[2] SAMPLLAB Signal Acquisition Modeling Processing and Learning

[1] CRISP

Unfolding Neural Networks for Compressive Multichannel Blind Deconvolution

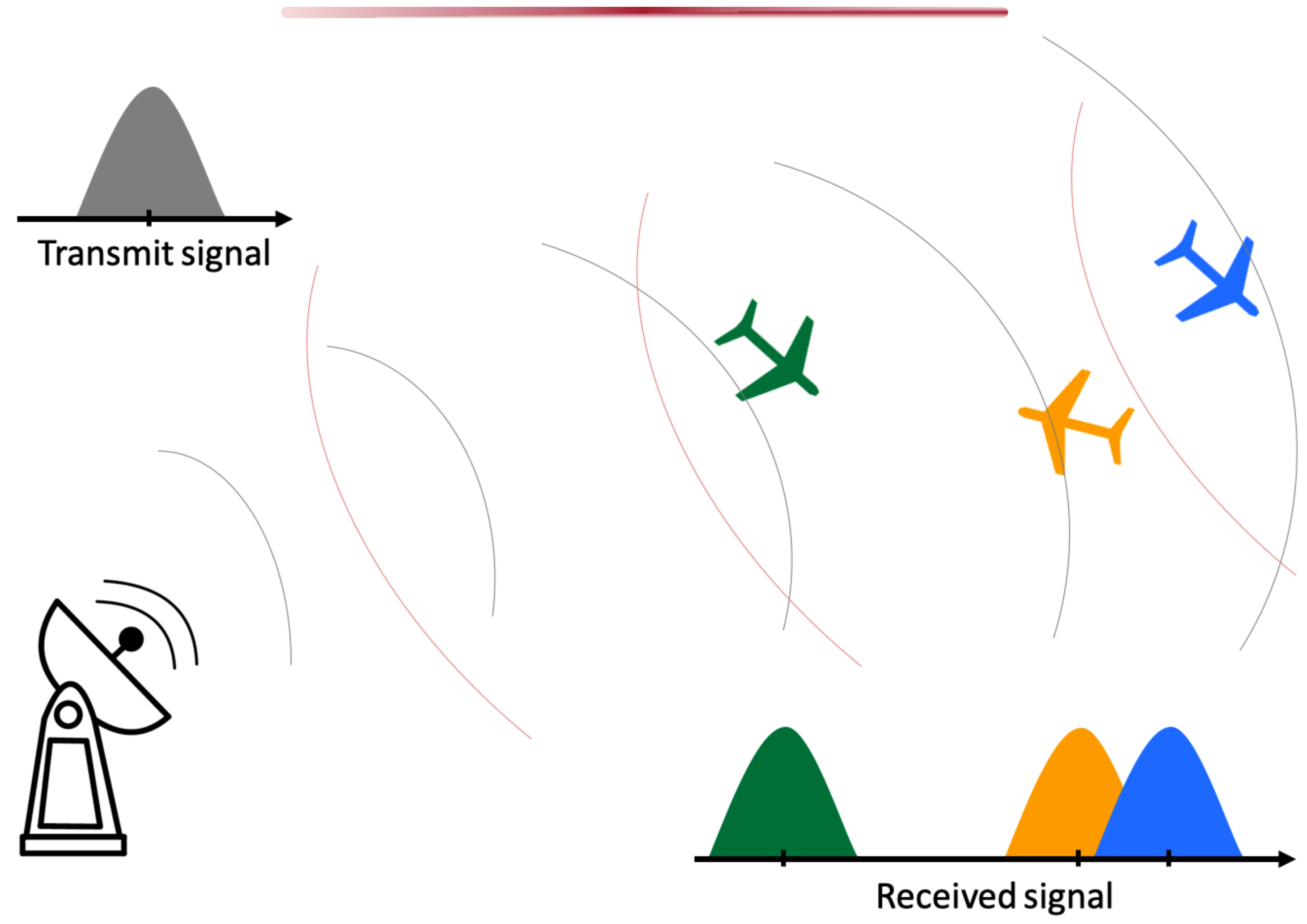
Bahareh Tolooshams*[1], Satish Mulleti*[2], Demba Ba[1], and Yonina C. Eldar[2]

[1] Harvard John A. Paulson School of Engineering and Applied Sciences

[2] מכון ויצמן למדע WEIZMANN INSTITUTE OF SCIENCE

Paper ID: 3563

Introduction



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.

Background

Sparse Multichannel Blind Deconvolution (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 s and x^{n} from measurements y^n.

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

Challenges: Requires *full* measurements. *Computationally* demanding and *slow*.

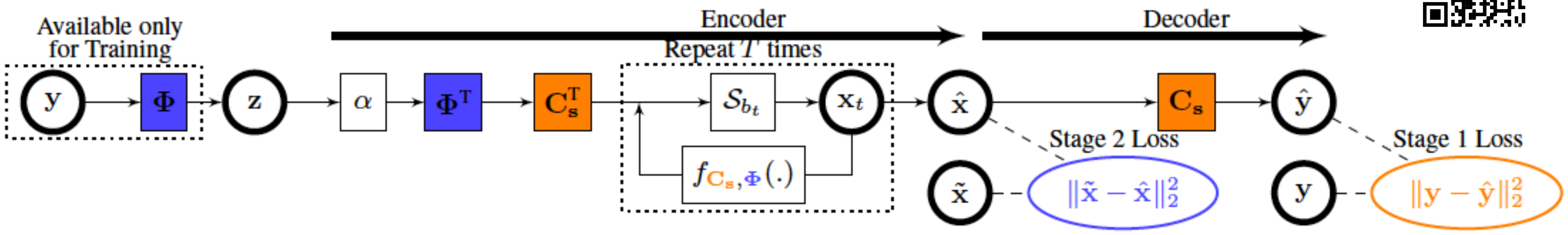
Compressive Sparse-MBD

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

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

Methods

Learned Structured Compressive Multichannel Blind Deconvolution Network (LS-MBD)



Compression Operator

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

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

Unfolding Neural Network

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

- **Encoder**: proximal gradient descent to map compressed measurements z^{n} to target locations x^{n}.
- **Decoder**: use the source to reconstruct full measurements.

Training

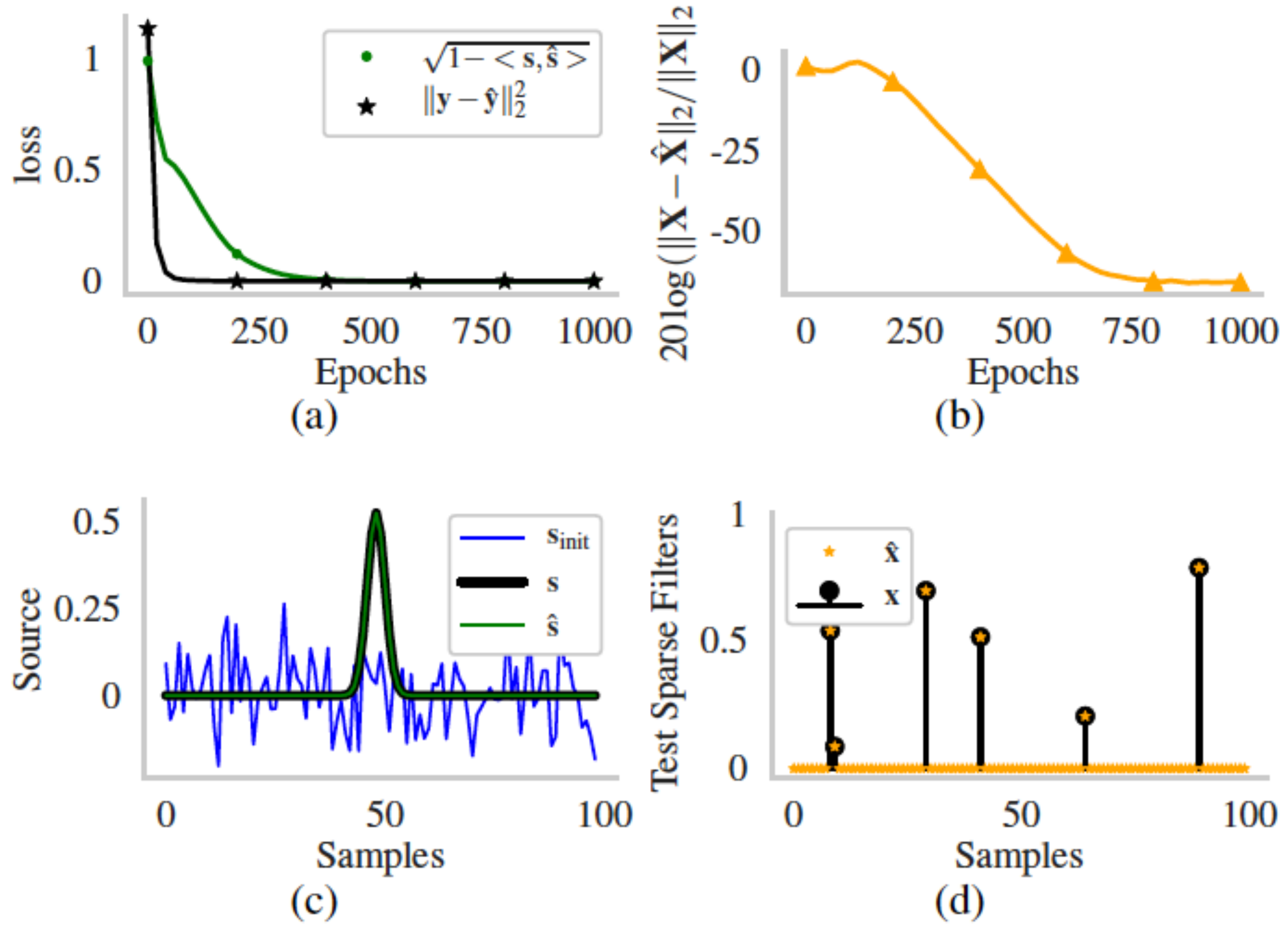
Stage 1:

- Train with full measurements to recover source.
- Forward pass: Estimate code & compute loss function.
- Backward pass: Estimate source.

Stage 2:

- Take estimated codes and source from stage 1.
- Forward pass: Estimate code & compute loss function.
- Backward pass: Learn compression.

Results



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

- **LS-MBD (ours)**: Φ is learned and structured.
- **LS-MBD-L**: Φ is learned and structured (relaxed as in LISTA).
- **GS-MBD**: Φ is random Gaussian and structured.
- **FS-MBD**: Φ is designed, fixed, and structured.
- **G-MBD**: Φ 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	-

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