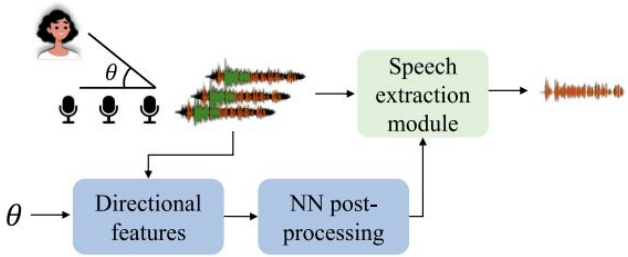
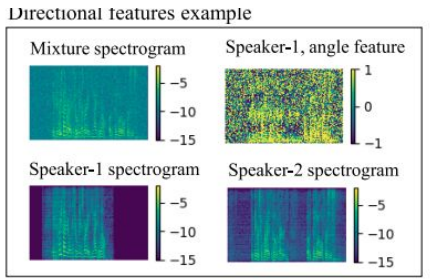


Fig. 2. Comparison of TSE with BSS and noise reduction

Spatial Clues

- Obtaining the spatial information:
  - The target speaker's location may be approx. known in advance (e.g., a driver's position)
  - Using an external source such as a camera
  - Detecting the direction of arrival (DoA) from a multi-channel enrollment utterance of the speaker recorded in the same position.
- The direction of arrival can be represented as a 1-hot vector, and can be processed using a NN model, before the fusion layer.





Source 1

Speech Enhancement- TSE- Spatial Clues

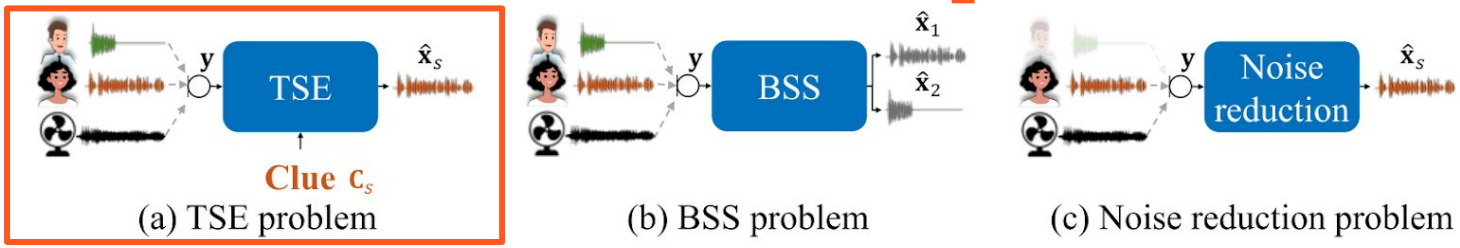
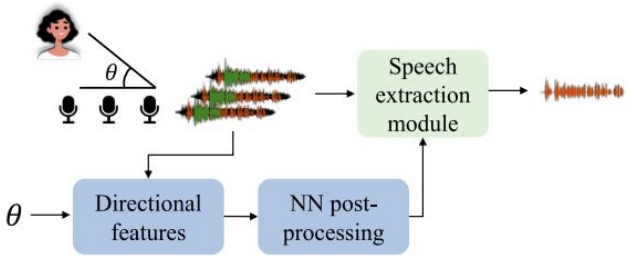


Fig. 2. Comparison of TSE with BSS and noise reduction



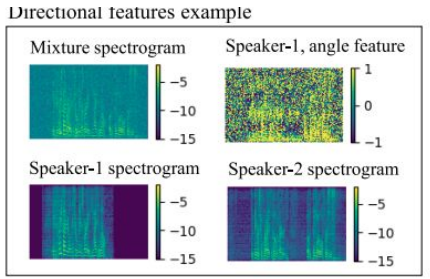
Spatial Clues

Pros:

- Impressive performance even in low SNR

Cons:

- Requires a microphone array (with the same clock)
- Estimation errors of DOA are harmful to proper extraction.
- If the spatial separation of the speakers with respect to the microphone array is not significant enough, the spatial clue may not discriminate between them.



Source 1

Speech Enhancement- TSE- Other Clues

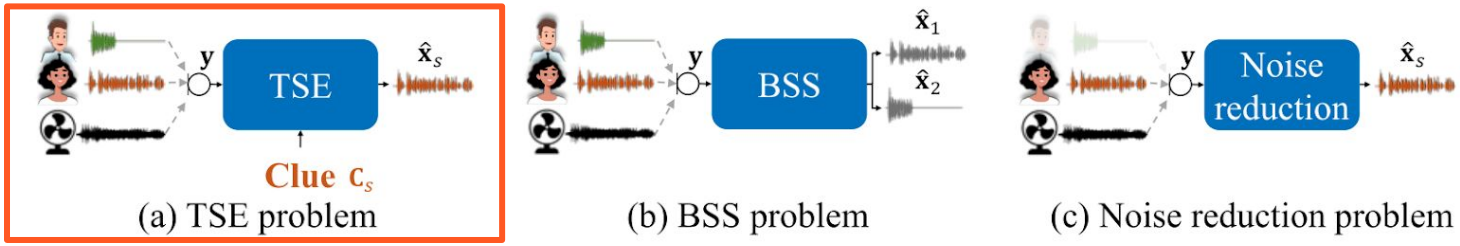


Fig. 2. Comparison of TSE with BSS and noise reduction

Other Clues

- Other interesting clues consist of signals that measure a listener's brain activity to guide the extraction process.
- Electroencephalogram (EEG) signal of a listener focusing on a speaker correlates with the envelope of that speaker's speech signal.

Source 1

Speech Enhancement Questions?

Tal Rosenwein