Data-Driven Tuning of Proximal Gradient Descent Algorithms for Signal Recovery Problems

Tadashi WADAYAMA

wadayama@nitech.ac.jp

Nagoya Institute of Technology

This work was partly supported by JSPS Grant-in-Aid for Scientific Research

(A) Grant Number 17H01280 and (B) 19H02138

Physical layer wireless comm + ML

- **Wireless communications + Machine Learning (ML)** is becoming a hot research topic, recently

Arise of deep learning (DL)

Deep learning became a key technology for image recognition, speech recognition, and natural text processing

Next generation signal processing is required for 5G/6G

mmWave massive MIMO, blind channel estimation, error correction, terminal localization, beam forming

Fusion of AI/ML and wireless communications in near future

AI/ML becomes a key task for next gen. wireless comm: e.g., distributed learning, federated learning

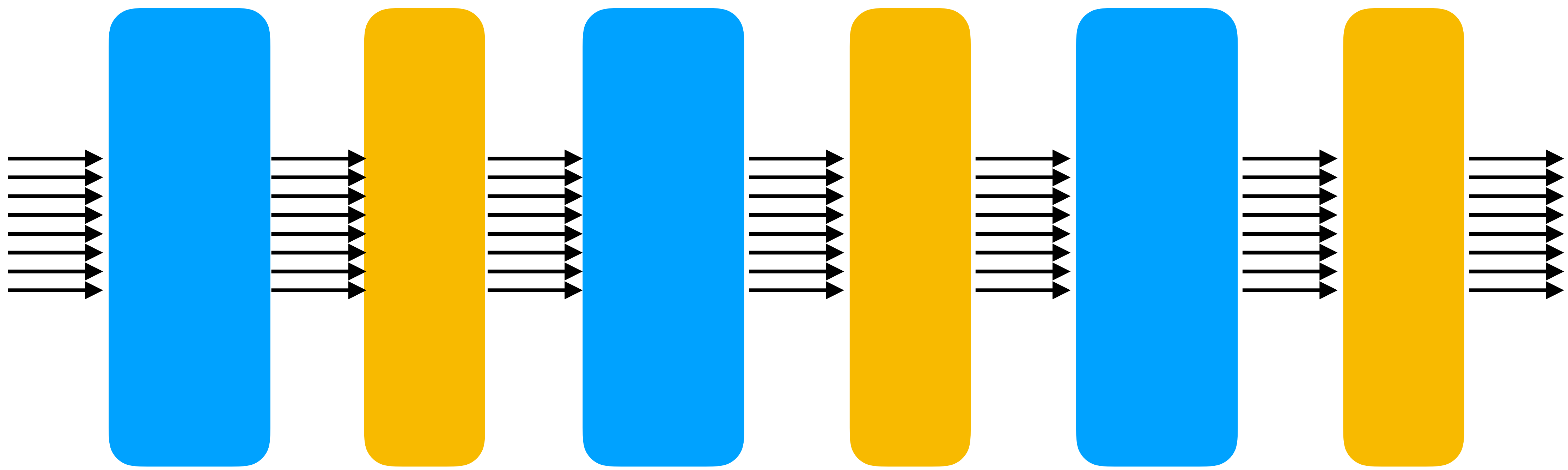
Physical layer wireless comm + ML

Type	idea/concept
Black box modeling	Simulation of channel / detector by deep neural networks, no prior knowledge
Deep unfolding	Applying standard DL technique to optimize parameterized iterative algorithms
Learning to optimize	Simulation of convex/non-convex solver by neural networks
Distributed learning	Independent training and gradient aggregation
Reinforcement learning	Learning optimal strategy to control of an agent based on rewards

Deep neural networks

$$f_\Theta : \mathbb{R}^n \to \mathbb{R}^m$$

$$\Theta = \{W_1, b_1, W_2, b_2, \dots\}$$



$W_1 h_1 + b_1$

Linear layer

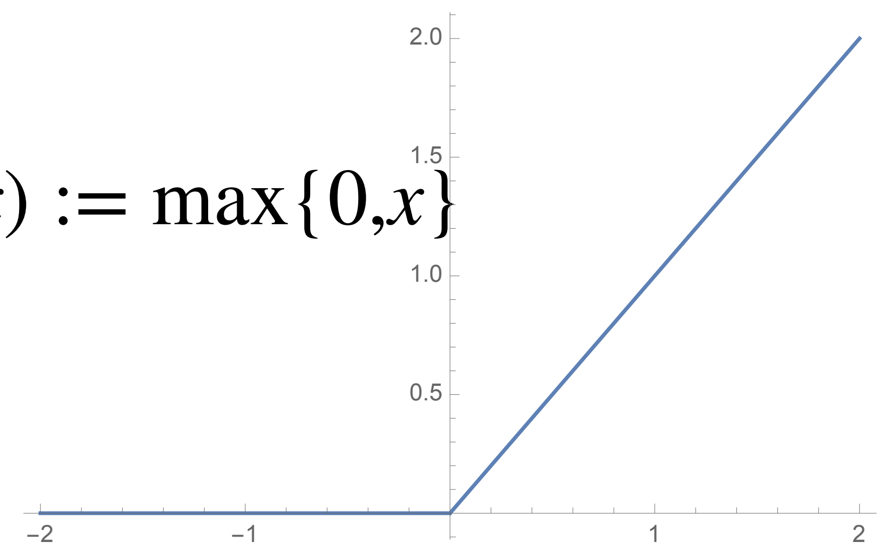
Nonlinear activation function

$h' = W h + b$

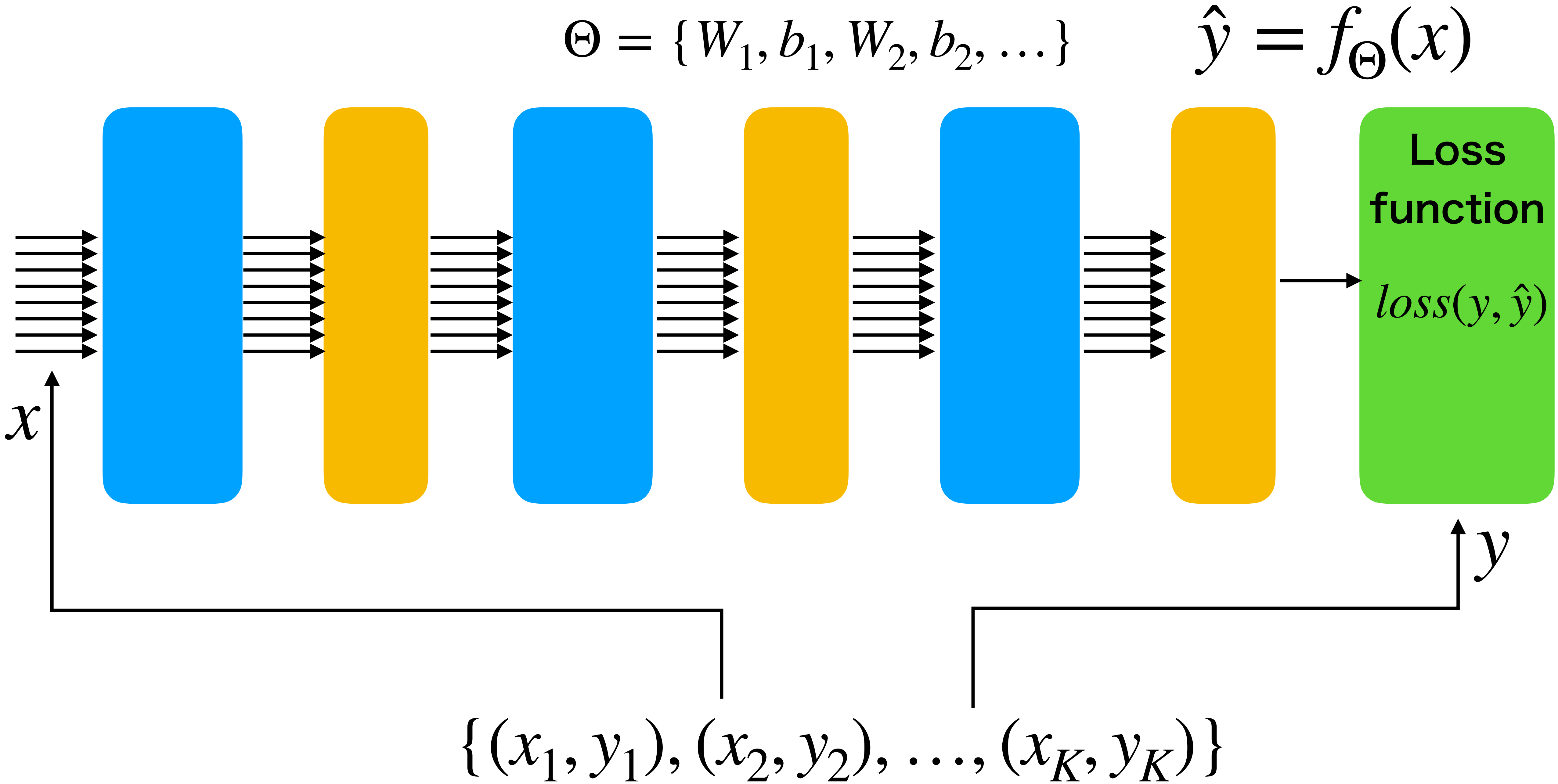
Bias vector

Weight matrix

$ReLU(x) := \max\{0, x\}$

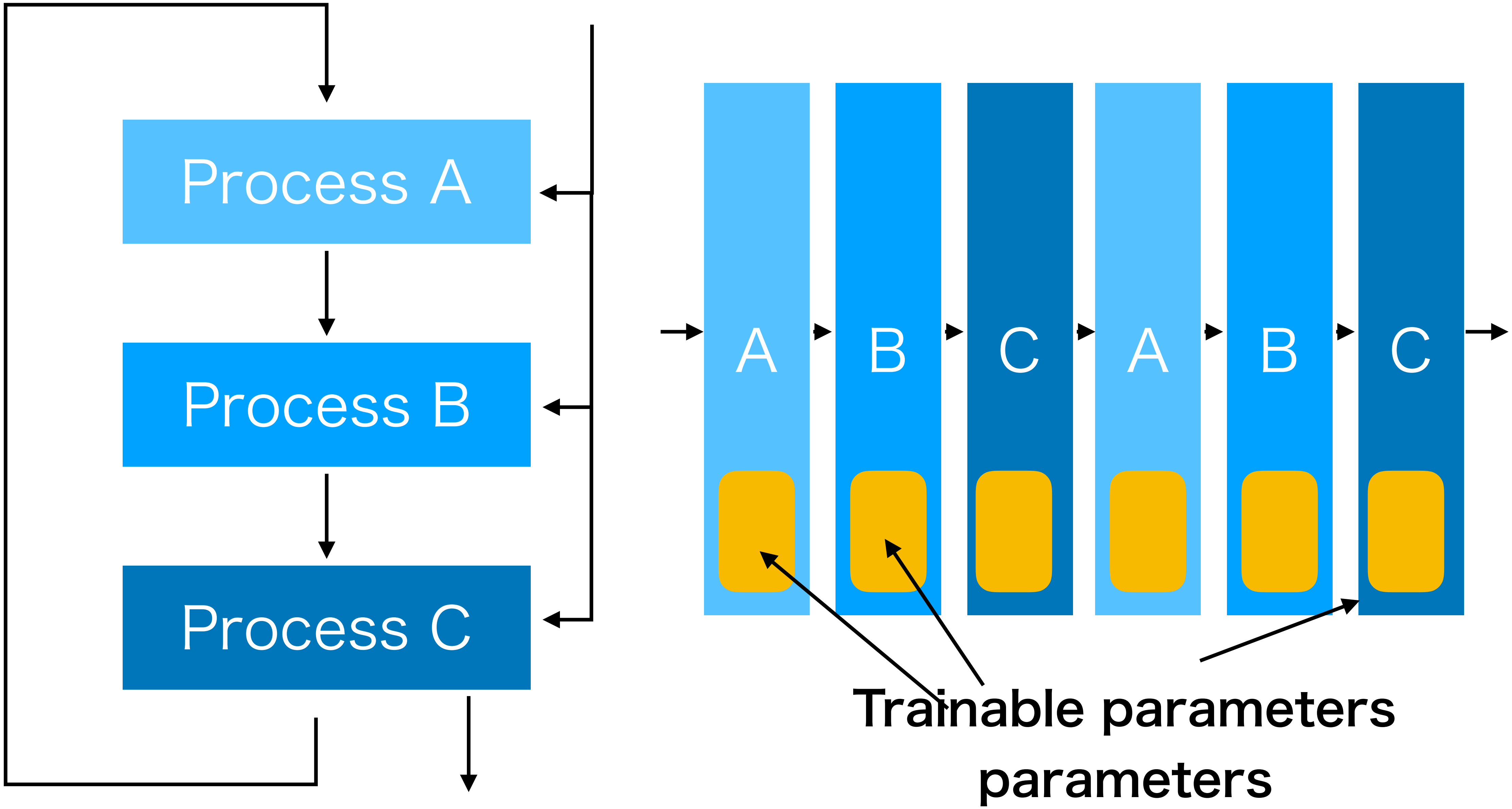


Training of deep neural networks



In a training process, the trainable parameters in Θ is optimized to minimize the value of the loss function

Deep unfolding



A recent survey: A. Balatsoukas-Stimming and C. Studer, arXiv:1906.05774, 2019.

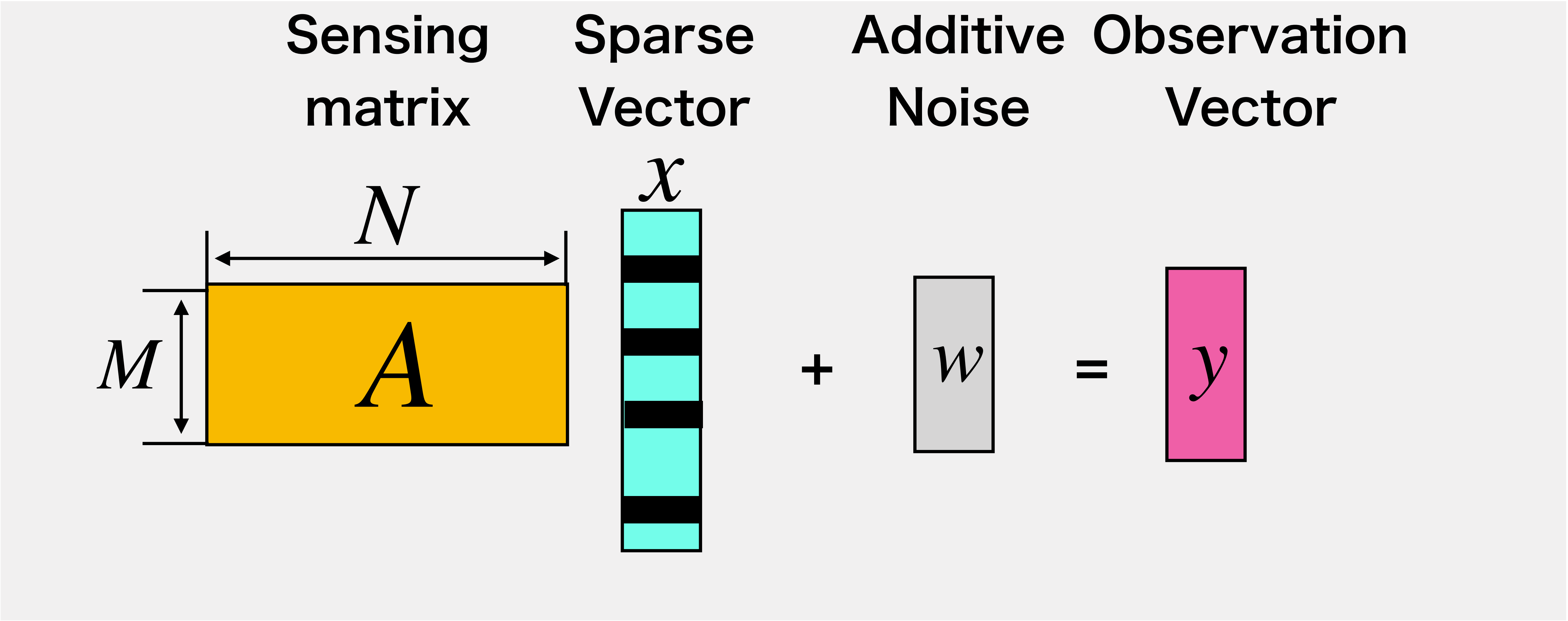
Goal and outline

Goal

To present the basic concept of deep unfolding for proximal gradient descent algorithm

- Preliminaries
  - Compressed sensing
  - Lasso
  - Proximal gradient and ISTA
- Concept of deep unfolding
  - LISTA
  - Toy examples
- Brief review of TISTA

Compressed sensing



Goal: from the knowledge of A and y, estimate the source signal x as correct as possible

# of unknown vars. > # of equations : under determined problem

Applications of CS in wireless comm.

- Channel estimation
- Angle of Arrival estimation
- Spectrum sensing
- NOMA

A User's Guide to Compressed Sensing for Communications Systems
K.HAYASHI, M. NAGAHARA and T. TANAKA
IEICE TRANS. COMMUN., VOL.E96–B, NO.3 MARCH 2013

LASSO formulation for CS

LASSO

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \frac{1}{2} ||y - Ax||_2^2 + \lambda ||x||_1$$

Squared error term

(1) Prefers a vector x satisfying y = Ax
(2) Differentiable

L1-regularization term

(1) Prefers a sparse vector
(2) non-differentiable

- **Convex problem**
- **A plain gradient descent method cannot be applied to LASSO because of the non-differentiable term**

Proximal operator

Proximal operator

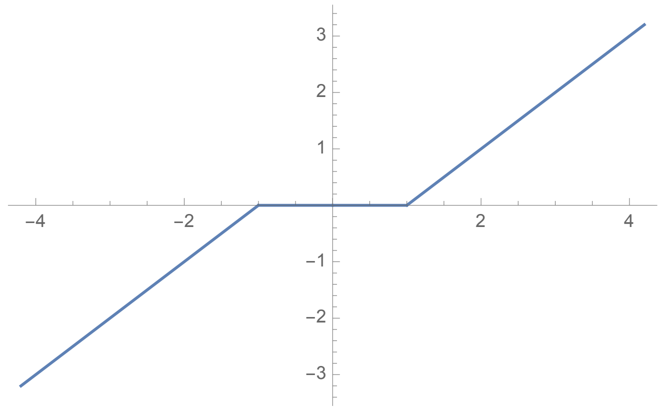
$$f : \mathbb{R}^n \to \mathbb{R}$$

$$\textbf{Prox}_{\gamma f}(x) := \arg\min_{u \in \mathbb{R}^n} \left[f(u) + \frac{1}{2\gamma} ||x - u||_2^2 \right]$$

(1) Similar to projection operation

(2) Smoothness can be added to a non-differentiable function

Example: Proximal operator of L1-norm



$$f(x) := |x| \ (f : \mathbb{R} \to \mathbb{R})$$

$$\textbf{Prox}_{\gamma f}(x) := \textbf{sign}(x) \max\{|x| - \gamma, 0\} =: \eta(x; \gamma)$$

i.e., Soft thresholding function

Proximal gradient method

Goal $$\min_{x \in \mathbb{R}^n} \left[f(x) + g(x) \right]$$

$f(x)$: **differentiable**
$g(x)$: **has simple proximal operator**

Proximal gradient method

$$x^{n+1} := \mathbf{prox}_{\gamma g}(x^n - \gamma \nabla f(x^n))$$

- $\nabla f(x^n)$: **gradient of** f
- $x^n - \gamma \nabla f(x^n)$: **gradient descent step**

ISTA: proximal gradient for LASSO

LASSO

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \frac{1}{2} ||y - Ax||_2^2 + \lambda ||x||_1$$

ISTA (Daubechies et.al, 2004)

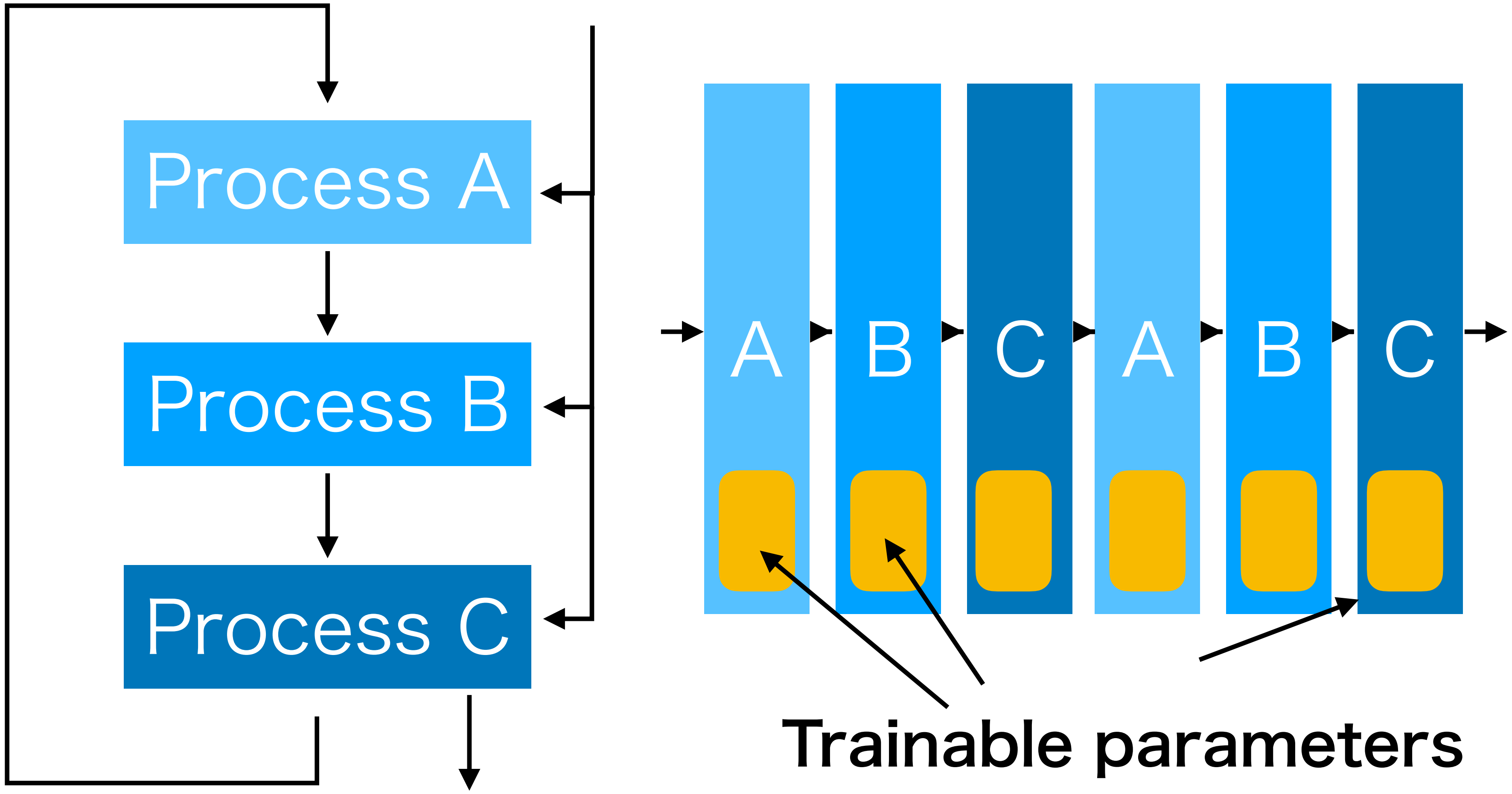
$$r_t = s_t + \beta A^T (y - A s_t)$$: gradient descent step

$$s_{t+1} := \eta(r_t; \tau)$$: proximal step

(1) simple but slow to converge

(2) choice of hyper parameters are very critical

Birth of deep unfolding



Gregor and LeCun first proposed this concept:

K. Gregor, and Y. LeCun,

``Learning fast approximations of sparse coding,''

Proc. 27th Int. Conf. Machine Learning, pp. 399--406, 2010.

Recursive formula of LISTA

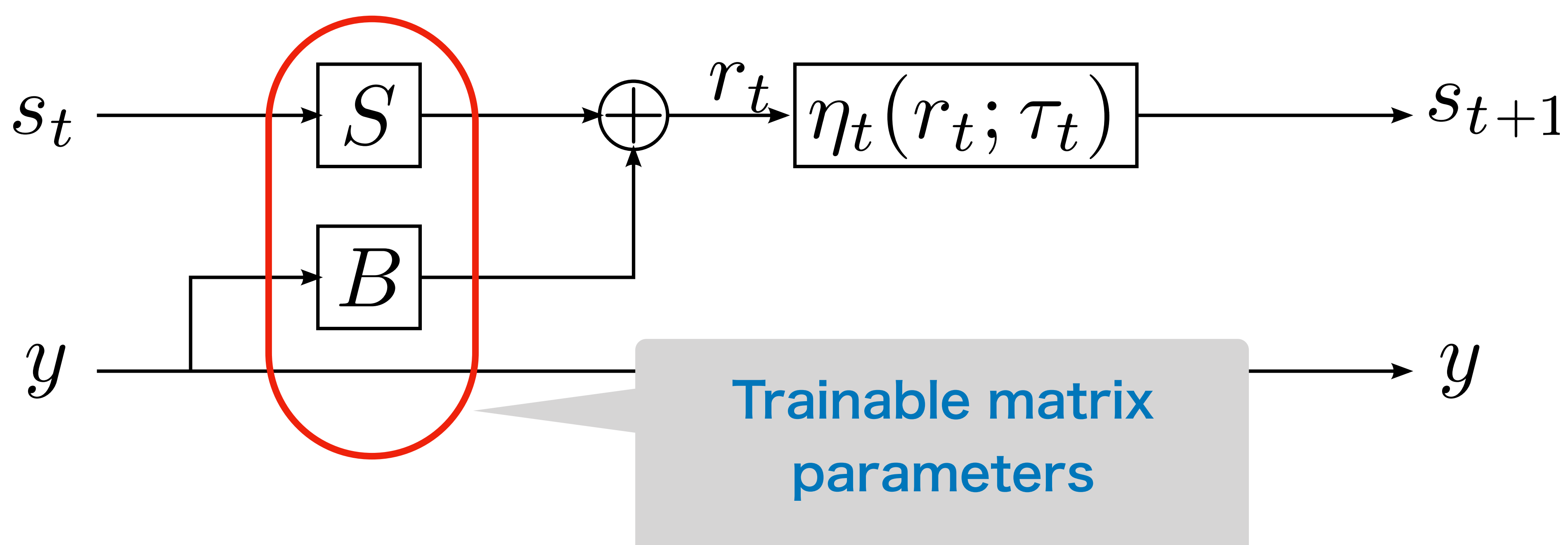
Original ISTA:

$$r_t = s_t + \beta A^T(y - As_t)$$
$$s_{t+1} := \eta(r_t; \tau)$$

LISTA

$$r_t = Bs_t + Sy$$

$$s_{t+1} = \eta(r_t; \tau_t)$$



What is main benefit of deep unfolding?

- **Simple quadratic function (convex)**
- **Naive gradient descent with random initial value**

$$f(x_1, x_2) = x_1^2 + q x_2^2$$

Gradient descent (GD)

$$\boldsymbol{s}_{t+1} = \boldsymbol{s}_t - \gamma \nabla f(\boldsymbol{s}_t)$$

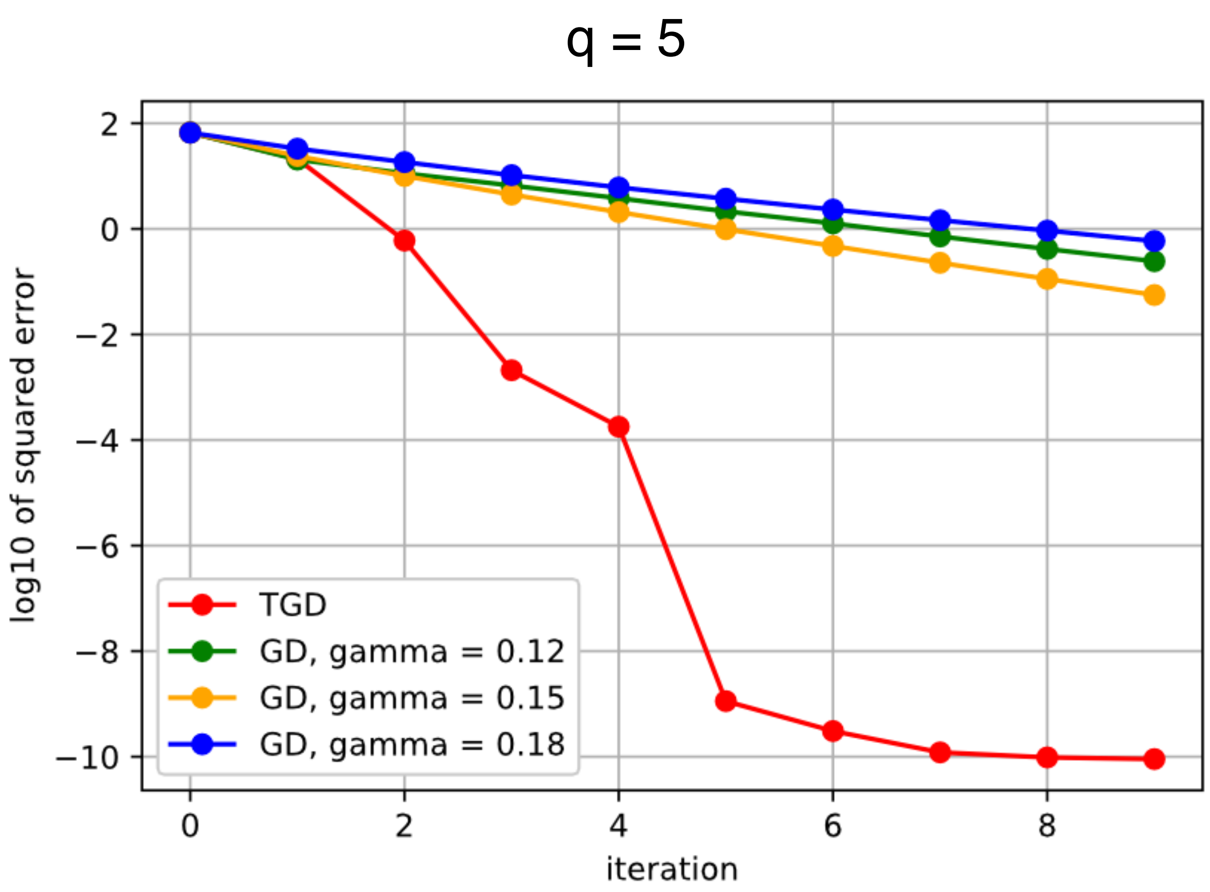
Trainable gradient descent (TGD)

$$\boldsymbol{s}_{t+1} = \boldsymbol{s}_t - \underline{\gamma_t} \nabla f(\boldsymbol{s}_t).$$

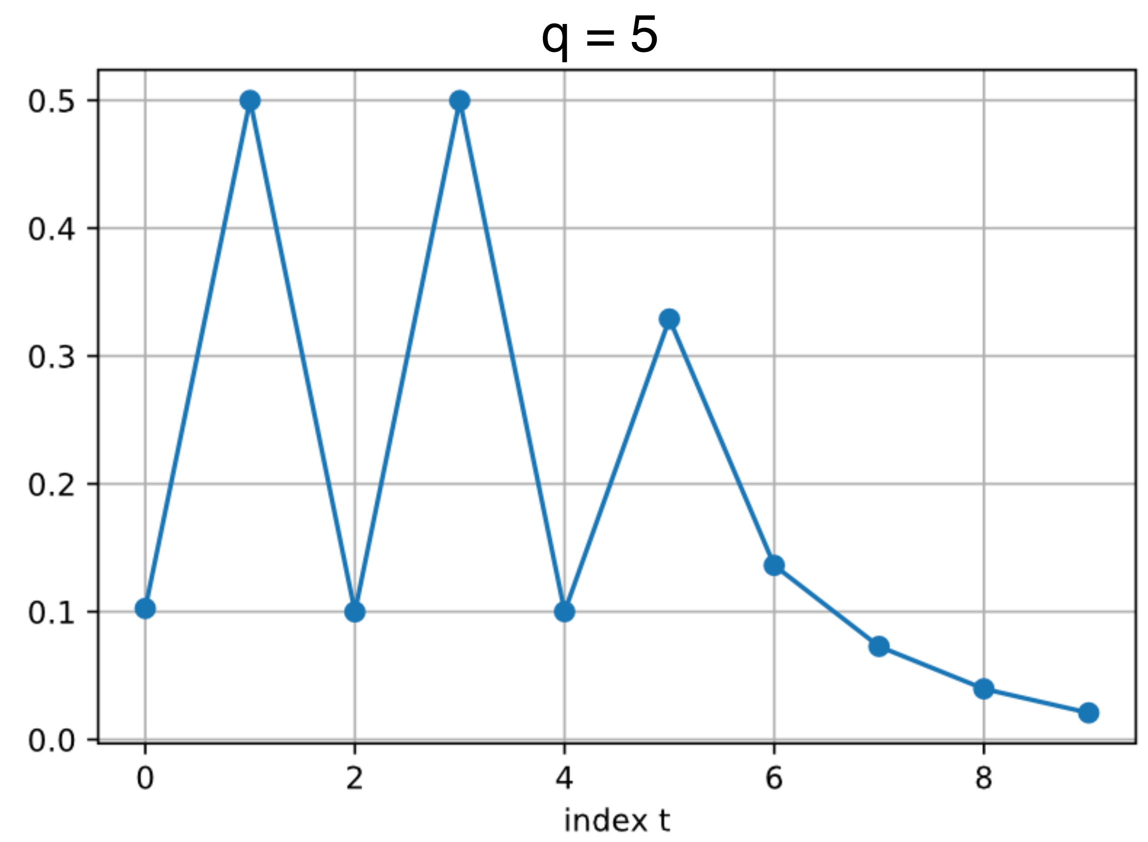
Trainable parameter

Minimization of quadratic function

Squared errors v.s. iterations



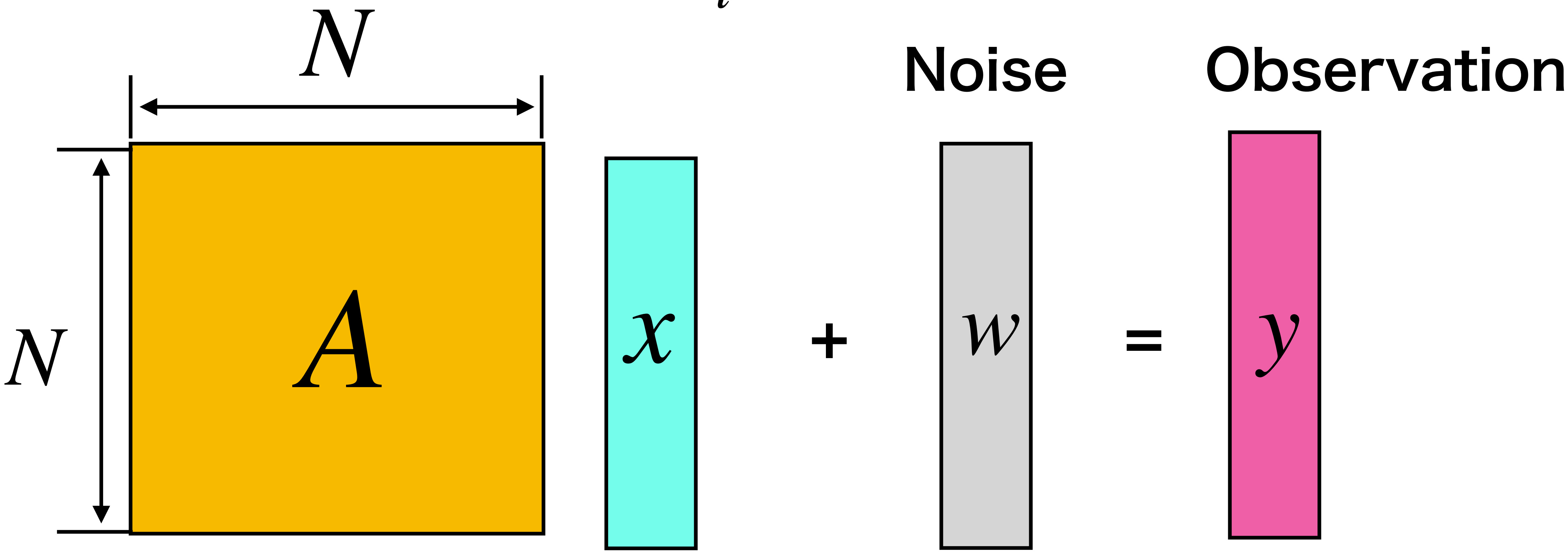
Values of γ_t



- Trainable GD shows faster convergence than GD with the fixed step size
- Iteration dependent step size is beneficial to accelerate the convergence

Deep unfolding: projected gradient

$$\forall i, -1 \leq x_i \leq 1$$



Problem: recover x from y

Projected gradient method

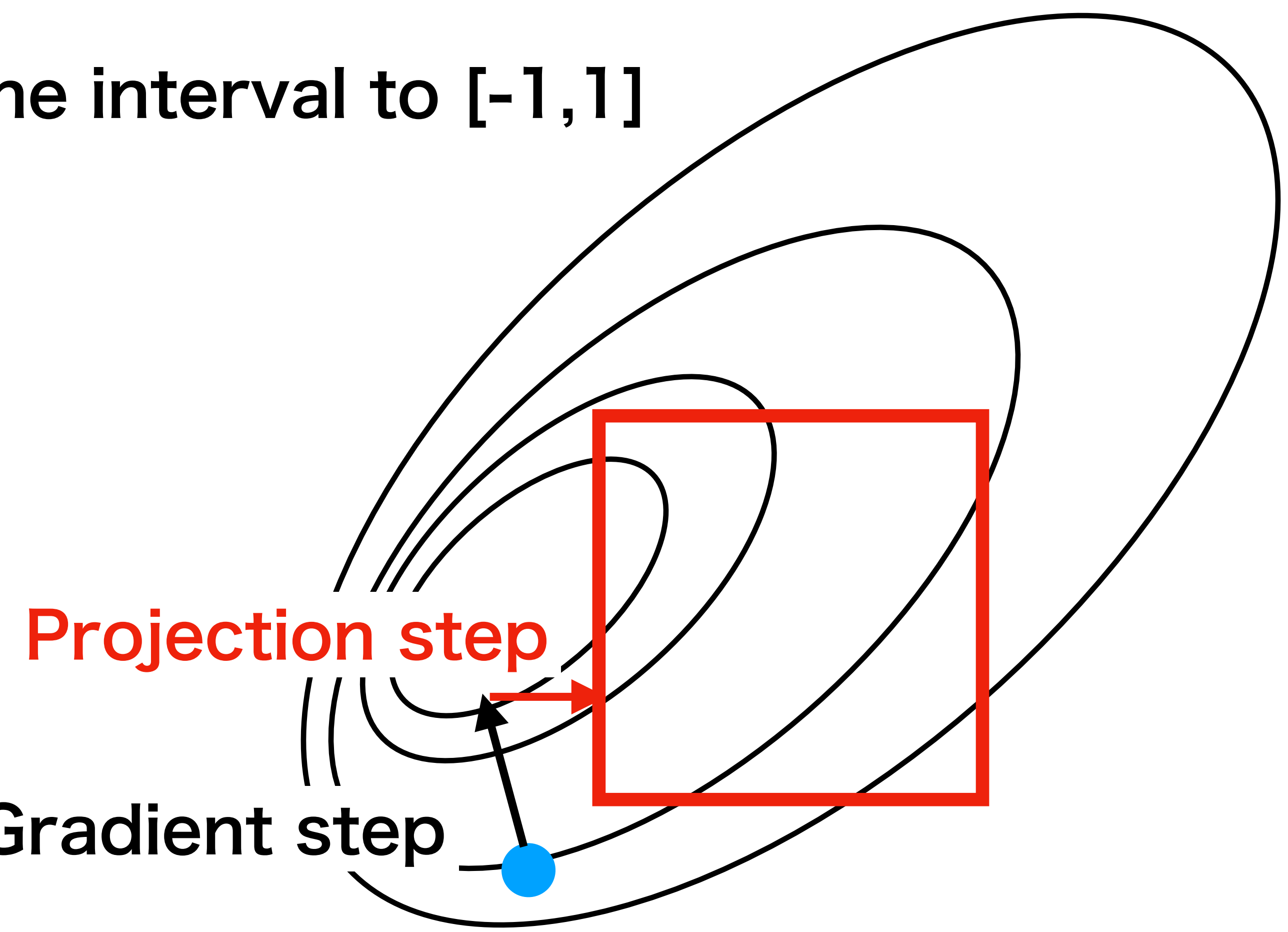
Gradient descent step

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \gamma \boldsymbol{A}^T (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t),$$

Projection step

$$\boldsymbol{s}_{t+1} = \varphi\left(\xi \boldsymbol{r}_t\right)$$

φ is the interval to [-1,1]



Experiments: Trainable projected gradient

Gradient descent step

$$\boldsymbol{r}_t = \boldsymbol{s}_t + \gamma_t \boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}_t),$$

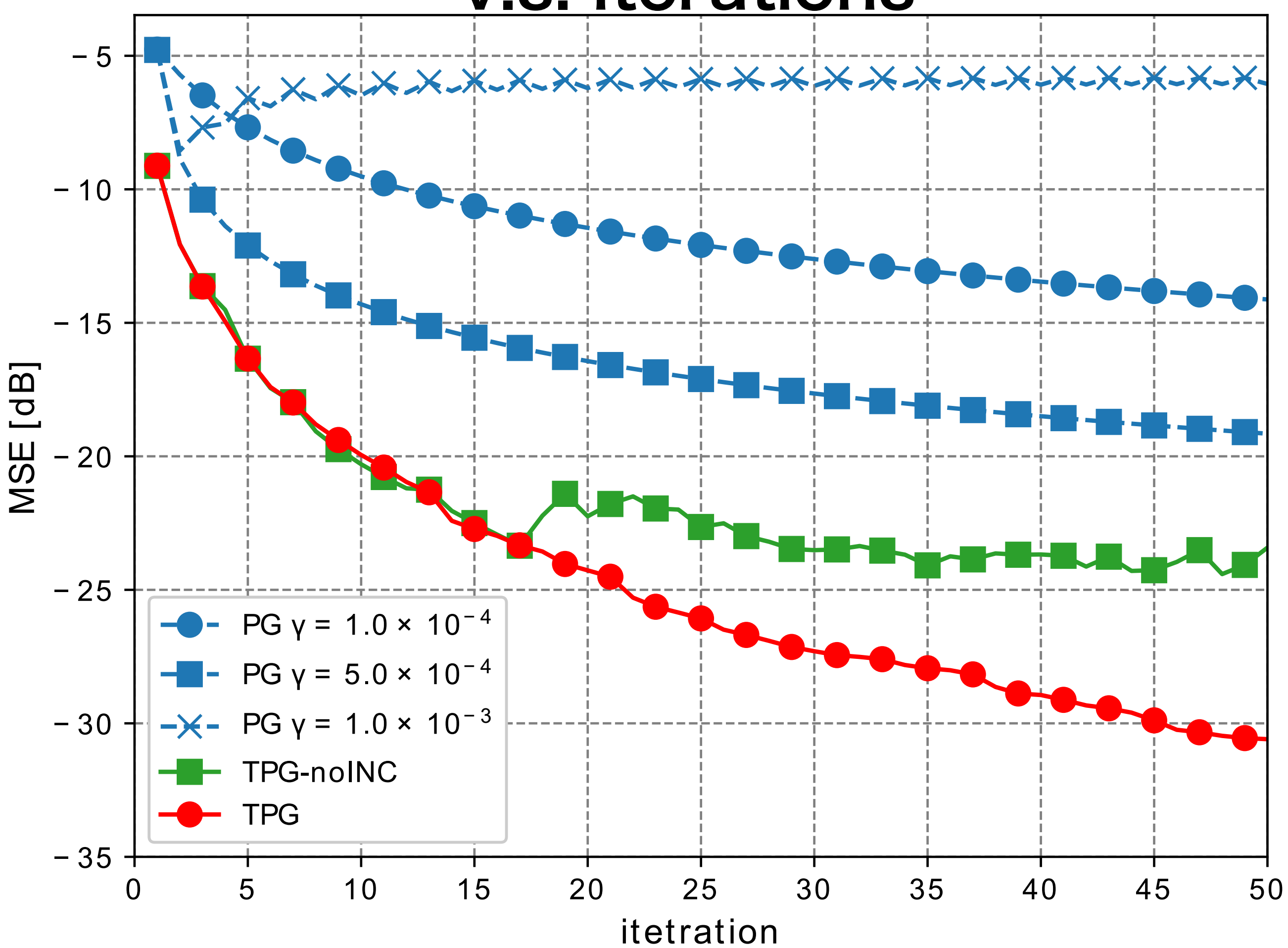
Projection step

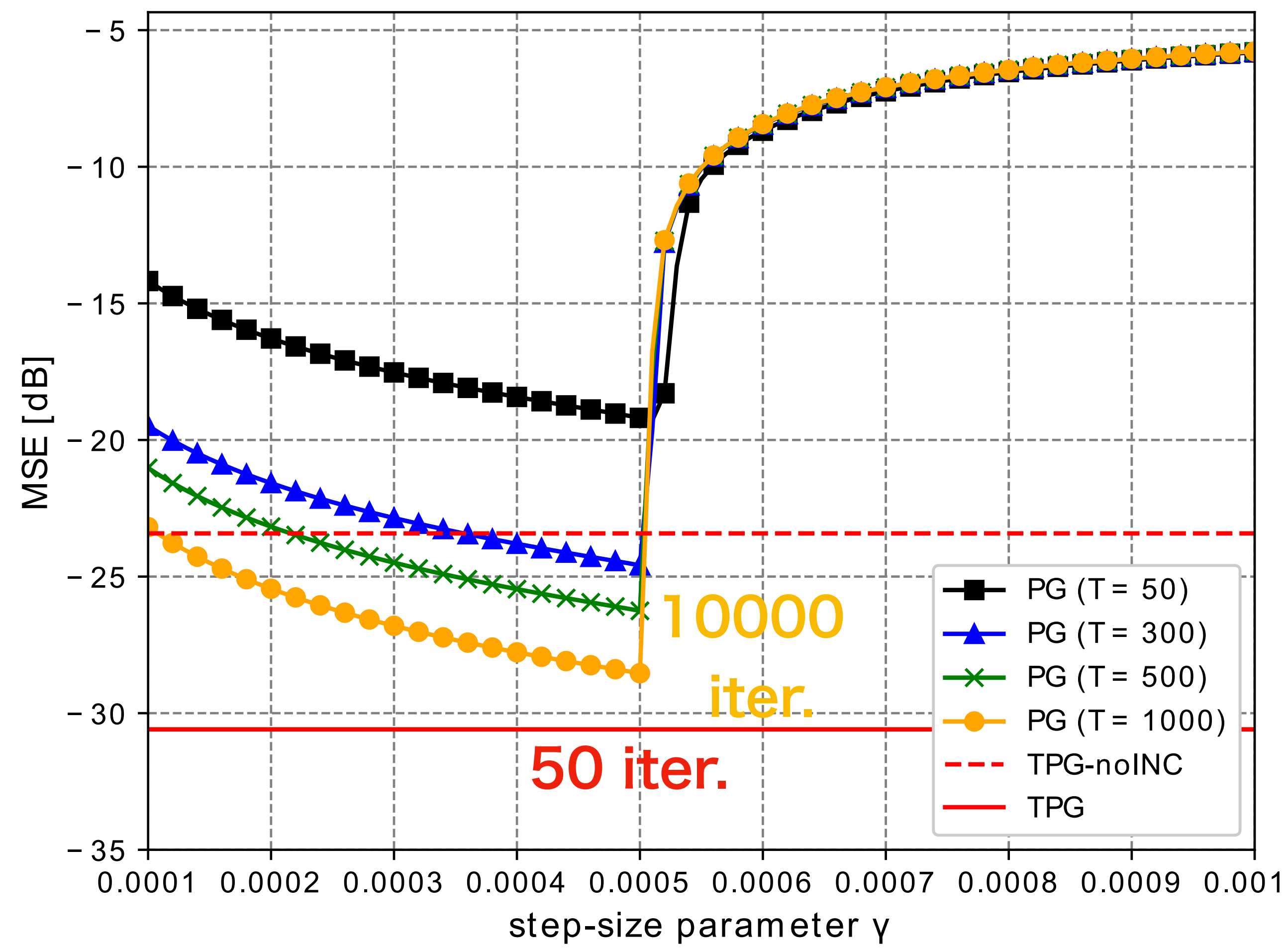
$$\boldsymbol{s}_{t+1} = \varphi\left(\xi \boldsymbol{r}_t\right),$$

Mean squared errors v.s. iterations

$n = 1000$

Mean squared errors v.s. value of gamma





TISTA (Ito, Takabe, W)

D. Ito, S. Takabe, and T. Wadayama, ``Trainable ISTA for sparse signal recovery,''
IEEE Trans. Signal Processing, vol. 67, no. 12, pp. 3113-3125, Jun., 2019.

$$
\begin{aligned}
r_t &= s_t + \gamma_t W(y - As_t), \\
s_{t+1} &= \eta_{MMSE}(r_t; \tau_t^2), \\
v_t^2 &= \max\left\{ \frac{\|y - As_t\|_2^2 - M\sigma^2}{\mathrm{trace}(A^T A)}, \epsilon \right\}, \\
\tau_t^2 &= \frac{v_t^2}{N}(N + (\gamma_t^2 - 2\gamma_t)M) + \frac{\gamma_t^2 \sigma^2}{N}\mathrm{trace}(WW^T)
\end{aligned}
$$

- W is the Moore-Penrose pseudo inverse of A
- Only scalar step size parameter becomes trainable

TISTA (Ito, Takabe, W,2019)

Trainable parameter

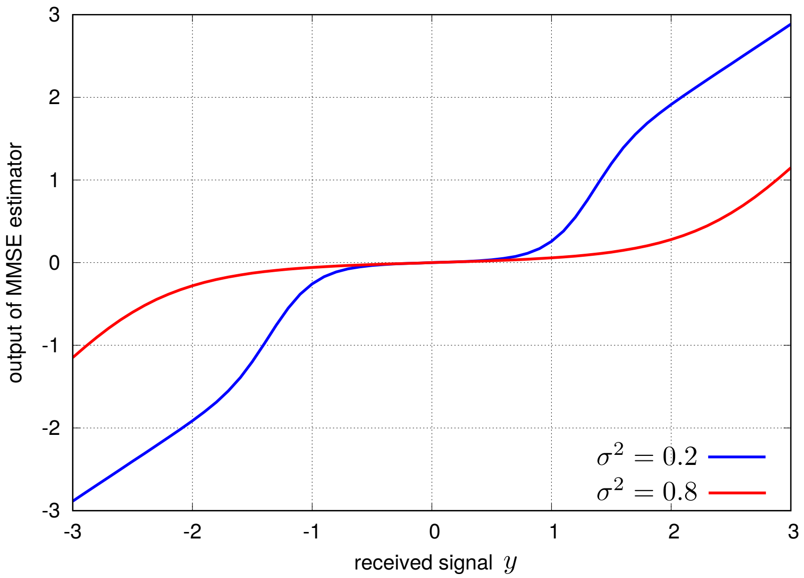
$$r_t = s_t + \gamma_t W(y - As_t),$$
$$s_{t+1} = \eta_{MMSE}(r_t; \tau_t^2),$$

Proximal gradient

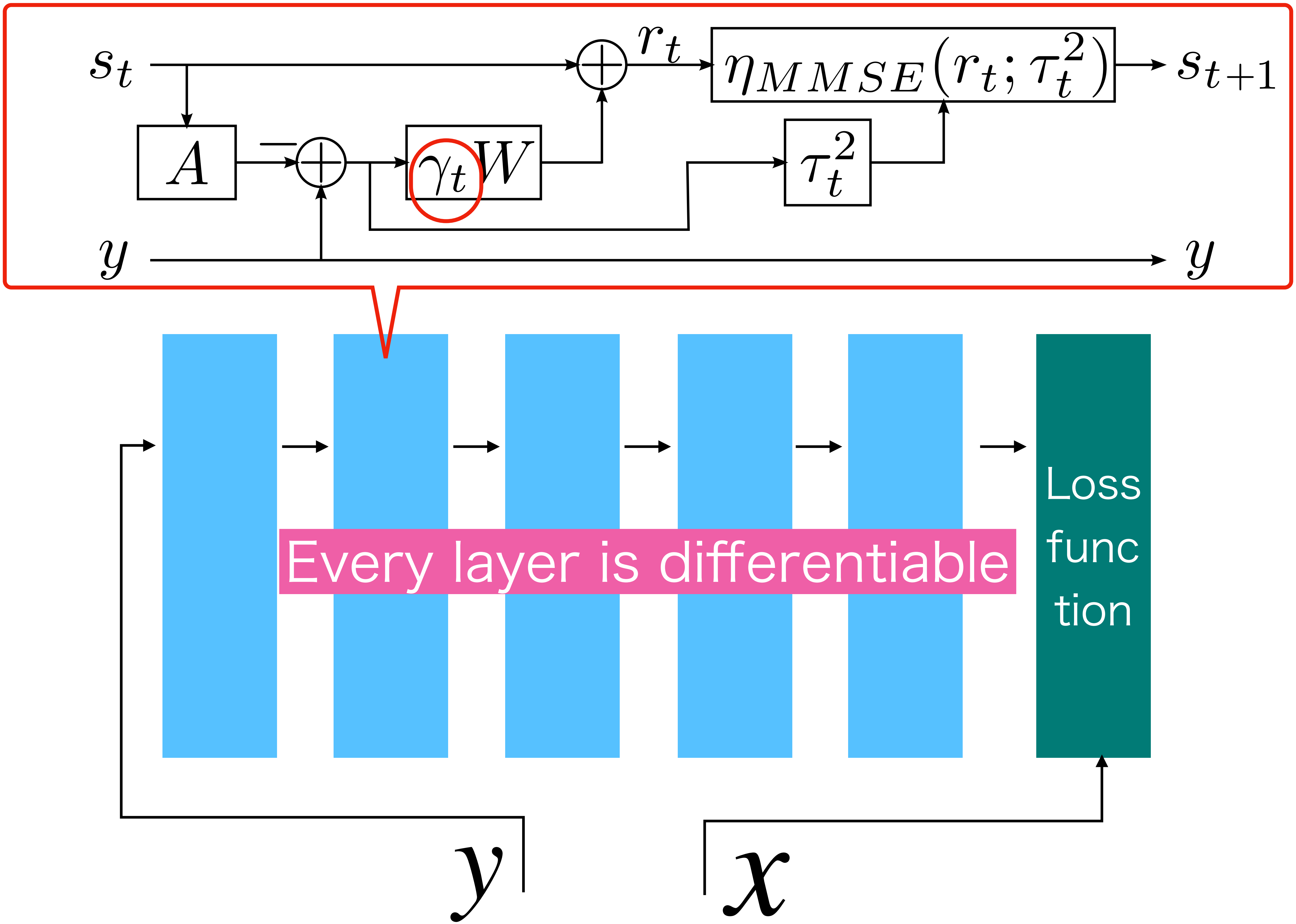
$$v_t^2 = \max\left\{\frac{\|y - As_t\|_2^2 - M\sigma^2}{\mathrm{trace}(A^T A)}, \epsilon\right\},$$
$$\tau_t^2 = \frac{v_t^2}{N}(N + (\gamma_t^2 - 2\gamma_t)M) + \frac{\gamma_t^2\sigma^2}{N}\mathrm{trace}(WW^T)$$

Estimation of error variance

Proximal mapping based on MMSE estimator



An iteration of TISTA



Recovery performance of TISTA

10% of elements in a sparse vector are nonzero

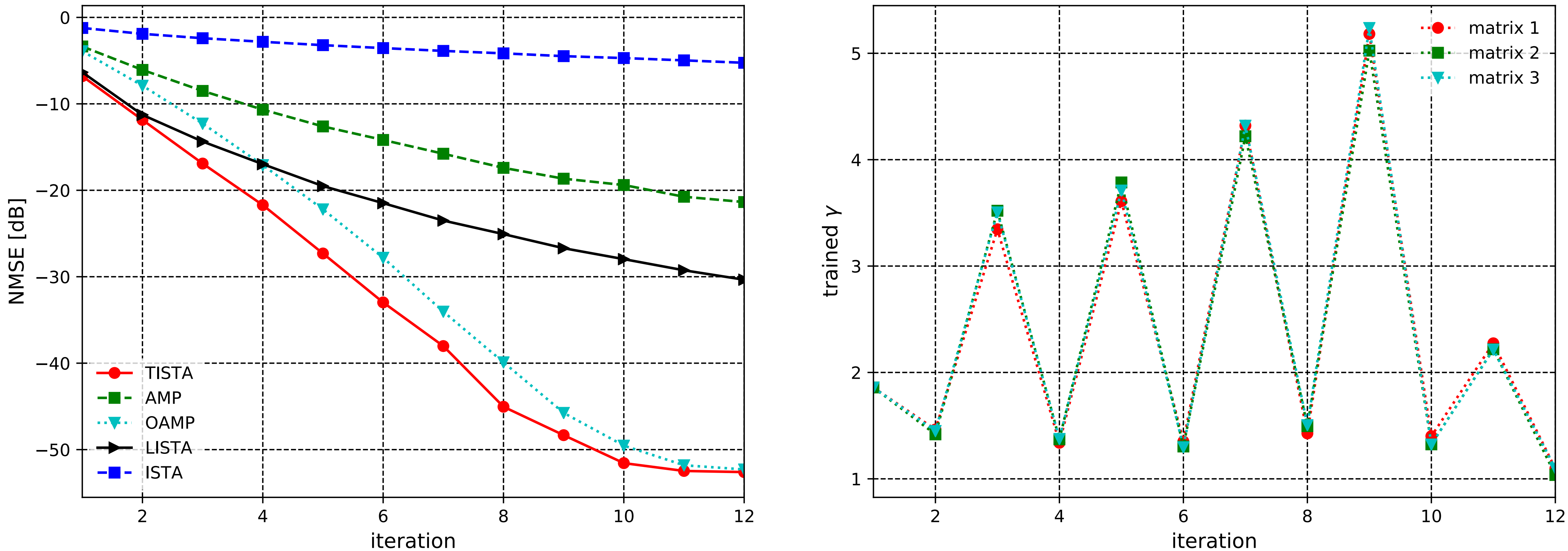


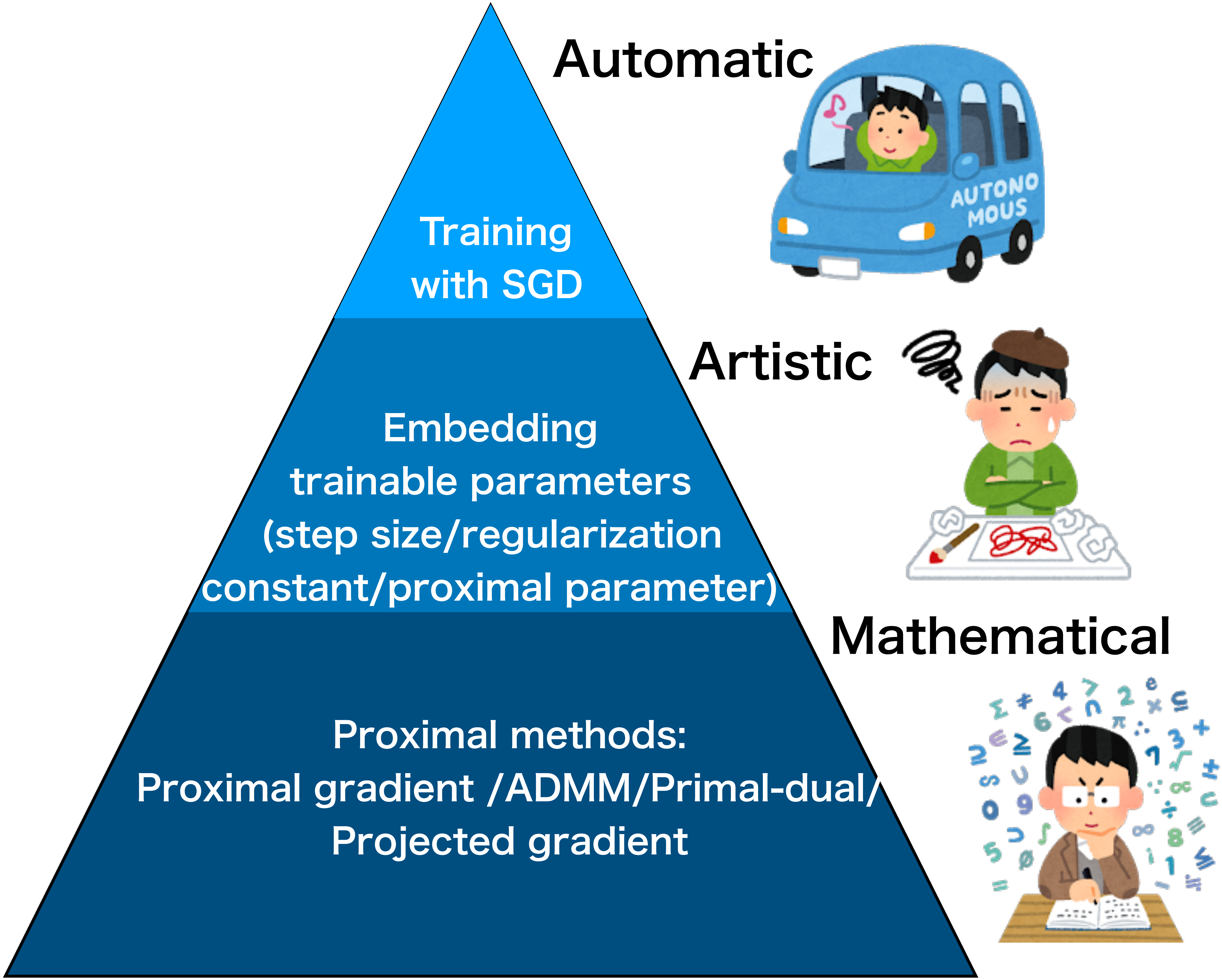
Figure 7: Sparse signal recovery for $(n, m) = (300, 150)$, $p = 0.1$ and SNR$= 50$ dB. (Left) MSE performances as a function of iteration steps. (Right) Trained values of γ_t under three different mesurement matrices. The initial value is set to 1.0.

Number of trainable parameters

	TISTA	LISTA	LAMP
# of params	T	$T(N^2 + MN + 1)$	$T(NM + 2)$

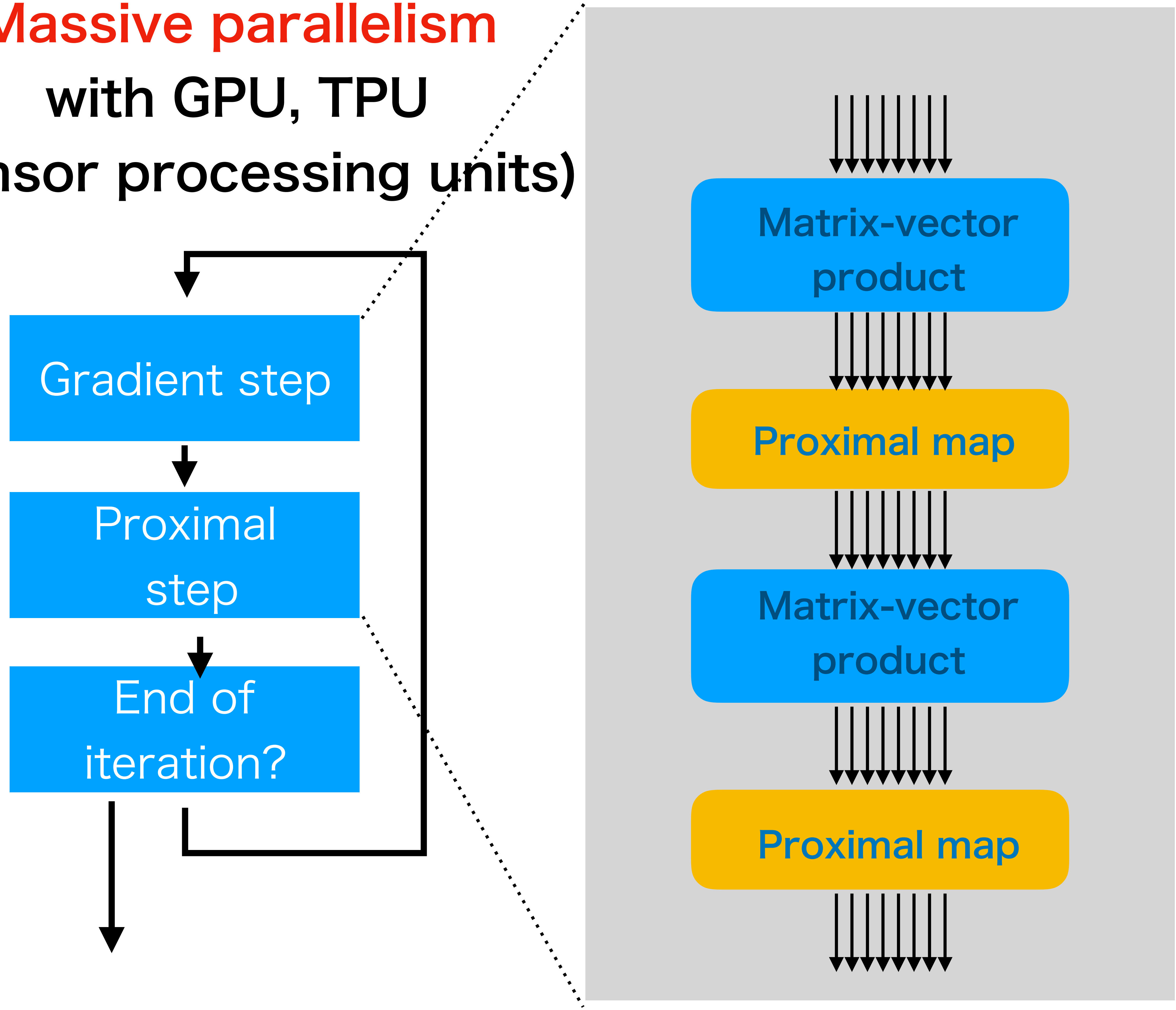
[2] M. Borgerding and P. Schniter, "Onsager-corrected deep learning for sparse linear inverse problems," *2016 IEEE Global Conf. Signal and Inf. Proc. (GlobalSIP)*, Washington, DC, Dec. 2016, pp. 227-231.

Deep unfolding for proximal algorithms



``Neural-net friendly" hardware

Massive parallelism with GPU, TPU (Tensor processing units)



Our contributions based on TISTA

Currently ranked in popular Items 50 of IEEE Trans.on Signal Processing

Name of Algorithm	Journal/Conference
Trainable ISTA (TISTA) for sparse signal recovery	ICC Workshop 2018 IEEE Trans. on Signal Processing, 2019
Trainable projected gradient detector for massive overloaded MIMO	ICC 2019, IEEE Access (accepted) , 2019
Complex-field TISTA(MIMO, OFDM, CS…)	Globecom 2019 (submitted)
Trainable projected gradient decoder for LDPC codes	ISIT 2019 (to be presented in next week)

On going projects:

Sparse CDMA (simplest NOMA)

Quantized MIMO,　Phase retrieval

Combinatorial optimization based on proximal methods

Conclusion

- Concept of deep unfolding is introduced
- Ideas of compressed sensing, LASSO, ISITA, proximal gradient descent are explained
- TISTA for sparse signal recovery
  - Fast: Shows fastest convergence among most of known sparse signal recovery algorithms
  - Simple: Number of trainable parameters is small
  - Massive parallelism: Suitable for neural-net friendly hardware (like GPU, TPU)

Model-based v.s. Blackbox-based

	Model-based approach	DNN-based blackbox approach
Pros:	Domain knowledge can be used Many algorithms induced from mathematical principles	No domain knowledge needed Powerful new algorithm can be constructed
Cons:	How can we improve known algorithms further ?	Interpretability problem Scalability

Model-based v.s. Blackbox-based

Deep unfolding

	Model-bas approach	DNN-based blackbox approach
Pros:	Domain knowledge can be used Many algorithms induced from mathematical principles	No domain knowledge needed Powerful new algorithm can be constructed
Cons:	How can we improve known algorithms further ?	Interpretability problem Scalability

TISTA vs OAMP (1)

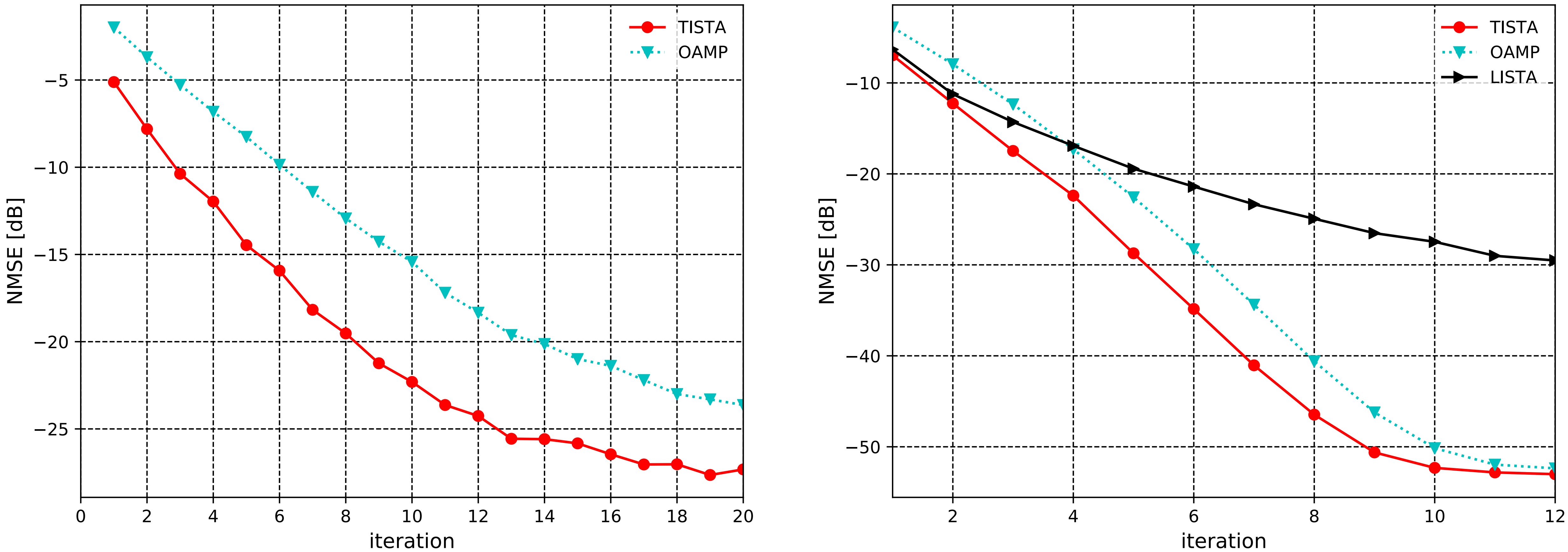


Figure 8: (Left) MSE performance of OAMP and TISTA for $(n, m) = (300, 150)$, $p = 0.18$ and SNR$=$ 30 dB. (Right) MSE performances for $(n, m) = (300, 150)$, $p = 0.1$ and SNR$=$ 30 dB with binary sensing matrix $\boldsymbol{A} \in \{1, -1\}^{m \times n}$ as a function of iteration steps.

TISTA vs OAMP (2)