Unfolding Neural Networks for Compressive Multichannel Blind Deconvolution

Bahareh Tolooshams[*1], Satish Mulleti[*2], Demba Ba[1], and Yonina C. Eldar[2]

[1]Harvard University [2]Weizmann Institute of Science [*]Equal contributions

IEEE ICASSP 2021

Harvard John A. Paulson School of Engineering and Applied Sciences

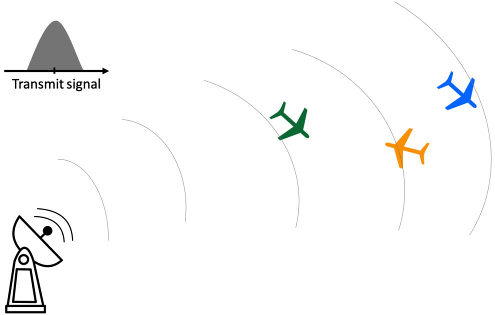
CRISP

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

SAMPLLAB
Signal Acquisition Modeling Processing and Learning

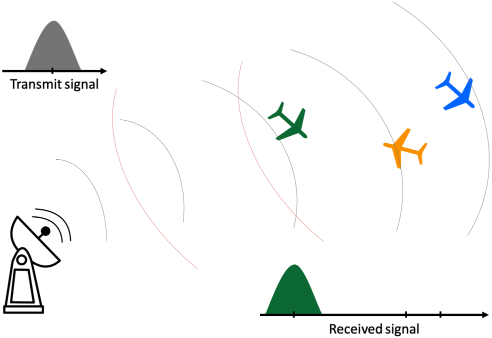
Motivation

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.



Motivation

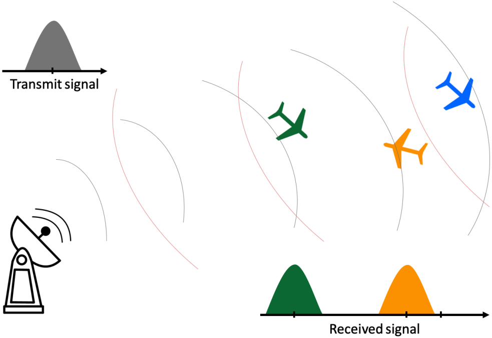
A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.



Motivation

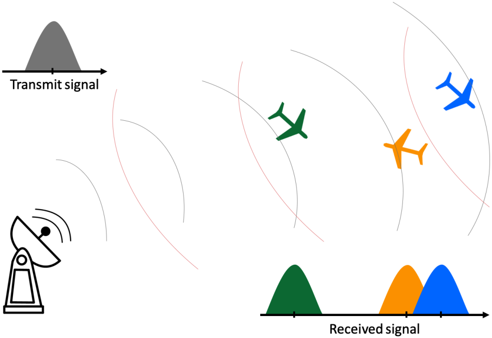
Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.



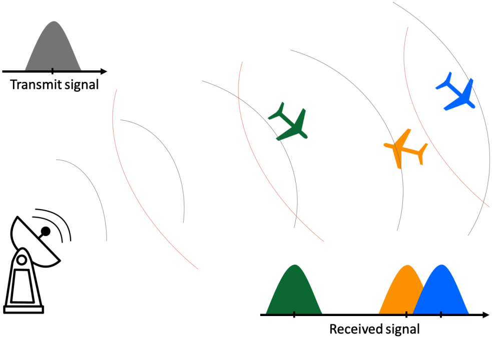
Motivation

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.



Motivation

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.

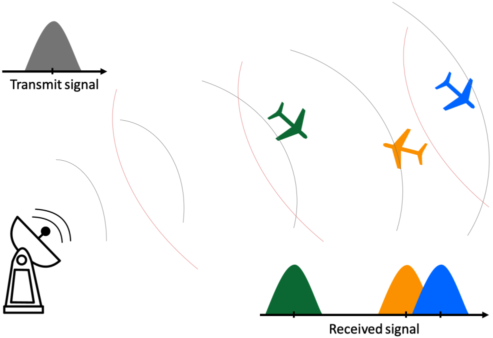


Problem: Recover source (if unknown) and target locations.

Motivation

Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.

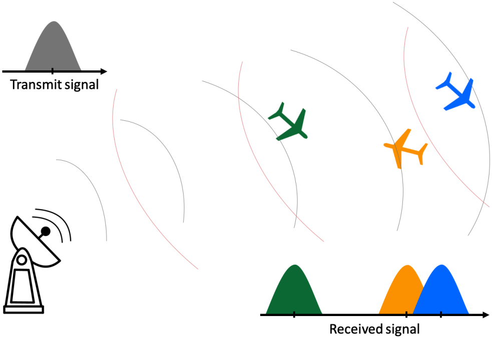


Problem: Recover source (if unknown) and target locations.

Challenges: Receivers' complexity increases with number of measurements.

Motivation

A transmit source signal is reflected from *sparsely* located targets and measured at the receiver.



Problem: Recover source (if unknown) and target locations.

Challenges: Receivers' complexity increases with number of measurements.

Goal: Design a *hardware-efficient* and *data-driven* compression to enable recovery from compressed measurements.

Given $n = 1, \ldots, N$ receiver channels,

Given $n = 1, \ldots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Given $n = 1, \ldots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Sparse target locations (filters): $\mathbf{x}^n$

Given $n = 1, \ldots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Sparse target locations (filters): $\mathbf{x}^n$

Measurements: $\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Given $n = 1, \ldots, N$ receiver channels,

Transmit signal (source):	$\mathbf{s}$
Sparse target locations (filters):	$\mathbf{x}^n$
Measurements:	$\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Goal: Recover $\mathbf{s}$ and $\mathbf{x}^n$ from measurements $\mathbf{y}^n$.

Given $n = 1, \ldots, N$ receiver channels,

Transmit signal (source):	$\mathbf{s}$
Sparse target locations (filters):	$\mathbf{x}^n$
Measurements:	$\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Goal: Recover $\mathbf{s}$ and $\mathbf{x}^n$ from measurements $\mathbf{y}^n$.

One way to solve:

$$\min_{\mathbf{s}, \{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{y}^n - \mathbf{C}_s \mathbf{x}^n\|_2^2 + \lambda \|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Multichannel Blind Deconvolution (MBD)

Sparse-MBD

Given $n = 1, \dots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Sparse target locations (filters): $\mathbf{x}^n$

Measurements: $\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Goal: Recover $\mathbf{s}$ and $\mathbf{x}^n$ from measurements $\mathbf{y}^n$.

One way to solve:

$$\min_{\mathbf{s}, \{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{y}^n - \mathbf{C}_s \mathbf{x}^n\|_2^2 + \lambda \|\mathbf{x}^n\|_1 \quad \text{s.t.} \quad \|\mathbf{s}\|_2 = 1$$

Challenges:

Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Given $n = 1, \dots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Sparse target locations (filters): $\mathbf{x}^n$

Measurements: $\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Goal: Recover $\mathbf{s}$ and $\mathbf{x}^n$ from measurements $\mathbf{y}^n$.

One way to solve:

$$\min_{\mathbf{s}, \{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{y}^n - \mathbf{C}_s \mathbf{x}^n\|_2^2 + \lambda \|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Challenges:

- Requires access to full measurements $\mathbf{y}^n$.

Given $n = 1, \dots, N$ receiver channels,

Transmit signal (source): $\mathbf{s}$

Sparse target locations (filters): $\mathbf{x}^n$

Measurements: $\mathbf{y}^n = \mathbf{s} * \mathbf{x}^n = \mathbf{C}_s \mathbf{x}^n$

Goal: Recover $\mathbf{s}$ and $\mathbf{x}^n$ from measurements $\mathbf{y}^n$.

One way to solve:

$$\min_{\mathbf{s}, \{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{y}^n - \mathbf{C}_s \mathbf{x}^n\|_2^2 + \lambda \|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Challenges:

- Requires access to full measurements $\mathbf{y}^n$.
- Computationally demanding.

Recover $\mathbf{s}$ and $\mathbf{x}^n$ from *compressive* measurements $\mathbf{z}^n = \mathbf{\Phi}\mathbf{y}^n$.

Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Recover $\mathbf{s}$ and $\mathbf{x}^n$ from *compressive* measurements $\mathbf{z}^n = \mathbf{\Phi y}^n$.

Solve:

$$\min_{\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^N \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Recover $\mathbf{s}$ and $\mathbf{x}^n$ from *compressive* measurements $\mathbf{z}^n = \mathbf{\Phi}\mathbf{y}^n$.

Solve:

$$\min_{\mathbf{s}, \{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Prior works:

Compressive sparse-MBD

Recover $\mathbf{s}$ and $\mathbf{x}^n$ from *compressive* measurements $\mathbf{z}^n = \mathbf{\Phi}\mathbf{y}^n$.

Solve:

$$\min_{\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Prior works:

- Pick $\mathbf{\Phi}$ as a random matrix [1]:
  - fast ✓, not hardware-efficient ✗

Recover $\mathbf{s}$ and $\mathbf{x}^n$ from *compressive* measurements $\mathbf{z}^n = \mathbf{\Phi}\mathbf{y}^n$.

Solve:

$$\min_{\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = 1$$

Prior works:

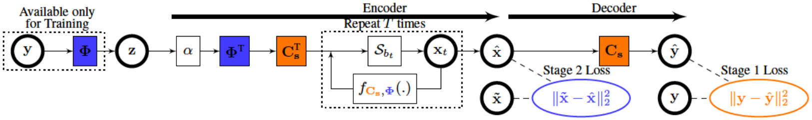
- Pick $\mathbf{\Phi}$ as a random matrix [1]:
  - fast ✓, not hardware-efficient ✗
- Design a *structured* $\mathbf{\Phi}$ [2]:
  - slow ✗, hardware-efficient ✓

Our Contributions

Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution

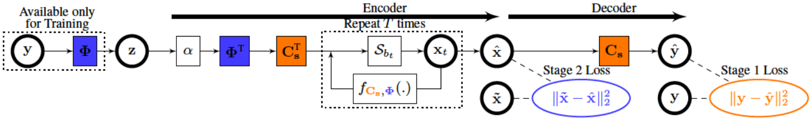


Our Contributions

Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

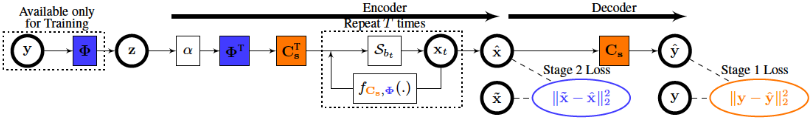
Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution



- An unfolding neural network

Our Contributions

Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution



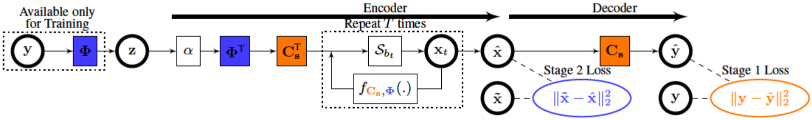
- An unfolding neural network
  - Fast and computational efficient ✓

Our Contributions

Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution

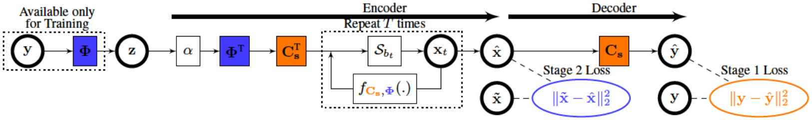


- An unfolding neural network
  - Fast and computational efficient ✓
- Learned structured compression

Our Contributions

Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution

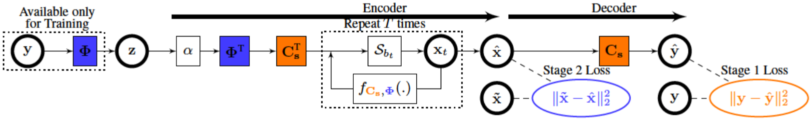


- An unfolding neural network
  - Fast and computational efficient ✓
- Learned structured compression
  - Hardware-efficient ✓

Our Contributions

Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

Learned **S**tructured compressive **M**ultichannel **B**lind **D**econvolution



- An unfolding neural network
  - Fast and computational efficient ✓
- Learned structured compression
  - Hardware-efficient ✓
  - Superior performance against prior works ✓

LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\mathsf{trunc}} = \mathbf{\Phi y}$$

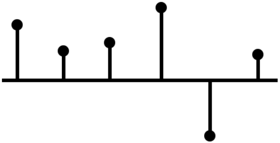
LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\text{trunc}} = \mathbf{\Phi}\mathbf{y}$$

Full Measurements

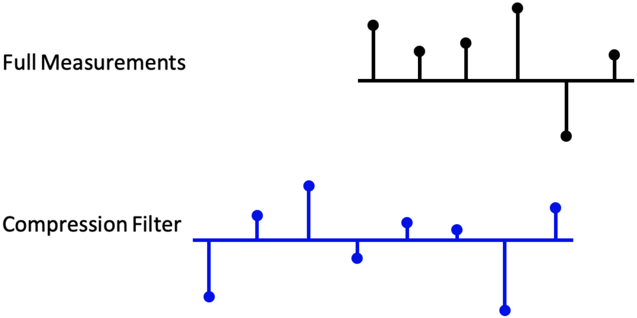


LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\text{trunc}} = \mathbf{\Phi}\mathbf{y}$$

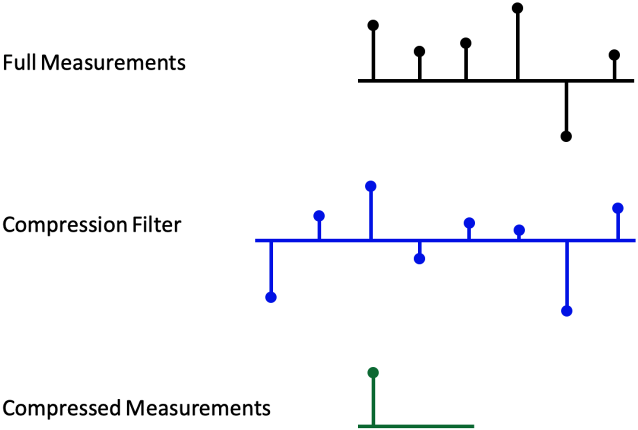


LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\text{trunc}} = \mathbf{\Phi y}$$

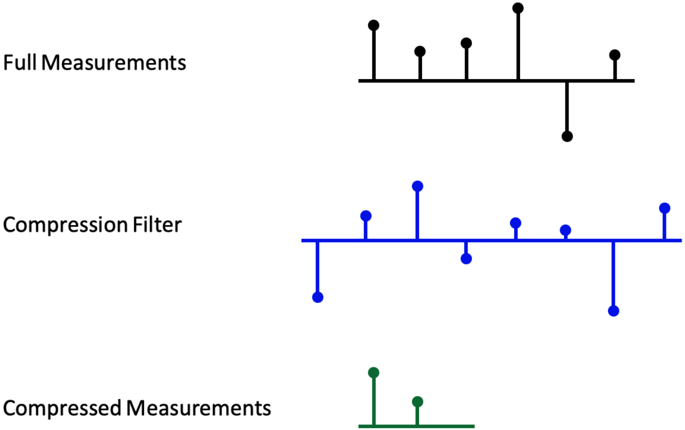


LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\text{trunc}} = \mathbf{\Phi}\mathbf{y}$$

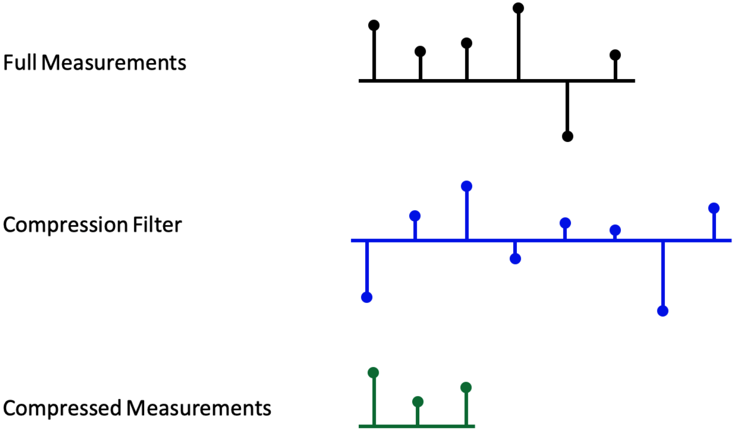


LS-MBD

Compression operator

Compress through a *convolution* followed by a *truncation*.

$$\mathbf{z} = \lfloor \mathbf{h} * \mathbf{y} \rfloor_{\text{trunc}} = \boldsymbol{\Phi} \mathbf{y}$$



LS-MBD

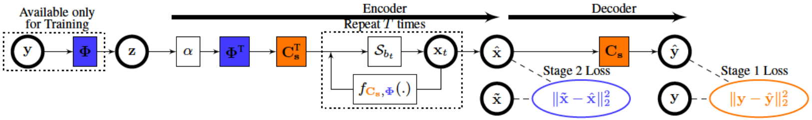
Network architecture

$$\min_{\mathbf{\Phi},\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^N \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = \|\mathbf{h}\|_2 = 1$$

LS-MBD

Network architecture

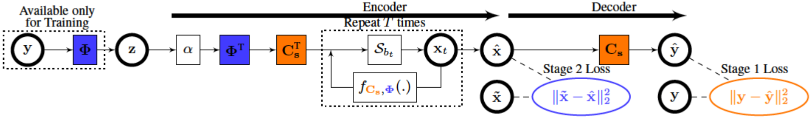
$$\min_{\boldsymbol{\Phi},\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \boldsymbol{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t.} \ \|\mathbf{s}\|_2 = \|\mathbf{h}\|_2 = 1$$



Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

$$\min_{\mathbf{\Phi},\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = \|\mathbf{h}\|_2 = 1$$

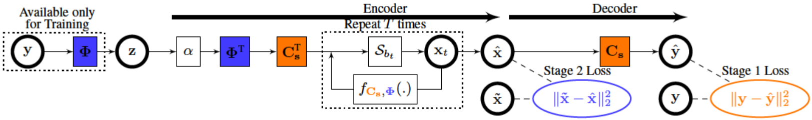


Unfolding neural network:

LS-MBD

Network architecture

$$\min_{\Phi,\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \Phi \mathbf{C}_s \mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t.} \quad \|\mathbf{s}\|_2 = \|\mathbf{h}\|_2 = 1$$



Unfolding neural network:

- Encoder: proximal gradient descent to map compressed measurements $\mathbf{z}^n$ to target locations $\mathbf{x}^n$.

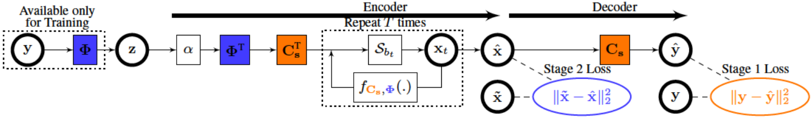
LS-MBD

Network architecture

Harvard John A. Paulson School of Engineering and Applied Sciences

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

$$\min_{\mathbf{\Phi},\mathbf{s},\{\mathbf{x}^n\}_{n=1}^N} \sum_{n=1}^{N} \frac{1}{2}\|\mathbf{z}^n - \mathbf{\Phi}\mathbf{C}_s\mathbf{x}^n\|_2^2 + \lambda\|\mathbf{x}^n\|_1 \quad \text{s.t. } \|\mathbf{s}\|_2 = \|\mathbf{h}\|_2 = 1$$

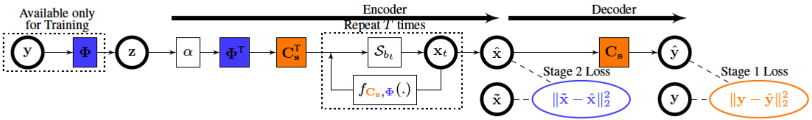


Unfolding neural network:

- Encoder: proximal gradient descent to map compressed measurements $\mathbf{z}^n$ to target locations $\mathbf{x}^n$.
- Decoder: use the source $\mathbf{C}_s$ to reconstruct full measurements $\mathbf{y}^n$.

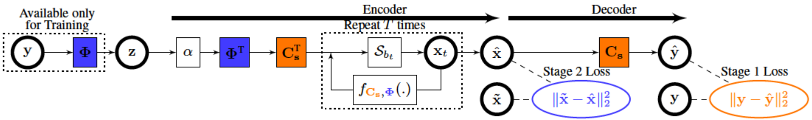
LS-MBD

Training



LS-MBD

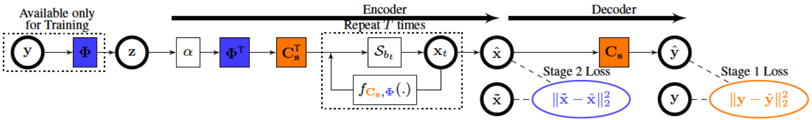
Training



Stage 1:

LS-MBD

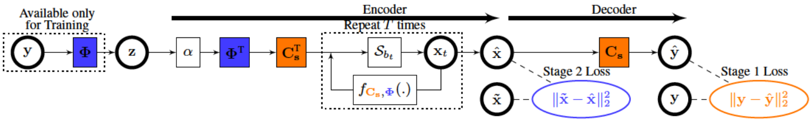
Training



Stage 1:

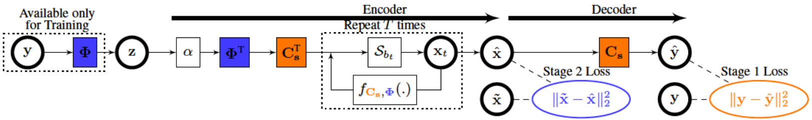
- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).

Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE



Stage 1:

- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.

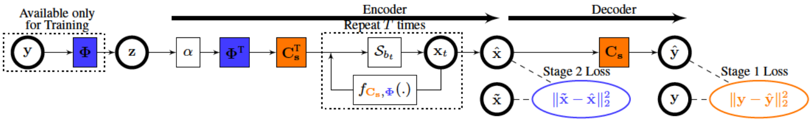


Stage 1:

- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.
- Backward pass: Learn source $\mathbf{C}_s$.

LS-MBD

Training



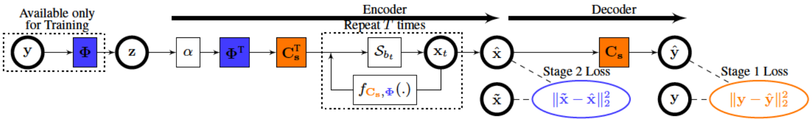
Stage 1:

- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.
- Backward pass: Learn source $\mathbf{C}_s$.

Stage 2:

LS-MBD

Training



Stage 1:

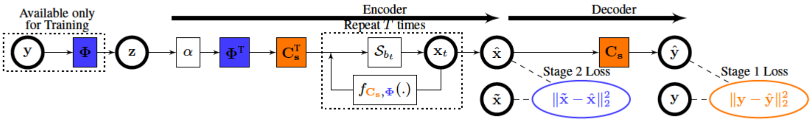
- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.
- Backward pass: Learn source $\mathbf{C}_s$.

Stage 2:

- Take estimated codes $\tilde{\mathbf{x}}$ and source $\mathbf{C}_s$ from stage 1.

LS-MBD

Training



Stage 1:

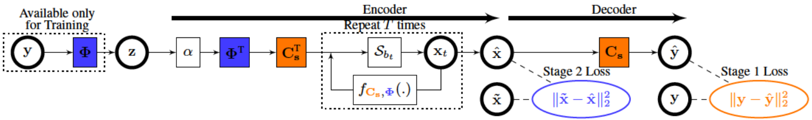
- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.
- Backward pass: Learn source $\mathbf{C}_s$.

Stage 2:

- Take estimated codes $\tilde{\mathbf{x}}$ and source $\mathbf{C}_s$ from stage 1.
- Forward pass: Estimate code & compute loss function $\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|_2^2$.

LS-MBD

Training



Stage 1:

- Train with full measurements to recover source $\mathbf{C}_s$ (i.e., set $\mathbf{\Phi} = \mathbf{I}$).
- Forward pass: Estimate code & compute loss function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$.
- Backward pass: Learn source $\mathbf{C}_s$.

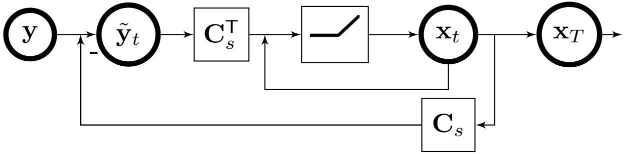
Stage 2:

- Take estimated codes $\tilde{\mathbf{x}}$ and source $\mathbf{C}_s$ from stage 1.
- Forward pass: Estimate code & compute loss function $\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|_2^2$.
- Backward pass: Learn compression $\mathbf{\Phi}$.

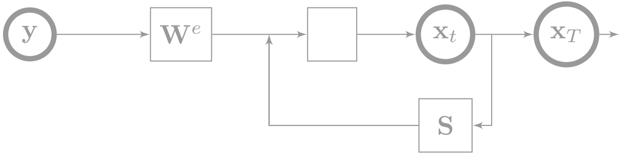
Harvard John A. Paulson School of Engineering and Applied Sciences
מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

Solve sparse coding by iterative proximal gradient algorithm.

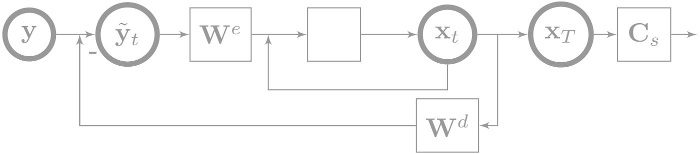
ISTA [3]:



LISTA [4]:



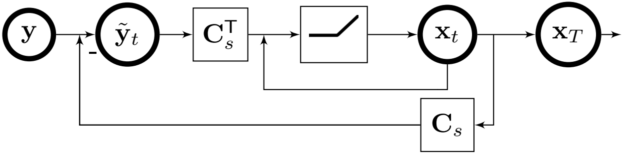
CRsAE [5]:



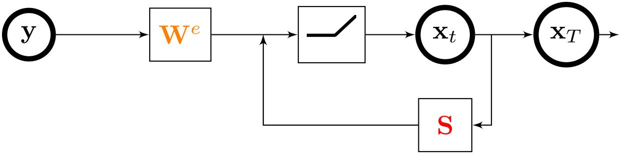
Prior Works on Unfolding Networks

Solve sparse coding by iterative proximal gradient algorithm.

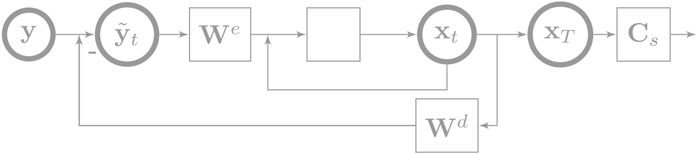
ISTA [3]:



LISTA [4]:



CRsAE [5]:

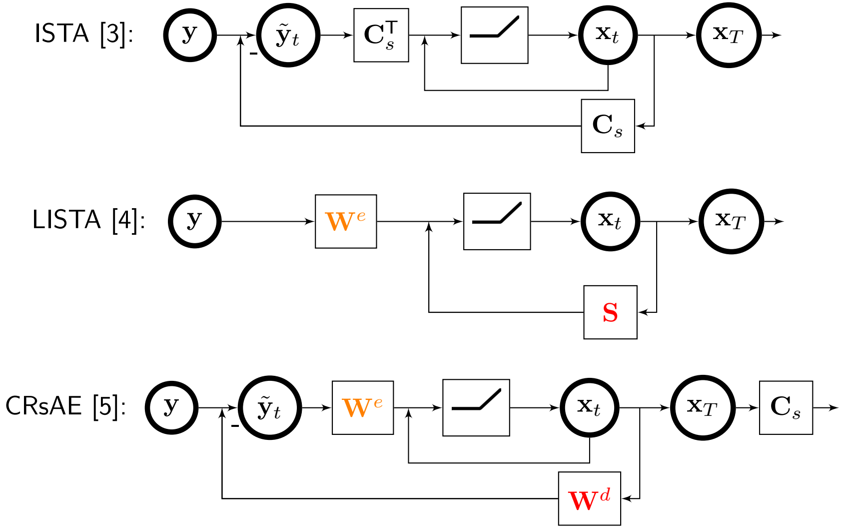


Prior Works on Unfolding Networks

Harvard John A. Paulson School of Engineering and Applied Sciences

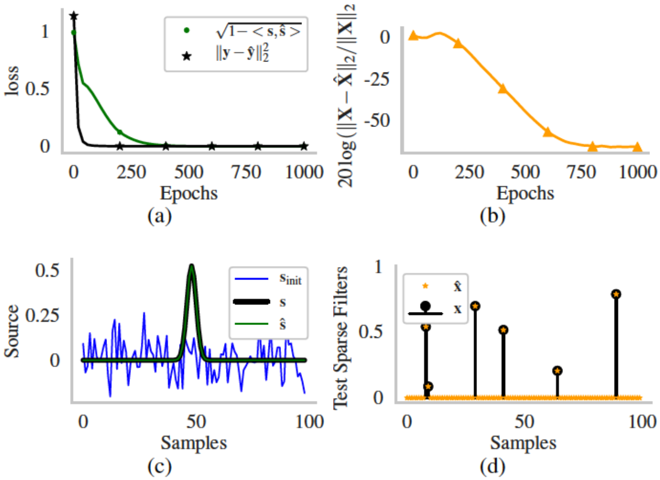
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Solve sparse coding by iterative proximal gradient algorithm.



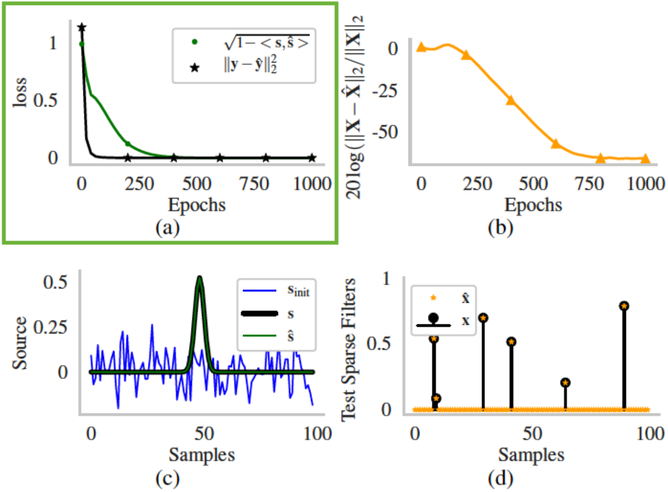
Results

Recovery performance (I)



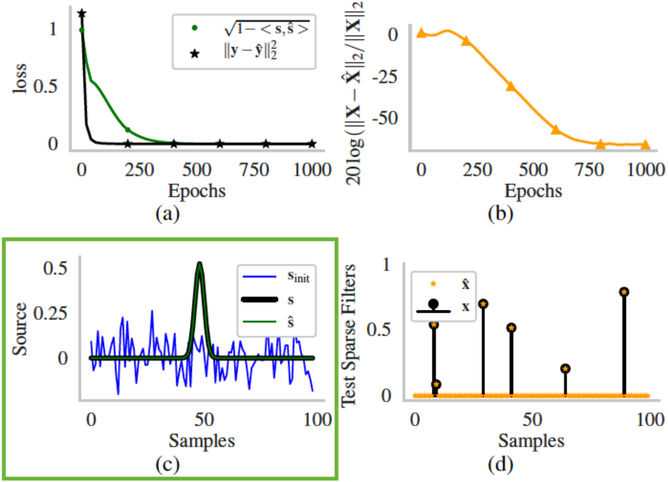
Results

Recovery performance (I)



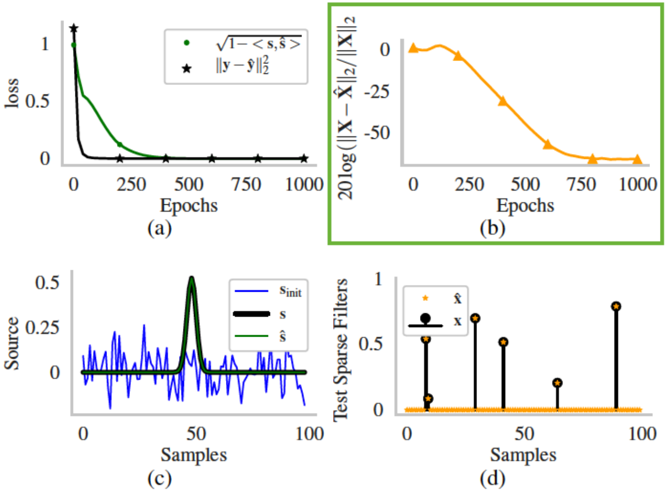
Results

Recovery performance (I)



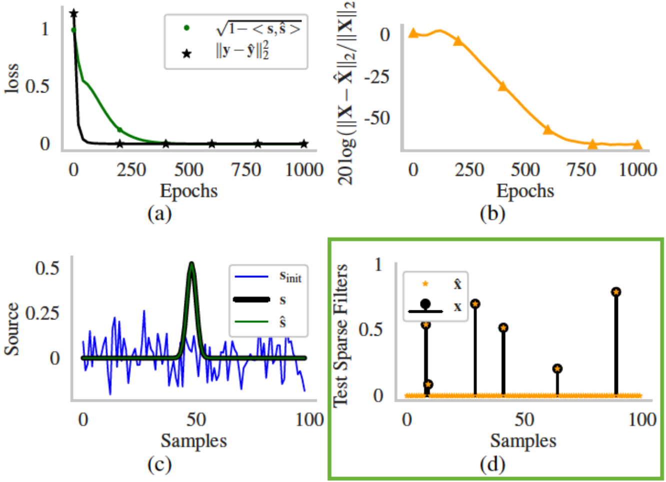
Results

Recovery performance (I)



Results

Recovery performance (I)



Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Method \ Cost	Memory Storage	Complexity
Structured	$O(M_h)$	$O(M_h \log M_h)$
Unstructured	$O(M_y M_z)$	$O(M_y M_z)$

Results

Recovery performance (II)

- **LS-MBD**: $\mathbf{\Phi}$ is learned and structured.
- **LS-MBD-L**: $\mathbf{\Phi}$ is learned and structured (relaxed network as in LISTA).
- **GS-MBD**: $\mathbf{\Phi}$ is random Gaussian and structured.
- **FS-MBD**: $\mathbf{\Phi}$ is designed, fixed, and structured.
- **G-MBD**: $\mathbf{\Phi}$ is random Gaussian matrix.

Sparse code recovery error

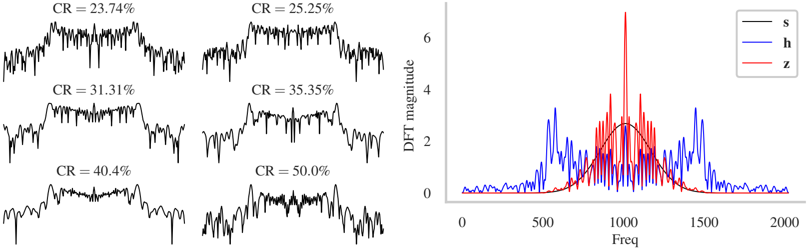
CR [%]	M_z	G-MBD	GS-MBD	FS-MBD	LS-MBD	LS-MBD-L
50	99	**-54.05**	-44.93	-43.96	**-53.27**	-26.54
40.4	80	**-55.07**	-40.55	-26.52	**-52.80**	-
35.35	70	**-52.43**	-40.00	-22.76	**-51.50**	-
31.31	62	**-53.63**	-37.13	-21.86	**-54.71**	-
25.25	50	**-53.36**	-28.57	-8.40	**-51.41**	-
23.74	47	**-50.60**	-26.11	-6.84	**-50.35**	-
22.72	45	**-52.98**	-23.17	-6.14	-43.61	-
20.20	40	-47.39	-14.75	-5.13	-17.07	-

Method \ Cost	Memory Storage	Complexity
Structured	$O(M_h)$	$O(M_h \log M_h)$
Unstructured	$O(M_y M_z)$	$O(M_y M_z)$

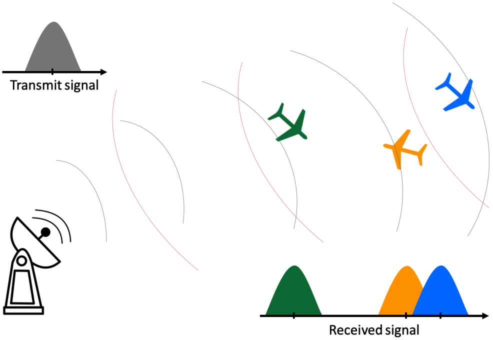
Speed \ Method	G-MBD	FS-MBD	LS-MBD	LS-MBD-L
runtime [s]	4.9087	164	5.4204	**0.0028**

Results

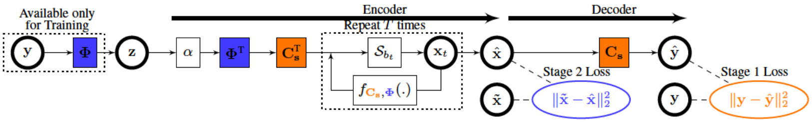
Compression visualizations



Conclusion



Transmit signal
Received signal



Available only for Training
Encoder
Decoder
Repeat T times
Stage 2 Loss
Stage 1 Loss

References

T. Chang, B. Tolooshams, and D. Ba, "Randnet: Deep learning with compressed measurements of images," in *Proc. Workshop on Machine Learning for Signal Process. (MLSP)*, pp. 1–6, 2019.

S. Mulleti, K. Lee, and Y. C. Eldar, "Identifiability conditions for compressive multichannel blind deconvolution," *IEEE Trans. Signal Process.*, vol. 68, pp. 4627–4642, 2020.

I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

K. Gregor and Y. Lecun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, pp. 399–406, 2010.

B. Tolooshams, S. Dey, and D. Ba, "Deep residual autoencoders for expectation maximization-inspired dictionary learning," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2020.