Jointly Sparse Signal Recovery and Support Recovery via Deep Learning with Applications in MIMO-based Grant-Free Random Access

Ying Cui

Department of Electrical Engineering
Shanghai Jiao Tong University

September 22, 2020

Outline

Outline

Jointly sparse signal or support recovery

- Jointly sparse signal or support recovery in Multiple Measurement Vector (MMV) models refers to the estimation of M jointly sparse N-dimensional signals or the common support, respectively, from L ($\ll N$) limited noisy linear measurements based on a common measurement matrix
- Two main challenges:
  - Design a common measurement matrix which maximally retains the information on sparse signals or their common support when reducing signal dimension
  - Recover the jointly sparse signal or the common support with high recovery accuracy and short computation time

Grant-free random access

- Grant-free random access has been widely regarded as one promising solution for supporting massive machine-type communications (mMTC) for Internet of Things (IoT)
- In mMTC, there are massive IoT devices in each cell, but only a small number of devices are active at a time
- In grant-free random access, each device is assigned a specific pilot sequence, all active devices send their pilot sequences, and each base station (BS) detects the activities of its associated devices or estimates their channel states

Channel estimation and device activity detection in MIMO-based grant-free random access

- Key applications of the jointly sparse signal and support recovery include channel estimation and device activity detection in MIMO-based grant-free random access
- Common measurement matrix design, jointly sparse signal recovery and jointly sparse support recovery for complex signals correspond to design of pilot sequences, channel estimation and device activity detection in MIMO-based grant-free random access
  - related work: [Senel & Larsson (2018); Liu & Yu (2018)] and device activity detection [Haghighatshoar et al. (2018); Chen et al. (2019); Jiang & Cui (2020); Liu et al. (2018); Chen et al. (2018)]
- Our primary goals:
  - Address the aforementioned two challenges in jointly sparse signal recovery and sparse support recovery for complex signals
  - Provide practical solutions with high recovery accuracy and short computation time for channel estimation and device activity detection in MIMO-based grant-free random access

Previous work

- Most existing works on sparse support recovery for SMV models and jointly support recovery for MMV models focus on investigating support recovery methods for a given measurement matrix
  - Optimization-based methods: LASSO [Wainwright (2009); Pal & Vaidyanathan (2015)], ML estimation [Haghighatshoar et al. (2018); Chen et al. (2019)] and MAP estimation [Jiang & Cui (2020)]
  - Iterative thresholding methods: AMP [Liu et al. (2018); Chen et al. (2018)] and EM-AMP [Ke et al. (2020); Wei et al. (2016)]
- Sparse signal recovery for SMV models and jointly sparse signal recovery for MMV models are more widely investigated, but still mainly for a given measurement matrix
  - Optimization-based methods: LASSO [Tibshirani (1996)] and GROUP LASSO [Qin et al. (2013)]
  - Iterative thresholding methods: AMP [Donoho et al. (2009); Senel & Larsson (2018); Liu & Yu (2018); Ziniel & Schniter (2012)] and EM-AMP [Ke et al. (2020); Wei et al. (2016)]

Previous work

- LASSO [Wainwright (2009); Pal & Vaidyanathan (2015); Tibshirani (1996)] and GROUP LASSO [Qin et al. (2013)] do not rely on any information of signals or noise
- MAP, AMP and EM-AMP all assume that the components of signals are independent, and hence their recovery performance may be unsatisfactory when such assumption is not satisfied
- Neural networks are recently utilized to exploit properties of sparse signals from training samples, for the purpose of designing effective sparse signal recovery methods when the components of signals are correlated and prior distributions do not have analytic models
  - Some works exploit properties of sparsity patterns of real signals [Gregor & LeCun (2010); Yao et al. (2017); He et al. (2018); Wu et al. (2019)]
  - Some works exploit properties of sparsity patterns of complex signals [Taha et al. (2019); Yang et al. (2018); Zhang et al. (2019)]

Previous work

- Very few papers [Candes (2008); Eldar et al. (2010)] investigate the impact of the measurement matrix in sparse signal recovery
  - The authors in [Candes (2008)] show that a measurement matrix can preserve sparsity information in sparse signals, if it satisfies the restricted isometry property (RIP)
  - The authors in [Eldar et al. (2010)] consider group sparse signals and show that block-coherence and sub-coherence of a measurement matrix affect signal recovery performance
  - The results in [Candes (2008); Eldar et al. (2010)] may not hold for sparse support recovery

Previous work

- Joint design of signal compression and recovery methods for real signals [Wu et al. (2019); Nguyen et al. (2017); Mousavi et al. (2017); Adler et al. (2017)] or complex signals [Wen et al. (2018); Li et al. (2019)] using deep auto-encoders can significantly improve recovery performance
  - All these works are all for SMV models, and their extensions to MMV models are highly nontrivial
  - Neither the neural network for complex signals in [Wen et al. (2018)] nor direct extensions of the neural networks for real signals in [Wu et al. (2019); Nguyen et al. (2017); Mousavi et al. (2017); Adler et al. (2017)] to complex signals can achieve linear compression for complex signals
- How to jointly design the common measurement matrix and jointly sparse signal or support recovery methods in MMV models for complex signals remains open

Outline

Joint sparse signal/support recovery in MMV models

- The support of sparse signal $\mathbf{x} \in \mathbb{C}^N$ is the set of locations of non-zero elements of $\mathbf{x}$:

$$\mathrm{supp}(\mathbf{x}) \triangleq \{n \in \mathcal{N} | x_n \neq 0\}$$

  where $\mathcal{N} \triangleq \{1, \cdots, N\}$

- $\mathbf{x}$ is sparse if the number of non-zero elements of $\mathbf{x}$ is much smaller than its total number of elements, i.e., $|\mathrm{supp}(\mathbf{x})| \ll N$
- M signals $\mathbf{x}_m \in \mathbb{C}^N, m \in \mathcal{M}$ are jointly sparse if they share a common support $\mathcal{S} \triangleq \mathrm{supp}(\mathbf{x}_m), m \in \mathcal{M}$, where $\mathcal{M} \triangleq \{1, \cdots, M\}$
- Consider $L \ll N$ noisy linear measurements $\mathbf{y}_m \in \mathbb{C}^L$ of $\mathbf{x}_m$:

$$\mathbf{y}_m = \mathbf{A}\mathbf{x}_m + \mathbf{z}_m, \quad m \in \mathcal{M}$$

  - $\mathbf{A} \in \mathbb{C}^{L \times N}$ is the (known) common measurement matrix
  - $\mathbf{z}_m \sim \mathcal{CN}(\mathbf{0}_{L \times 1}, \sigma^2 \mathbf{I}_{L \times L})$ is the additive white Gaussian noise

Joint sparse signal/support recovery in MMV models

- The compact form of noisy linear measurements is:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{Z}$$

  - $\mathbf{X} \in \mathbb{C}^{N \times M}$ with $\mathbf{X}_{:,m} = \mathbf{x}_m, m \in \mathcal{M}$
  - $\mathbf{Y} \in \mathbb{C}^{L \times M}$ with $\mathbf{Y}_{:,m} = \mathbf{y}_m, m \in \mathcal{M}$
  - $\mathbf{Z} \in \mathbb{C}^{L \times M}$ with $\mathbf{Z}_{:,m} = \mathbf{z}_m, m \in \mathcal{M}$
- Jointly sparse signal recovery in MMV models aims to estimate $\mathbf{X}$ from $\mathbf{Y}$
- Jointly sparse support recovery in MMV models aims to estimate $\mathcal{S}$ from $\mathbf{Y}$

Application: channel estimation and device activity detection in MIMO-based grant-free random access

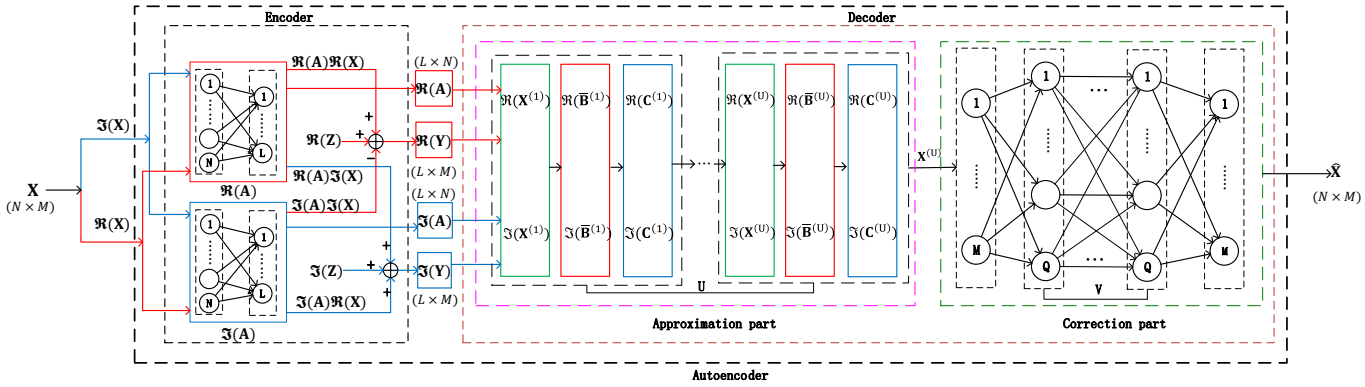
- Consider MIMO-based grant-free random access in mMTC with one M-antenna BS and N single-antenna devices
- Device-activity patterns for IoT traffic are typically sporadic
- $\mathbf{x}_m \in \mathbb{C}^N, m \in \mathcal{M}$, with $x_m(n) = \alpha(n)h_m(n)$, are jointly sparse with a common support $\mathcal{S} = \{n \in \mathcal{N} | \alpha(n) = 1\}$
  - $\alpha(n) \in \{0, 1\}$ denotes the active state of device n
  - $h_m(n) \in \mathbb{C}$ denotes the complex channel between the m-th antenna at the BS and device n
- Each device n has a unique pilot sequence $\mathbf{a}_n \in \mathbb{C}^L$, and $\mathbf{A} \in \mathbb{C}^{L \times N}$ with $\mathbf{A}_{:,n} = \mathbf{a}_n, n \in \mathcal{N}$ is known at the BS
- In the pilot transmission phase, active devices synchronously send their pilot sequences to the BS, and $\mathbf{Y} = \mathbf{AX} + \mathbf{Z}$ represents the received signal at the BS
  - Channel estimation, i.e., estimating $\mathbf{X}$ from $\mathbf{Y}$ corresponds to jointly sparse signal recovery in MMV models
  - Device activity detection, i.e., estimating $\mathcal{S}$ from $\mathbf{Y}$, corresponds to jointly sparse support recovery in MMV models

Outline

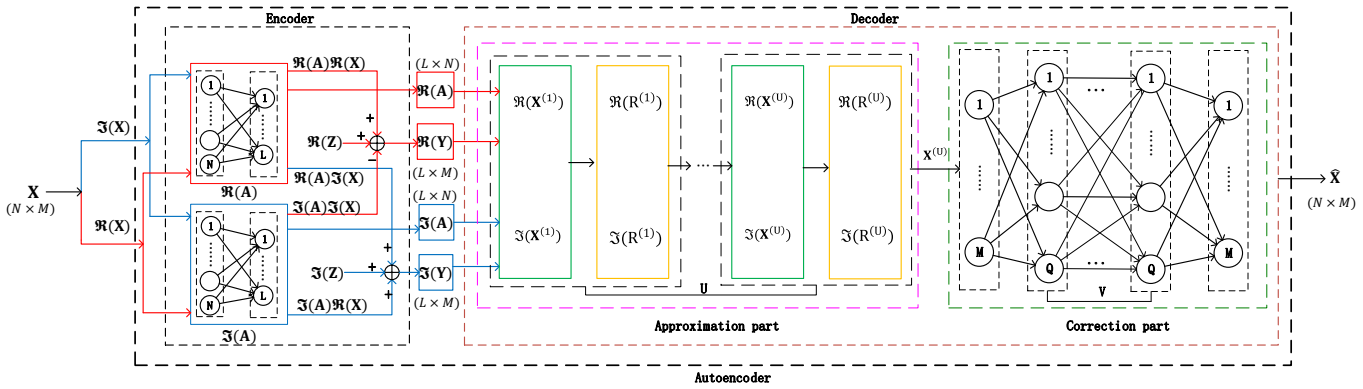
A model-driven approach for jointly sparse signal recovery

- Jointly design the common measurement matrix and jointly sparse signal recovery method for jointly sparse complex signals
- Utilize the standard auto-encoder structure for real numbers in deep learning
- Consist of an encoder and a model-driven decoder
  - The encoder mimics noisy linear measurement process
  - The model-driven decoder mimics jointly sparse signal recovery process via an approximation part and a correction part
    - The approximation part is used to approximate a particular method for jointly sparse signal recovery
    - The correction part aims to reduce the approximation error
- After training, obtain the common measurement matrix by extracting the weights of the encoder, and directly use the model-driven decoder for jointly sparse signal recovery
  - Should be jointly utilized

Two instances



(a) Proposed approach with GROUP LASSO-based decoder



(b) Proposed approach with AMP-based decoder

Figure: Proposed model-driven approach for jointly sparse signal recovery

Encoder for jointly sparse signal recovery

- Mimic the noisy linear measurement process for complex signals based on the two equations for real numbers:

$$\Re(\mathbf{Y}) = \Re(\mathbf{A})\Re(\mathbf{X}) - \Im(\mathbf{A})\Im(\mathbf{X}) + \Re(\mathbf{Z})$$
$$\Im(\mathbf{Y}) = \Im(\mathbf{A})\Re(\mathbf{X}) + \Re(\mathbf{A})\Im(\mathbf{X}) + \Im(\mathbf{Z})$$

- Consist of two fully-connected two-layer neural networks to implement the multiplications with $\Re(\mathbf{A}) \in \mathbb{R}^{L\times N}$ and $\Im(\mathbf{A}) \in \mathbb{R}^{L\times N}$, respectively
  - The input layer has N neurons
  - The output layer has L neurons
  - The weight of the connection from the n-th neuron in the input layer to the l-th neuron in the output layer corresponds to the (l, n)-th element of $\Re(\mathbf{A})$ or $\Im(\mathbf{A})$
  - No activation functions are used in the output layer
  - The elements of $\Re(\mathbf{Z}) \in \mathbb{R}^{L\times M}$ and $\Im(\mathbf{Z}) \in \mathbb{R}^{L\times M}$ are generated independently according to $\mathcal{N}(0, \frac{\sigma^2}{2})$
- Extend the one for SMV models [Li et al. (2019)]

Model-driven decoder for jointly sparse signal recovery

- Mimic the jointly sparse signal recovery process of a particular method
- Consist of an approximation part and a correction part
  - The approximation part uses U (≥ 0) building blocks to implement U iterations of the iterative algorithm
    - Input $\mathbf{Y}$ and $\mathbf{A}$, and output $\mathbf{X}^{(U)}$
    - The difference between $\mathbf{X}^{(U)}$ and the actual $\mathbf{X}$ generally decreases with U
  - The correction part uses V (≥ 1) fully connected layers to reduce the difference between $\mathbf{X}^{(U)}$ and $\mathbf{X}$
    - Input $\mathbf{X}^{(U)}$, and output $\hat{\mathbf{X}}$
    - In each of the first $V-1$ correction layers, choose rectified linear unit (ReLU) as the activation function
    - In the last correction layer, no activation functions are used
    - V influences the training error and generalization error
  - U and V are jointly chosen to achieve higher recovery accuracy and shorter computation time than the underlying method
    - Degrade to a particular method (sufficiently large U and $V = 0$) or a purely data-driven method ($U = 0$ and $V > 0$)

GROUP LASSO-based decoder

- GROUP LASSO is an optimization-theoretic formulation of the jointly sparse signal recovery problem in MMV models:

$$\min_{\mathbf{X}} \ \frac{1}{2}|||\mathbf{AX}-\mathbf{Y}|||_F^2+\lambda\sum_{n\in\mathcal{N}}\|\mathbf{X}_{n,:}\|_2$$

  - $\lambda \geq 0$ is a regularization parameter influencing the signal recovery accuracy
  - Do not rely on any information of sparse signals or noise
  - Reduce to LASSO when $M=1$
- GROUP LASSO is convex and can be solved optimally
  - The gradient descent method has computational complexity $\mathcal{O}(N^2M)$, which is prohibitively high at large N
  - The block coordinate descent method [Qin et al. (2013)] has computational complexity $\mathcal{O}(LNM)$
    - The sequential operations cannot make use of the parallelizable neural network architecture and have unsatisfactory computation time

GROUP LASSO-based Decoder

- We develop a fast ADMM algorithm of computational complexity $\mathcal{O}(LNM)$
  - The parallel operations make use of the parallelizable neural network architecture and significantly reduce computation time
  - $\mathbf{X}^{(k)}$ converges to an optimal solution as $k \to \infty$
  - Two parameters $\lambda > 0$ and $\rho > 0$ influence the recovery accuracy and convergence speed, respectively

Algorithm 1 ADMM for GROUP LASSO

1: Set $\mathbf{X}^{(0)} = \mathbf{0}_{N\times M}$, $\mathbf{B}^{(0)} = \mathbf{0}_{N\times M}$, $\mathbf{C}^{(0)} = \mathbf{0}_{N\times M}$ and $k = 0$.
2: **repeat**
3: For all $n \in \mathcal{N}$, compute $\mathbf{X}_{n,:}^{(k+1)} = \max\left\{1 - \frac{\lambda}{\rho\|\mathbf{t}^{(k)}\|_2}, 0\right\}\mathbf{t}^{(k)}/\mathbf{A}_{:,n}^H\mathbf{A}_{:,n}$, where $\mathbf{t}^{(k)} = \mathbf{A}_{:,n}^H\left(\mathbf{A}_{:,n}\mathbf{X}_{n,:}^{(k)} + \overline{\mathbf{B}}^{(k)} - \overline{\mathbf{A}\mathbf{X}}^{(k)} - \mathbf{C}^{(k)}\right)$.
4: Update $\overline{\mathbf{B}}^{(k+1)} = \frac{1}{N+\rho}\left(\mathbf{Y} + \rho\overline{\mathbf{A}\mathbf{X}}^{(k+1)} + \rho\mathbf{C}^{(k)}\right)$.
5: Update $\mathbf{C}^{(k+1)} = \mathbf{C}^{(k)} + \overline{\mathbf{A}\mathbf{X}}^{(k+1)} - \overline{\mathbf{B}}^{(k+1)}$.
6: Set $k = k + 1$.
7: **until** $k = k_{max}$ or $\mathbf{X}^{(k)}$ satisfies some stopping criterion.

GROUP LASSO-based Decoder

- The approximation part of the GROUP LASSO-based decoder approximates the estimation obtained by Algorithm 1
  - The operations for complex numbers in Algorithm 1 are readily implemented with operations for real numbers using a standard neural network
  - Each building block of the approximation part of the GROUP LASSO-based decoder realizes one iteration of Algorithm 1
  - $\lambda > 0$ and $\rho > 0$ are tunable parameters

AMP-based Decoder

- The AMP algorithm with the MMSE denoiser in [Liu & Yu (2018)] has computational complexity $\mathcal{O}(LNM)$, and achieves excellent recovery accuracy and short computation time for large N, M and L
  - Assume that $\alpha(n)$, $n \in \mathcal{N}$ are i.i.d. Bernoulli random variables, each with probability $\epsilon \in (0, 1)$ being 1, ϵ is known, and non-zero elements of $\mathbf{X}$ are i.i.d. according to $\mathcal{CN}(0, 1)$
- We slightly generalize the AMP algorithm in [Liu & Yu (2018)] by replacing ϵ in the update for the estimate of $\mathbf{X}_{n,:}$ with $\epsilon(n)$
  - Assume that $\alpha(n)$, $n \in \mathcal{N}$ are independent Bernoulli random variables and the probability of $\alpha(n)$ being 1 is $\epsilon(n) \in (0, 1)$
  - The parallel operations make use of the parallelizable neural network architecture
  - Parameters $\epsilon(n)$, $n \in \mathcal{N}$ influence the recovery accuracy

AMP-based Decoder

Algorithm 2 Generalization of AMP [Liu & Yu (2018)]

1: Set $\mathbf{X}^{(0)} = \mathbf{0}_{M\times N}$, $\mathbf{R}^{(0)} = \mathbf{Y}$ and $k = 0$.

2: **repeat**

3: For all $n \in \mathcal{N}$, update $\mathbf{X}_{n,:}^{(k+1)} = \left(\frac{\frac{1}{1+(\tau^{(k)})^2}\left((\mathbf{R}^{(k)})^H\mathbf{A}_{:,n}+(\mathbf{X}_{n,:}^{(k)})^H\right)}{1+\frac{1-\epsilon(n)}{\epsilon(n)}\left(\frac{(\tau^{(k)})^2+1}{(\tau^{(k)})^2}\right)^M \exp\left((\eta(n))^k\right)} \right)^H$, where $\tau^{(k)} = \sqrt{\frac{1}{ML}}|||\mathbf{R}^{(k)}|||_F$ and $(\eta(n))^k = \frac{-\|(\mathbf{R}^{(k)})^H\mathbf{A}_{:,n}+(\mathbf{X}_{n,:}^{(k)})^H\|_2^2}{(\tau^{(k)})^2((\tau^{(k)})^2+1)}$.

4: Update $\mathbf{R}^{(k+1)} = \mathbf{Y} - \mathbf{A}\mathbf{X}^{(k+1)} + \frac{N}{L}\mathbf{R}^{(k)} \sum_{n\in\mathcal{N}} \left(\frac{\frac{1}{1+(\tau^{(k)})^2}\mathbf{I}_{M\times M}}{1+\frac{1-\epsilon(n)}{\epsilon(n)}\left(\frac{(\tau^{(k)})^2+1}{(\tau^{(k)})^2}\right)^M \exp\left((\eta(n))^k\right)} + \frac{(t(n))^{(k)}(\eta(n))^k}{\left((1+(\tau^{(k)})^2)(\tau^{(k)})^2\right)(1+(t(n))^{(k)})^2} \right)$, where $(t(n))^{(k)} = \frac{1-\epsilon(n)}{\epsilon(n)}\left(\frac{(\tau^{(k)})^2+1}{(\tau^{(k)})^2}\right)^M \exp\left(\frac{-\|(\mathbf{R}^{(k)})^H\mathbf{A}_{:,n}+(\mathbf{X}_{n,:}^{(k)})^H\|_2^2}{(\tau^{(k)})^2((\tau^{(k)})^2+1)} \right)$.

5: Set $k = k + 1$.

7: **until** $k = k_{max}$ or $\mathbf{X}^{(k)}$ satisfies some stopping criterion.

AMP-based Decoder

- The approximation part of the AMP-based decoder approximates the estimation obtained by Algorithm 2
  - The operations for complex numbers in Algorithm 2 are readily implemented with operations for real numbers using a standard neural network
  - Each building block of the approximation part of the AMP-based decoder realizes one iteration of Algorithm 2
  - $\epsilon(n) \in (0, 1)$, $n \in \mathcal{N}$ are tunable parameters

Training Process for jointly sparse signal recovery

- Consider I training samples $(\mathbf{X}^{[i]}), i = 1, \cdots, I$
- $\hat{\mathbf{X}}^{[i]}$ denotes the output of the auto-encoder corresponding to input $\mathbf{X}^{[i]}$
- Adopt the mean squared error (MSE) loss function to measure the difference between $\hat{\mathbf{X}}^{[i]}$ and $\mathbf{X}^{[i]}$:

$$\text{Loss}\left((\mathbf{X})_{i=1,\cdots,I}^{[i]}, (\hat{\mathbf{X}}^{[i]})_{i=1,\cdots,I}\right) = \frac{1}{NI}\sum_{i=1}^{I} |||\mathbf{X}^{[i]} - \hat{\mathbf{X}}^{[i]}|||_F^2$$

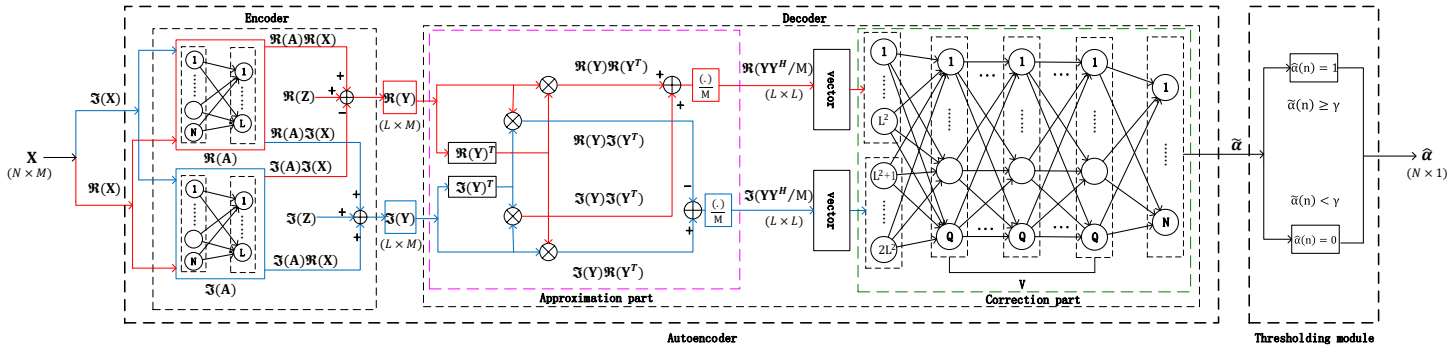
- Train the auto-encoder using the adaptive moment estimation (ADAM) algorithm [Kingma et al. (2015)]
  - ADAM is an algorithm for first-order gradient-based optimization and is well suited for problems that are large in terms of data and/or parameters

Outline

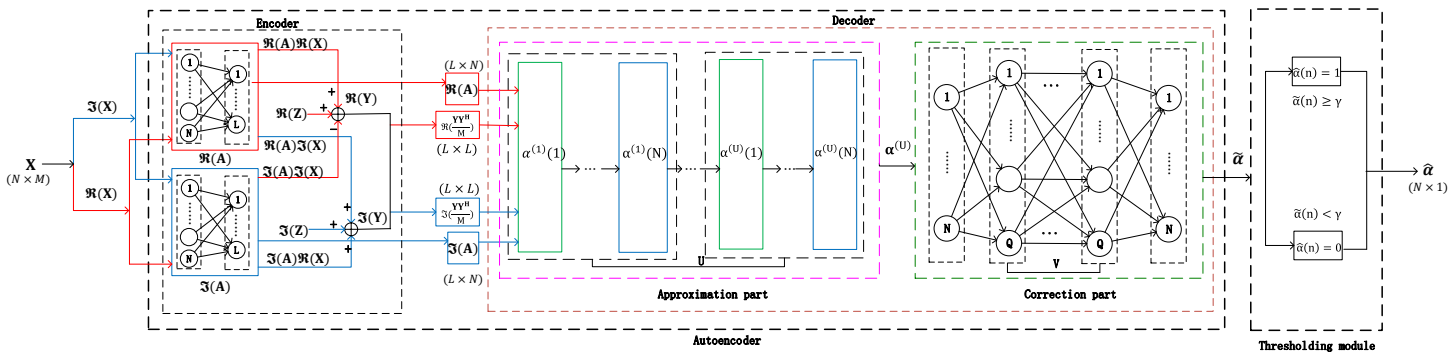
Proposed approach for jointly sparse support recovery

- Jointly design the common measurement matrix and jointly sparse support recovery method for jointly sparse complex signals
- Utilize the standard auto-encoder structure for real numbers in deep learning
- Consist of an encoder, a model-driven decoder and a thresholding module
  - The encoder mimics noisy linear measurement process, same as for jointly sparse signal recovery
  - The model-driven decoder mimics jointly sparse support recovery process via an approximation part and a correction part
  - The thresholding module generates binary approximations to obtain the common support
- After training, obtain the common measurement matrix from the encoder, and directly use the decoder and thresholding module for jointly sparse support recovery

Two instances



(a) Proposed approach with covariance-based decoder ($U = 0$)



(b) Proposed approach with MAP-based decoder

Figure: Proposed model-driven approach for jointly sparse support recovery

Other instances

- Any model-driven decoder for jointly sparse signal recovery, e.g., the GROUP LASSO-based decoder and the AMP-based decoder, can also be used for jointly sparse support recovery when followed by the thresholding module
  - The training process for the encoder and the model-driven decoder remains the same as for jointly sparse signal recovery
  - The only difference is that a thresholding module is used for producing the common support

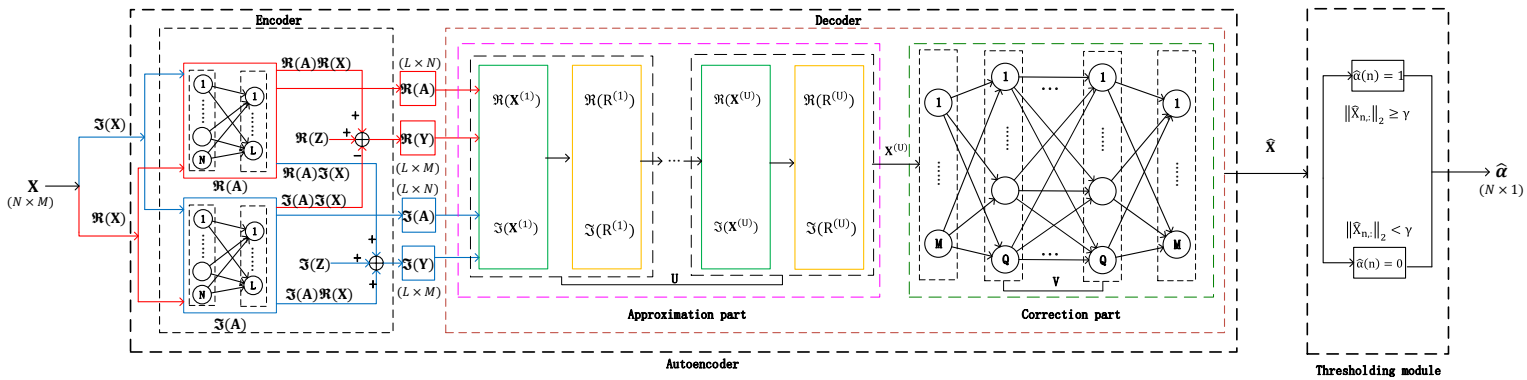


Figure: Proposed approach with AMP-based decoder. The auto-encoder is the same as that in the jointly sparse signal recovery part.

Model-driven decoder for jointly sparse support recovery

- Mimic the jointly sparse support recovery process of a particular method
- Consist of an approximation part and a correction part
  - The approximation part uses U (≥ 0) building blocks to implement U iterations of the iterative algorithm
    - Input $\mathbf{Y}$ (and $\mathbf{A}$), and output $\boldsymbol{\alpha}^{(U)}$
    - The different between $\boldsymbol{\alpha}^{(U)}$ and the actual $\boldsymbol{\alpha}$ generally decreases with U
  - The correction part uses V (≥ 1) fully connected layers to reduce the difference between $\boldsymbol{\alpha}^{(U)}$ and $\boldsymbol{\alpha}$
    - Input $\boldsymbol{\alpha}^{(U)}$, and output $\tilde{\boldsymbol{\alpha}}$
    - In each of the first $V-1$ correction layers, choose rectified linear unit (ReLU) as the activation function
    - In the last correction layer, choose sigmoid function as the activation function
    - V influences the training error and generalization error
  - U and V are jointly chosen to achieve higher recovery accuracy and shorter computation time than the underlying method
    - Degrade to a particular method (sufficiently large U and $V = 0$) or a purely data-driven method ($U = 0$ and $V > 0$)

Covariance-based decoder

- Properties of common support [Pal & Vaidyanathan (2015)]:

$$\begin{aligned}\mathrm{vec}(\mathbf{Y}\mathbf{Y}^H/M) =& \mathbf{A}^* \odot \mathbf{A}\mathbf{r} + \mathrm{vec}(\mathbf{E_1}) + \mathrm{vec}(\mathbf{E_2}) \\ \mathrm{supp}(\mathbf{r}) =& \mathrm{supp}(\mathbf{x}_m), m \in \mathcal{M}\end{aligned}$$

  where $r(n) = \|\mathbf{X}_{n,:}\|_2^2/M$, and $\odot$ represents the Khatri-Rao product between two matrices
  - Suppose the non-zero elements of $\mathbf{X}$ are i.i.d. with zero mean
  - As $M \to \infty$, $\mathbf{Y}\mathbf{Y}^H/M \to \mathrm{Cov}(\mathbf{y}_m)$, $\mathbf{E}_1 \to \mathbf{0}_{L\times L}$, $\mathbf{E}_2 \to \sigma^2\mathbf{I}_{L\times L}$, and $\mathrm{vec}(\mathbf{Y}\mathbf{Y}^H/M)$ provides of $\mathbf{r}$
- The covariance-based estimation method for jointly sparse support recovery in [Pal & Vaidyanathan (2015)] estimates the common support of $\mathbf{x}_m, m \in \mathcal{M}$ by estimating the support of $\mathbf{r}$ using LASSO for SMV models, in the case of very large M

Covariance-based decoder

- Motivated by [Pal & Vaidyanathan (2015)], the covariance-based decoder estimates $\mathrm{supp}(\mathbf{r})$ from noisy linear measurements $\mathrm{vec}(\mathbf{Y}\mathbf{Y}^H/M)$ of $\mathbf{r}$, obtained through a measurement matrix $\mathbf{A}^* \odot \mathbf{A}$
  - Work for small M
  - Recovery accuracy increases with M
- The approximation part of the covariance-based decoder provides $\mathbf{Y}\mathbf{Y}^H/M$ $(U = 0)$
- The correction part of the covariance-based decoder estimates $\mathrm{vec}(\mathbf{Y}\mathbf{Y}^H/M)$ from $\mathbf{Y}\mathbf{Y}^H/M$

MAP-based decoder

- Assume that $\alpha(n)$, $n \in \mathcal{N}$ are independent Bernoulli random variables, the probability of $\alpha(n)$ being 1 is $\epsilon(n) \in (0,1)$, $\epsilon(n)$, $n \in \mathcal{N}$ are known, and non-zero elements of $\mathbf{X}$ are i.i.d. according to $\mathcal{CN}(0,1)$
- The negative log function of the conditional density of $\mathbf{Y}$ given $\boldsymbol{\alpha}$ (omitting an additive constant)is [Jiang & Cui (2020)]:

$$f_{MAP}(\boldsymbol{\alpha}) = \log|\mathbf{A}\boldsymbol{\Gamma}\mathbf{A} + \sigma^2\mathbf{I}_{L\times L}| + \mathrm{tr}((\mathbf{A}\boldsymbol{\Gamma}\mathbf{A} + \sigma^2\mathbf{I}_{L\times L})^{-1}\hat{\boldsymbol{\Sigma}}) - \frac{1}{M}\sum_{n\in\mathcal{N}}(\alpha(n)\log\epsilon(n) + (1-\alpha(n))\log(1-\epsilon(n)))$$

  where $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\alpha})$ and $\hat{\boldsymbol{\Sigma}} = \mathbf{Y}\mathbf{Y}^H/M$
- The MAP estimation problem for jointly sparse support recovery in MMV models is:

$$\min_{\boldsymbol{\alpha}\succeq 0} f_{MAP}(\boldsymbol{\alpha})$$

MAP-based decoder

- The coordinate descent algorithm for the MAP estimation in [Jiang & Cui (2020)] can obtain a locally optimal solution and has computational complexity $\mathcal{O}(NL^2)$
- We slightly simplify the MAP algorithm in [Jiang & Cui (2020)] by ignoring the interference
- The approximation part of the MAP-based decoder approximates the estimation obtained by the MAP algorithm
  - The operations for complex numbers in the MAP algorithm are readily implemented with operations for real numbers using a standard neural network
  - Each building block of the approximation part of the MAP-based decoder realizes one iteration of the MAP algorithm
  - $\epsilon(n) \in (0, 1)$, $n \in \mathcal{N}$ are tunable parameters

Training Process for jointly sparse support recovery

- Consider I training samples $(\mathbf{X}^{[i]}, \boldsymbol{\alpha}^{[i]}), i = 1, \cdots, I$
- $\tilde{\boldsymbol{\alpha}}^{[i]}$ denotes the output of the auto-encoder corresponding to input $\mathbf{X}^{[i]}$
- Adopt the binary cross-entropy loss function to measure the difference between $\boldsymbol{\alpha}^{[i]}$ and $\tilde{\boldsymbol{\alpha}}^{[i]}$:

$$\text{Loss}\left((\boldsymbol{\alpha}^{[i]})_{i=1,\cdots,I}, (\tilde{\boldsymbol{\alpha}}^{[i]})_{i=1,\cdots,I}\right) =$$

$$-\frac{1}{NI}\sum_{i=1}^{I}\sum_{n\in\mathcal{N}}\left((\alpha(n))^{[i]}\log((\tilde{\alpha}(n))^{[i]}) + (1-(\alpha(n))^{[i]})\log(1-(\tilde{\alpha}(n))^{[i]})\right)$$

- Train the auto-encoder using the ADAM algorithm [Kingma et al. (2015)]

Thresholding module for jointly sparse support recovery

- The hard thresholding module with threshold r converts the output of the auto-encoder $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^N$ to the final output of the proposed approach $\hat{\boldsymbol{\alpha}} \in \{0,1\}^N$ [Li et al. (2019)]:

$$\hat{\alpha}(n) = \mathbb{I}[\tilde{\alpha}(n) \geq \gamma], \quad n \in \mathcal{N}$$

- For training samples $(\mathbf{X}^{[i]}, \boldsymbol{\alpha}^{[i]}), i = 1, \cdots, I$, choose the optimal threshold

$$\gamma^* = \arg\min_{\gamma} \frac{1}{NI} \sum_{i=1}^{I} \|\boldsymbol{\alpha}^{[i]} - \hat{\boldsymbol{\alpha}}^{[i]}\|_1$$

as the threshold for the hard thresholding module

Outline

Simulation setup

- Conduct numerical experiments on device activity detection and channel estimation in MIMO-based grant free random access
- Consider two choices for N, i.e., $N = 100$ and $N = 1000$
- Generate the jointly sparse signal $\mathbf{X} \in \mathbb{C}^{N \times M}$ with $x_m(n) = \alpha(n)h_m(n), m \in \mathcal{M}, n \in \mathcal{N}$ according to $h_m(n) \sim \mathcal{CN}(0, 1), m \in \mathcal{M}, n \in \mathcal{N}$
- Generate the additive white Gaussian noise $\mathbf{Z} \in \mathbb{C}^{L \times M}$ according to $\mathbf{Z}_{:,m} \sim \mathcal{CN}(\mathbf{0}_{L\times 1}, \sigma^2\mathbf{I}_{L\times L}), m \in \mathcal{M}$ where $\sigma^2 = 0.1$

Simulation setup

- To demonstrate the ability for effectively utilizing sparse patterns, consider three types of distributions for $\boldsymbol{\alpha}$
  - The independent case for channel estimation and device activity detection:
    - N devices are divided into two groups $\mathcal{N}_1$ and $\mathcal{N}_2$ of same size
    - The devices in $\mathcal{N}_i$ accessing the channel in an i.i.d. manner with access probability $\Pr[\alpha(n) = 1] = p_i$, $n \in \mathcal{N}_i$
    - $p = \frac{p_1+p_2}{2}$ denotes the average access probability
  - The correlated case with a single active group for channel estimation:
    - N devices are divided into G groups of the same size N/G
    - The active states of devices in each group are the same
    - Only one group is selected to be active uniformly at random
    - $1/G$ can be viewed as the access probability p
  - The correlated case with i.i.d group activity for device activity detection:
    - N devices are divided into G groups of the same size N/G
    - The active states of the devices in each group are the same
    - G groups activate in an i.i.d. manner with the access probability p

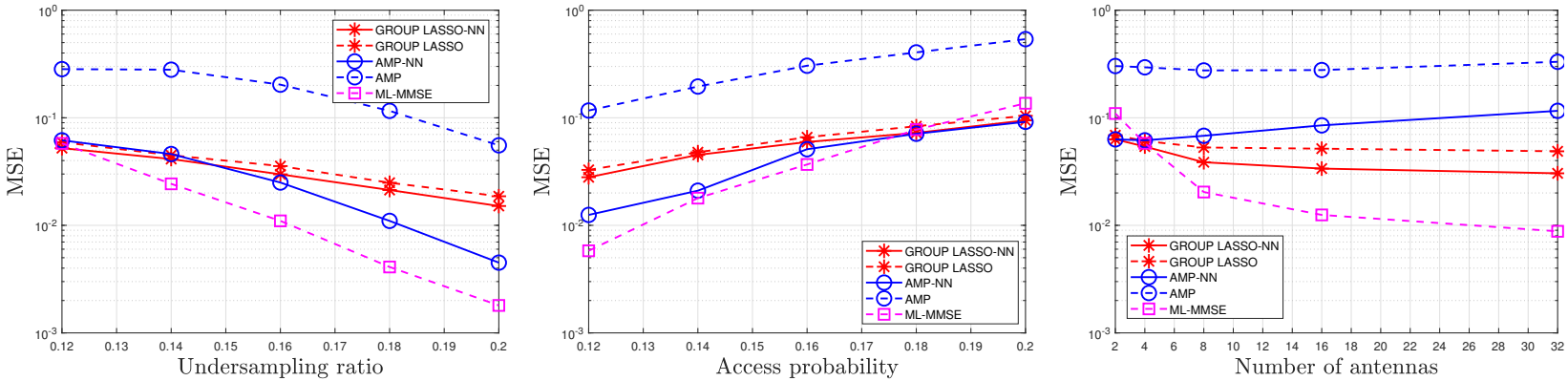
Simulation setup

- All the baseline schemes adopt the same set of pilot sequences for the N devices whose entries are i.i.d. $\mathcal{CN}(0,1)$
- In training the architectures of the proposed approaches, set $\|\mathbf{a}_n\|_2 = \sqrt{L}$
- The sizes of training samples and validation samples for training the architectures of the proposed approaches and the size of testing samples for evaluating the proposed approaches and the baseline schemes are selected as 9×10^3, 1×10^3 and 1×10^3, respectively
- The maximization epochs, learning rate and batch size in training the proposed architectures are set as 1×10^5, 0.0001 and 32, respectively

Channel estimation

- Evaluate the proposed model-driven approach with the GROUP LASSO-based decoder (at $N = 100$) and the AMP-based decoder (at $N = 100$ and $N = 1000$)
  - GROUP LASSO-NN: $U = 200$ and $V = 3$
  - AMP-NN: $U = 50$ and $V = 3$
- Consider three baseline schemes:
  - GROUP LASSO conducts channel estimation using the block coordinate descent algorithm of computational complexity $\mathcal{O}(LNM)$ [Qin et al. (2013)]: 200 iterations
  - AMP conducts channel estimation using the AMP algorithm with MMSE denoiser based on the known p of computational complexity $\mathcal{O}(LNM)$ [Liu & Yu (2018)]: 50 iterations
  - ML-MMSE uses the ML algorithm for device activity detection of computational complexity $\mathcal{O}(NL^2)$ [Haghighatshoar et al. (2018)], and then uses MMSE for channel estimation: 55 iterations
- Evaluate the MSE $\frac{1}{NT}\sum_{t=1}^{T}\|\|\mathbf{X}^{[t]} - \hat{\mathbf{X}}^{[t]}\|\|_F^2$ and computation time (on the same server) of each scheme, where $T = 1000$

Independent case at $N = 100$



(a) MSE versus L/N at $p = 0.1$, $M = 4$, $p_1/p_2 = 3$.

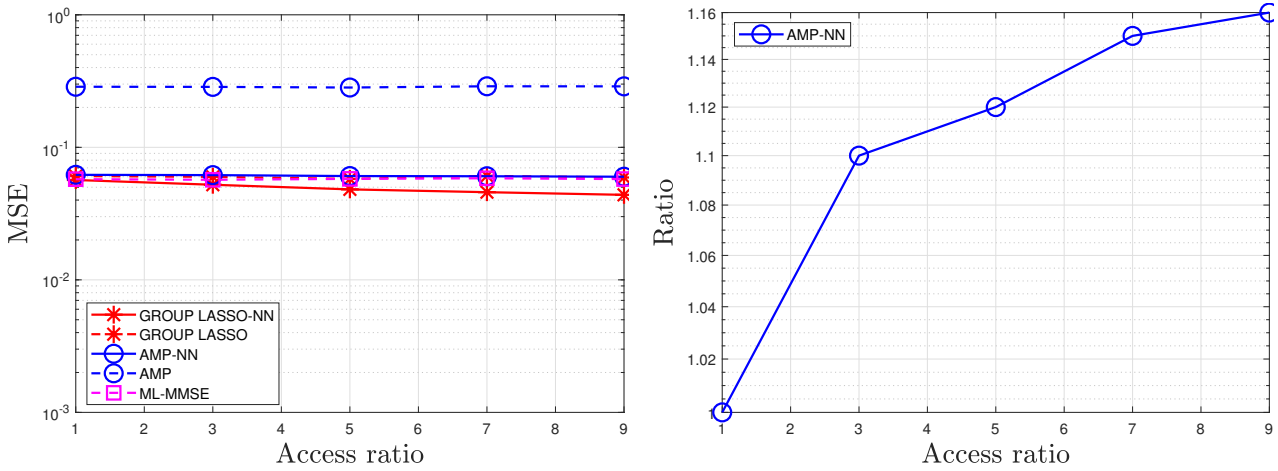
(b) MSE versus p at $L/N = 0.2$, $M = 4$, $p_1/p_2 = 3$.

(c) MSE versus M at $L/N = 0.12$, $p = 0.1$, $p_1/p_2 = 3$.

Figure: MSE in the independent case at $N = 100$.

- MSE of each scheme decreases with L/N
- MSE of each scheme increases with p
- MSEs of schemes besides AMP, AMP-NN decrease with M
- GROUP LASSO-NN and AMP-NN always outperform GROUP LASSO and AMP (due to effective **A** and correction layers)
- GROUP LASSO-NN outperforms AMP-NN at small N, L/N
- ML-MMSE has the smallest MSE (at long computation time)

Independent case at $N = 100$



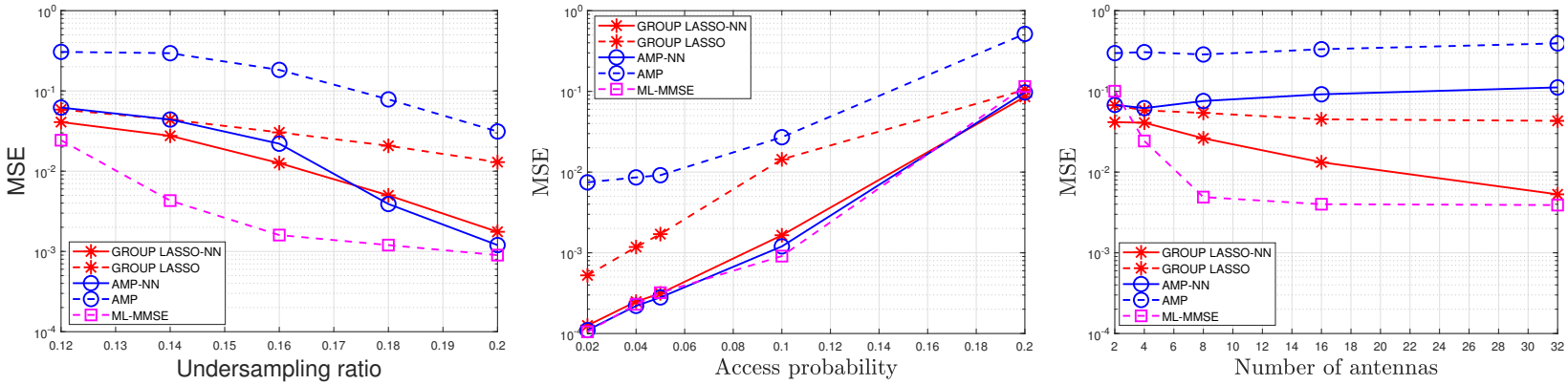
(a) MSE versus p_1/p_2 at $L/N = 0.12$, $M = 4$, $p = 0.1$.

(b) ϵ_1/ϵ_2 versus p_1/p_2 at $L/N = 0.12$, $M = 4$, $p = 0.1$.

Figure: MSE and ϵ_1/ϵ_2 in the independent case at $N = 100$. $\epsilon_1 = \frac{2}{N}\sum_{n\in\mathcal{N}_1}\epsilon(n)$ and $\epsilon_2 = \frac{2}{N}\sum_{n\in\mathcal{N}_2}\epsilon(n)$, where $\epsilon(n), n \in \mathcal{N}$ are extracted from the trained AMP-based decoder.

- MSEs of GROUP LASSO-NN and AMP-NN decrease with p_1/p_2 (due to exploitation of the difference of p_1 and p_2)
- MSEs of others almost do not change p_1/p_2
- ϵ_1/ϵ_2 increases p_1/p_2

Correlated case with a single active group at $N = 100$



(a) MSE versus L/N at $p = 0.1$, $M = 4$.

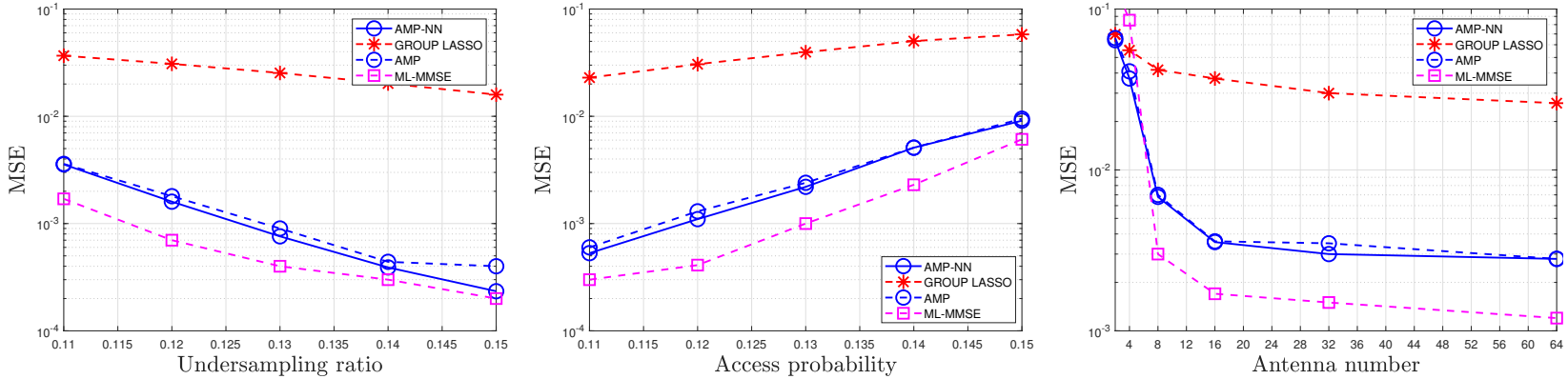
(b) MSE versus p at $L/N = 0.2$, $M = 4$.

(c) MSE versus M at $L/N = 0.12$, $p = 0.1$.

Figure: MSE in the correlated case with a single active group at $N = 100$.

- The gains of GROUP LASSO-NN and AMP-NN over GROUP LASSO and AMP are larger in the correlated case than in the independent case (due to exploitation of more features of sparsity pattens)

Independent case at $N = 1000$



(a) MSE versus L/N at $p = 0.1$, $M = 16$, $p_1/p_2 = 3$.

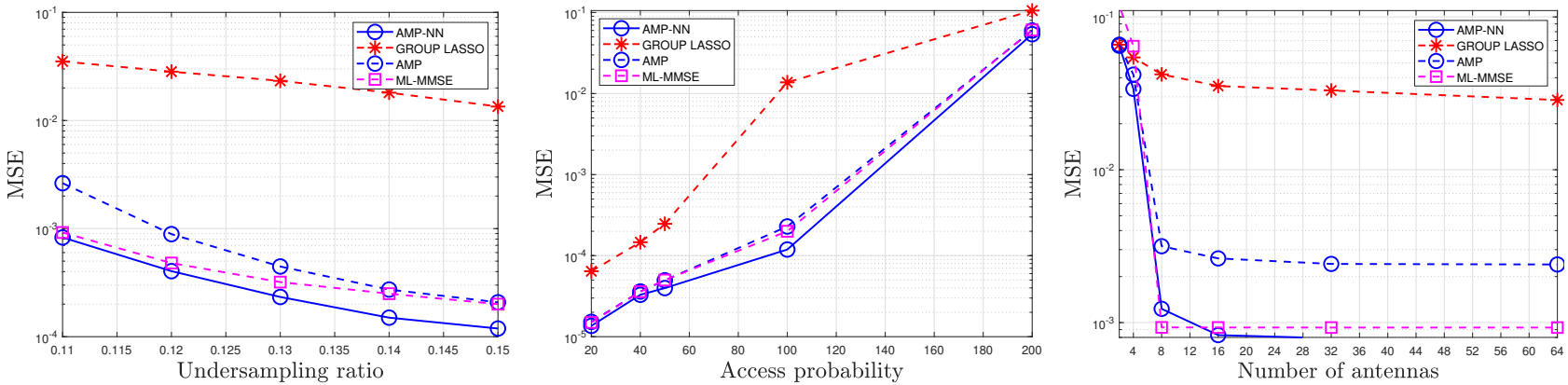
(b) MSE versus p at $L/N = 0.15$, $M = 16$, $p_1/p_2 = 3$.

(c) MSE versus M at $L/N = 0.11$, $p = 0.1$, $p_1/p_2 = 3$.

Figure: MSE in the independent case at $N = 1000$.

- MSE of each scheme always decreases with L/N
- MSE of each scheme increases with p
- MSE of each scheme decreases with M
- AMP-NN and AMP are close (as not many features of sparsity patterns can be utilized by AMP-NN and AMP already achieves very excellent recovery accuracy at large N)
- ML-MMSE has the smallest MSE (at long computation time)

Correlated case with a single active group at $N = 1000$



(a) MSE versus L/N at $p = 0.1$, $M = 16$.

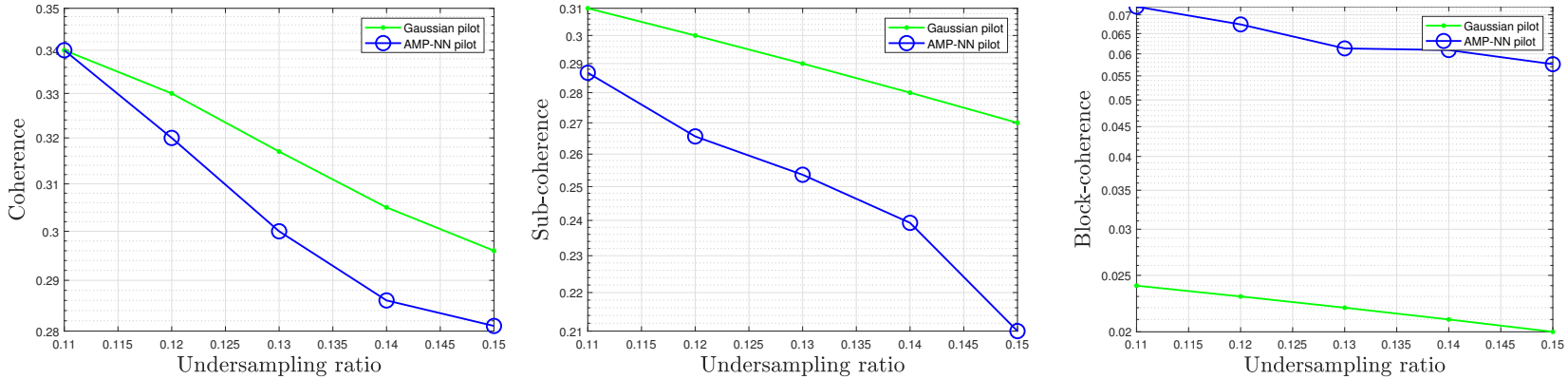
(b) MSE versus p at $L/N = 0.15$, $M = 16$.

(c) MSE versus M at $L/N = 0.11$, $p = 0.1$.

Figure: MSE in the correlated case with a single active group at $N = 1000$.

- The gain of AMP-NN over AMP is larger in the correlated case than in the independent case (due to exploitation of more features of sparsity pattens)
- AMP-NN achieves the smallest MSE (as the underlying AMP functions well at large N and the encoder and correction layers of AMP-NN effectively utilize the sparsity patterns)

Coherence, sub-coherence, block-coherence at $N = 1000$



(a) Coherence versus L/N in the independent case at $p_1/p_2 = 3$.

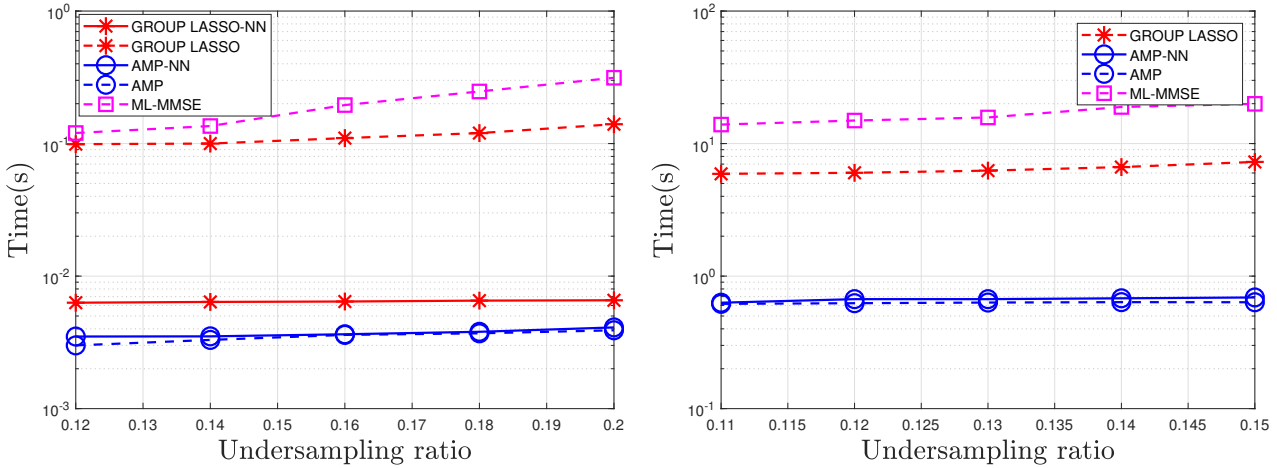
(b) Sub-coherence within one group versus L/N in the correlated case.

(c) Block-coherence across groups versus L/N in the correlated case.

Figure: Coherence, sub-coherence and block-coherence at $N = 1000$, $p = 0.1$, $M = 16$.

- The learned measurement matrix has smaller coherence and sub-coherence (to more effectively differentiate devices always active simultaneously)
  - Coherence and subcoherence reflect the orthogonality of all the columns or columns within one group
- The learned measurement matrix has larger block-coherence (to sacrifice the differentiability of devices never active simultaneously)
  - Block-coherence reflects the overall orthogonality between two groups of columns

Computation time at $N = 100$ and $N = 1000$



(a) Computation time versus L/N at $M = 4$, $N = 100$.

(b) Computation time versus L/N at $M = 16$, $N = 1000$.

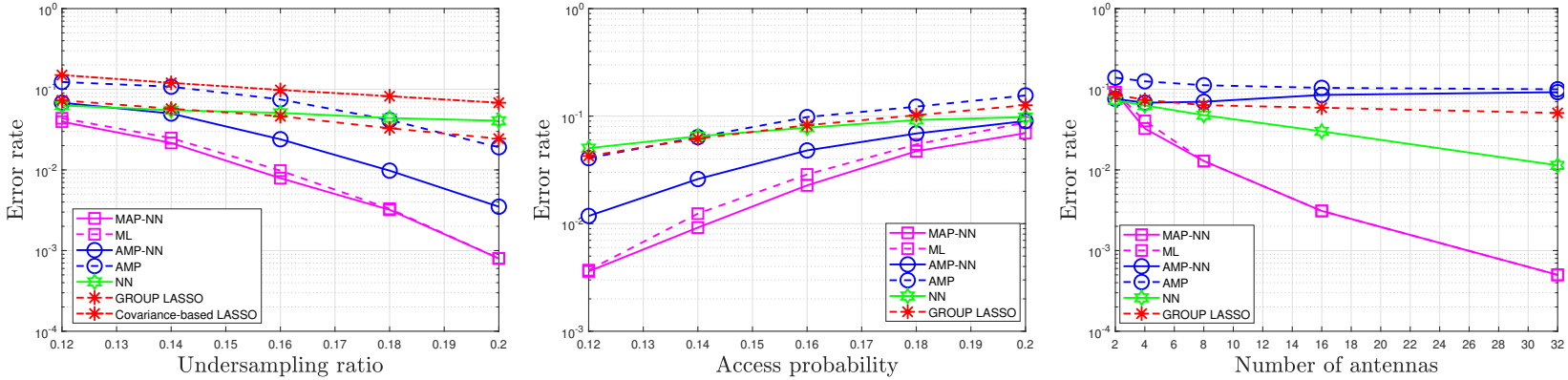
Figure: Computation time in the independent case at $p = 0.1$, $p_1/p_2 = 3$.

- GROUP LASSO-NN has shorter computation time than GROUP LASSO (due to parallel computation)
- GROUP LASSO and ML-MMSE have much longer computation time than others, especially at large N, and may not be applicable for practical mMTC
- AMP-NN and AMP have the shortest computation time ($V = 3$)

Device activity detection

- Evaluate the proposed model-driven approach with the covariance-based decoder and the MAP-based decoder (both at $N = 100$) and with the AMP-based decoder (at $N = 100$ and $N = 1000$)
  - NN: $U = 0$ and $V = 3$
  - MAP-NN: $U = 55$ and $V = 3$
  - AMP-NN: $U = 50$ and $V = 3$
- Consider four baseline schemes:
  - AMP (based on the known p) of computational complexity $\mathcal{O}(LNM)$ [Liu & Yu (2018)]: 50 iterations
  - ML of computational complexity $\mathcal{O}(NL^2)$ [Haghighatshoar et al. (2018)]: 55 iterations
  - GROUP LASSO of computational complexity $\mathcal{O}(LNM)$ [Qin et al. (2013)]: 200 iterations
  - covariance-based LASSO of computational complexity $\mathcal{O}(LNM)$ [Pal & Vaidyanathan (2015)]: 200 iterations
- Evaluate the error rate $\frac{1}{NT}\sum_{t=1}^{T}\|\boldsymbol{\alpha}^{[t]} - \hat{\boldsymbol{\alpha}}^{[t]}\|_1$ and computation time (on the same server), where $T = 1000$

Independent case at $N = 100$



(a) Error rate versus L/N at $p = 0.1$, $M = 4$, $p_1/p_2 = 3$.

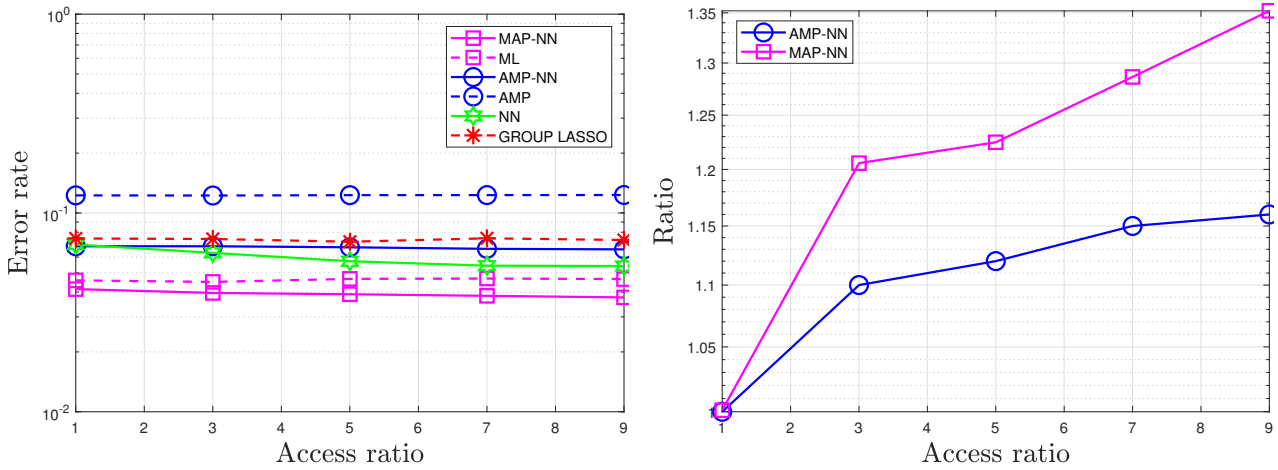
(b) Error rate versus p at $L/N = 0.2$, $M = 4$, $p_1/p_2 = 3$.

(c) Error rate versus M at $L/N = 0.12$, $p = 0.1$, $p_1/p_2 = 3$.

Figure: Error rate in the independent case at $N = 100$.

- covariance-based LASSO performs much worse than Group LASSO and AMP at small M (no longer compare with it later)
- Error rates of all schemes show similar trends with respect to L/N, p and M to MSEs
- MAP-NN and AMP-NN always outperform ML and AMP, respectively (due to effective **A** and correction layers)
- MAP-NN outperforms NN (as the underlying MAP is designed for independence and not many features of sparse patterns exist)

Independent case at $N = 100$



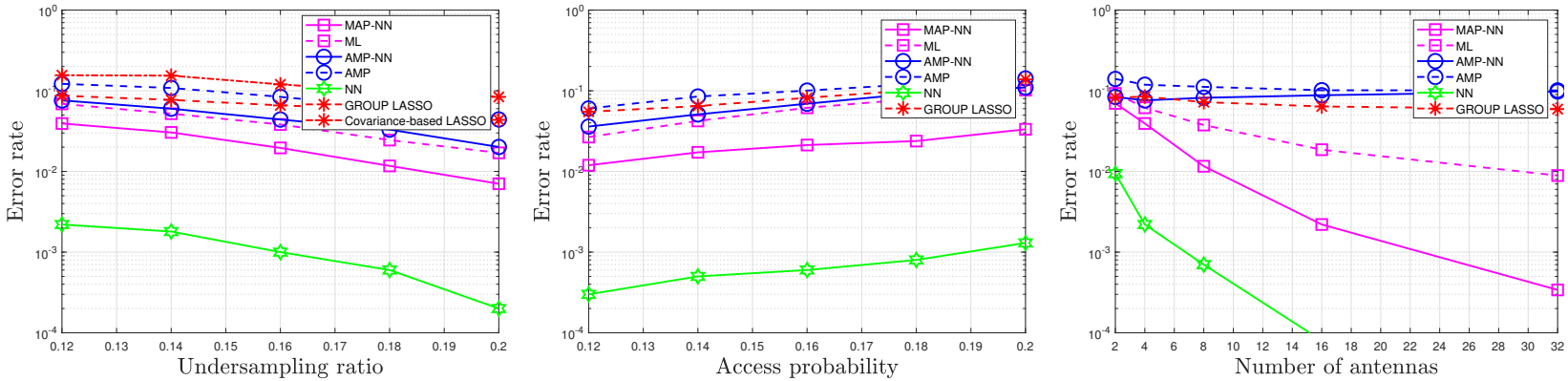
(a) Error rate versus p_1/p_2 at $L/N = 0.12$, $M = 4$, $p = 0.1$.

(b) ϵ_1/ϵ_2 versus p_1/p_2 at $L/N = 0.12$, $M = 4$, $p = 0.1$.

Figure: Error rate and ϵ_1/ϵ_2 in the independent case at $N = 100$. $\epsilon_1 = \frac{2}{N}\sum_{n\in\mathcal{N}_1}\epsilon(n)$ and $\epsilon_2 = \frac{2}{N}\sum_{n\in\mathcal{N}_2}\epsilon(n)$, where $\epsilon(n), n \in \mathcal{N}$ are extracted from the trained AMP-based decoder.

- The error rates of NN, MAP-NN and AMP-NN decrease with p_1/p_2
- The error rates of others almost do not change p_1/p_2
- ϵ_1/ϵ_2 increases p_1/p_2

Correlated case with i.i.d. group activity at $N = 100$



(a) Error rate versus L/N at $p = 0.1$, $M = 4$, $G = 20$.

(b) Error rate versus p at $L/N = 0.2$, $M = 4$, $G = 20$.

(c) Error rate versus M at $L/N = 0.12$, $p = 0.1$, $G = 20$.

Figure: Error rate in the correlated case with i.i.d. group activity at $N = 100$.

- NN outperforms MAP-NN (as NN can better utilize the correlation information and the underlying MAP is designed for the independent case)

Correlated case with i.i.d. group activity at $N = 100$

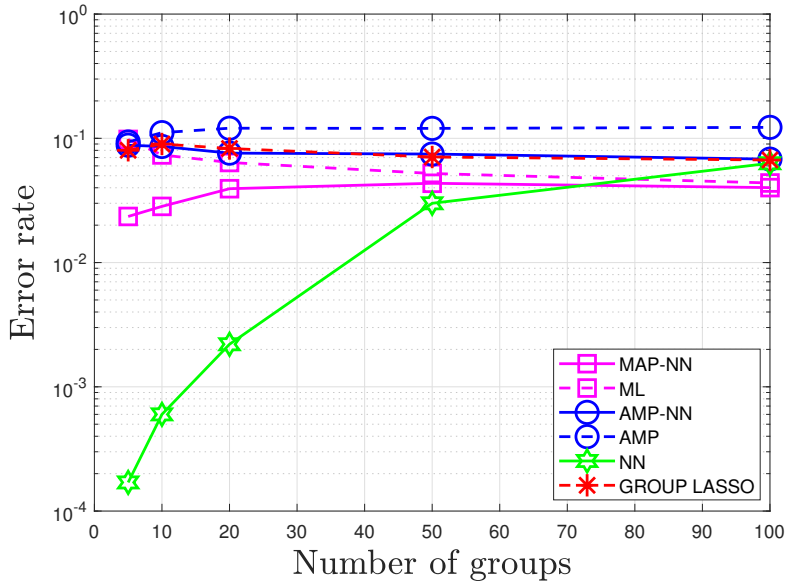
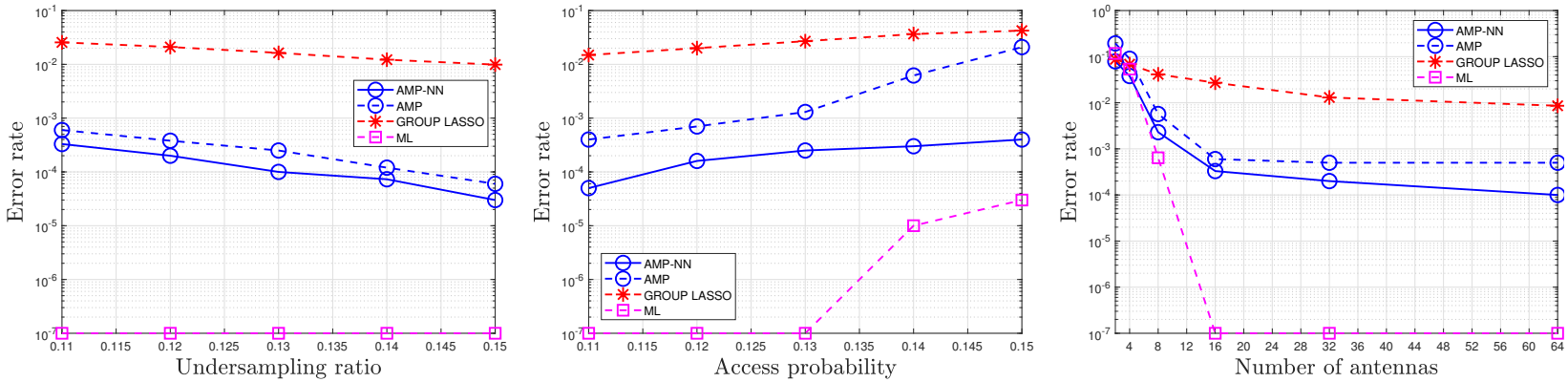


Figure: Error rate versus G in the correlated case with i.i.d. group activity at $L/N = 0.12$, $M = 4$, $p = 0.1$, $N = 100$.

- The error rates of NN and MAP-NN increase with G (as correlation that can be utilized decreases with G)
- The error rate of GROUP LASSO almost does not change with G
- The error rate of ML always decreases with G

Independent case at $N = 1000$



(a) Error rate versus L/N at $p = 0.1$, $M = 16$, $p_1/p_2 = 3$.

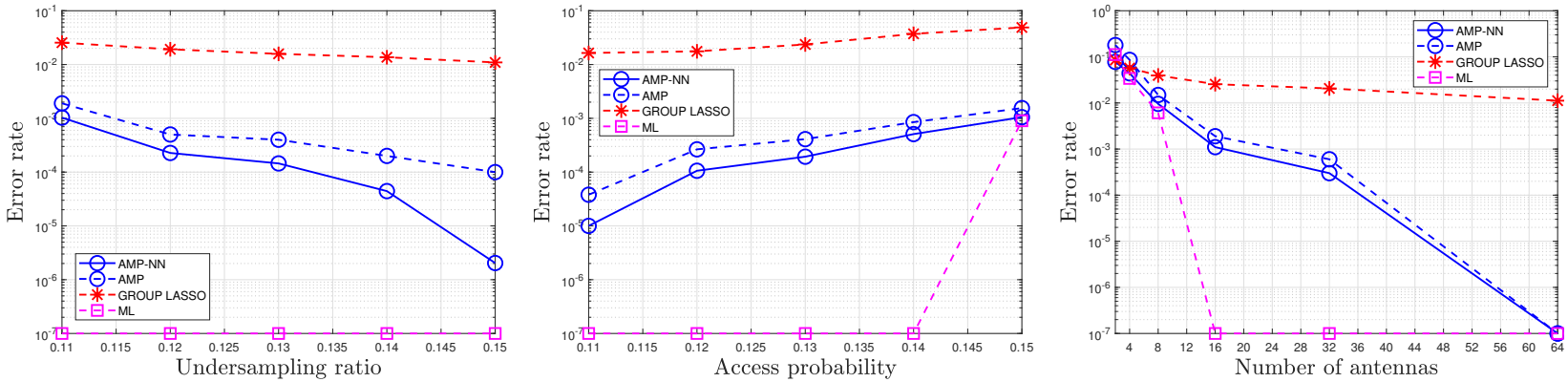
(b) Error rate versus p at $L/N = 0.15$, $M = 16$, $p_1/p_2 = 3$.

(c) Error rate versus M at $L/N = 0.11$, $p = 0.1$, $p_1/p_2 = 3$.

Figure: Error rate in the independent case at $N = 1000$.

- Error rates of all schemes show similar trends with respect to L/N, p and M to MSEs
- AMP-NN always outperforms AMP (due to effective **A** and correction layers)
- ML achieves the smallest error rate in the independent case (at the cost of computation time increase)

Correlated case with i.i.d. group activity at $N = 1000$



(a) Error rate versus L/N at $p = 0.1$, $M = 16$, $G = 200$.

(b) Error rate versus p at $L/N = 0.15$, $M = 16$, $G = 200$.

(c) Error rate versus M at $L/N = 0.11$, $p = 0.1$, $G = 200$.

Figure: Error rate in the correlated case with i.i.d. group activity at $N = 1000$.

- The gain of AMP-NN versus AMP in the correlated case is larger than in the independent case

Correlated case with i.i.d. group activity at $N = 1000$

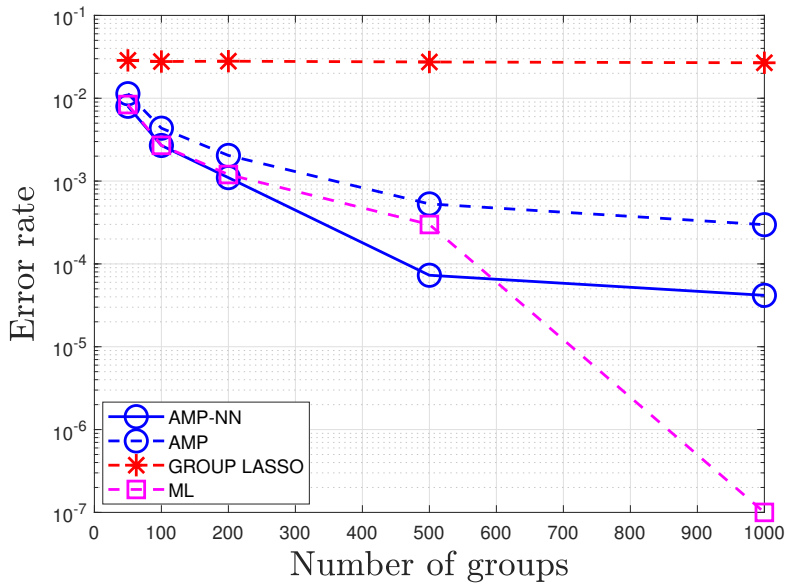
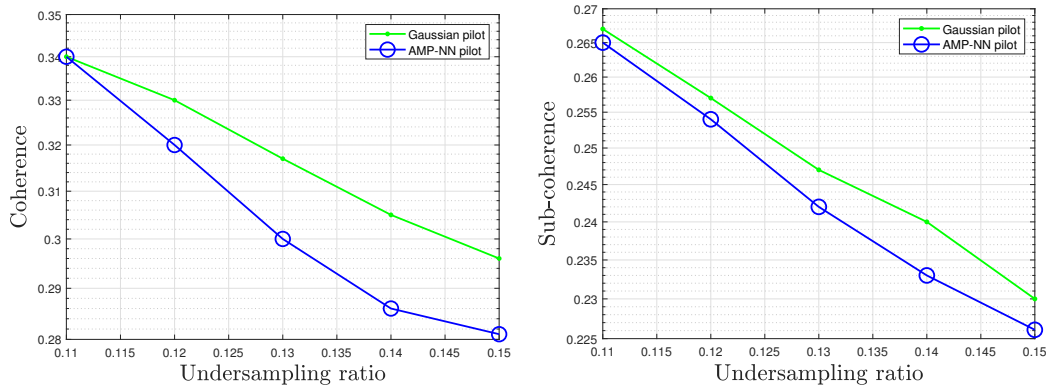


Figure: Error rate versus G in the correlated case with i.i.d. group activity at $L/N = 0.11$, $M = 16$, $p = 0.1$, $N = 1000$.

- AMP-NN achieves the smallest error in the correlated with i.i.d. group activity when G is small (as more correlation information can be utilized)

Coherence and sub-coherence at $N = 1000$



(a) Coherence versus L/N in the independent case at $p_1/p_2 = 3$.

(b) Sub-coherence within one group versus L/N in the correlated case.

Figure: Coherence and sub-coherence at $N = 1000$, $p = 0.1$, $M = 16$.

- Coherence for device activity detection in the independent case is the same as that for channel estimation in the independent case
  - The proposed approaches for jointly sparse support recovery and signal recovery with the AMP-based decoder share the same auto-encoder structure and training process
- The learned measurement matrix has smaller coherence and sub-coherence (to more effectively differentiate devices (may) active simultaneously)

Block-coherence at $N = 1000$

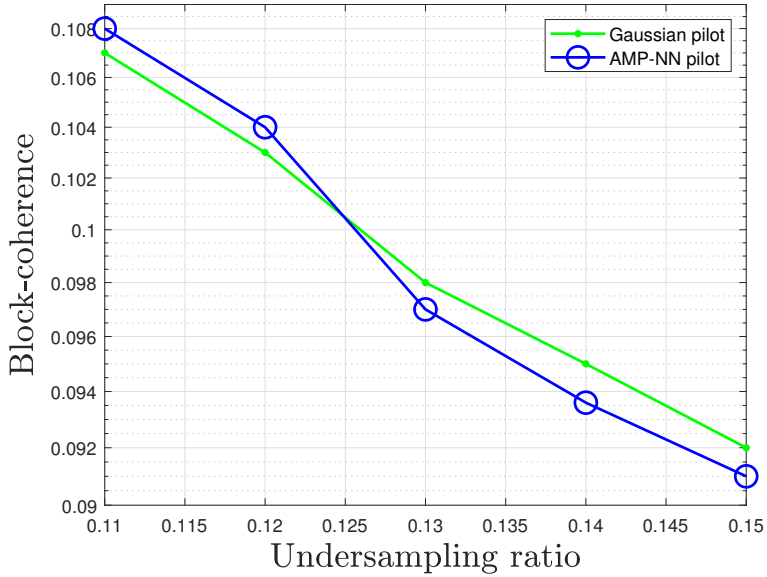
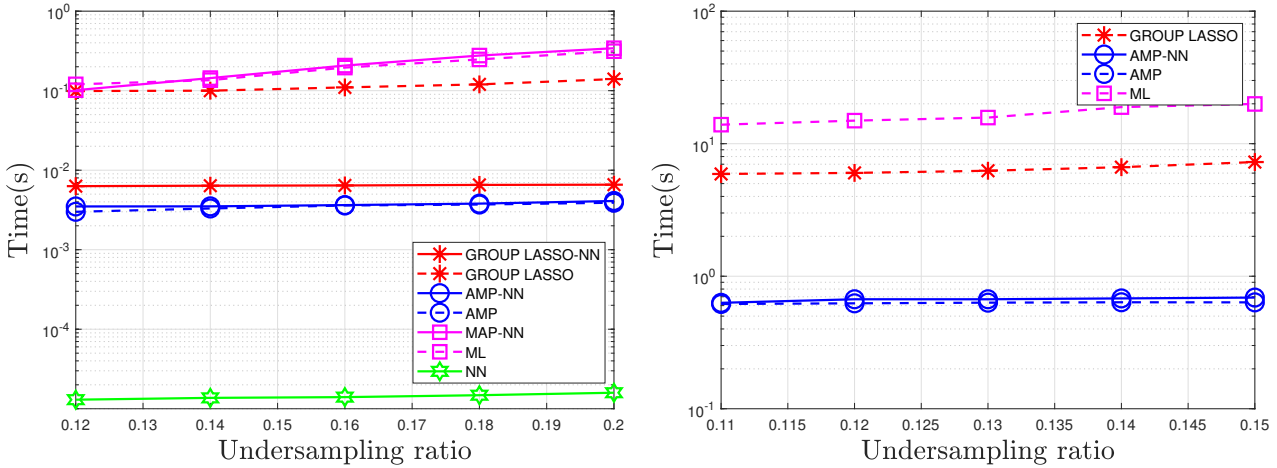


Figure: Block-coherence across groups versus L/N in the correlated case at $N = 1000$, $p = 0.1$, $M = 16$.

- The learned measurement matrix has larger block-coherence when L/N is small
  - The differentiability of the devices that have a smaller chance of being active is sacrificed at small L/N
- The learned measurement matrix has smaller block-coherence when L/N is large
  - The differentiability of the devices that have a smaller chance of being active can be taken into account at large L/N

Computation time at $N = 100$ and $N = 1000$



(a) Computation time versus L/N at $M = 4$, $N = 100$.

(b) Computation time versus L/N at $M = 16$, $N = 1000$.

Figure: Computation time in the independent case at $p = 0.1$, $p_1/p_2 = 3$.

- AMP-NN and MAP-NN have similar computation time to AMP and ML, respectively (due to the same number of iterations and $V = 3$)
- GROUP LASSO and ML have much longer computation time than others, especially at large N, and may not be applicable for practical mMTC
- When $N = 100$, NN has the shortest computation time ($U = 0$)
- When $N = 1000$, AMP-NN and AMP have the shortest computation time ($V = 3$)

Outline

Conclusion

- We propose two-model driven approaches using the standard auto-encoder structure for real numbers
  - One aims to jointly design the common measurement matrix and jointly sparse signal recovery methods for complex signals, and can be used in the joint design of pilot sequences and channel estimation methods
  - The other is to jointly design the common measurement matrix and jointly sparse support recovery methods for complex signals, and can be applied to the joint design of pilot sequences and device activity detection
- We propose the Group LASSO-based decoder and AMP-based decoder, as instances for the model-driven decoder for jointly sparse signal recovery
- We propose the covariance-based decoder, MAP-based decoder and AMP-based decoder, as instances for the model-driven decoder for jointly sparse support recovery

Conclusion

- The proposed model-driven approaches can greatly benefit from the underlying advanced methods with theoretical performance guarantee via the approximation parts of the model-driven decoders
- The proposed model-driven approaches can also effectively utilize features of sparsity patterns in designing the encoders for obtaining effective common measure matrices and adjusting the correction parts of the model-driven decoders
- The proposed model-driven approaches can provide higher recovery accuracy with shorter computation time than the underlying advanced recovery methods
- The obtained results are of critical importance for achieving massive access in mMTC

Publications

- S. Li, W. Zhang, Y. Cui*, H. V. Cheng and W. Yu, "Joint design of measurement matrix and sparse support recovery method via deep auto-encoder," IEEE Signal Process. Lett., vol. 26, no. 12, pp. 1778-1782, Dec. 2019.
- Y. Cui*, S. Li and W. Zhang, "Jointly Sparse Signal Recovery and Support Recovery via Deep Learning with Applications in MIMO-based Grant-Free Random Access," to appear in IEEE J. Select. Areas Commun., 2020.
- S. Li, W. Zhang, and Y. Cui, "Jointly sparse signal recovery via deep auto-encoder and parallel coordinate descent unrolling," in IEEE WCNC, 2020, pp. 1–6.
- W. Zhang, S. Li, and Y. Cui, "Jointly sparse support recovery via deep auto-encoder with applications in MIMO-based grant-free random access for mMTC," in IEEE SPAWC, 2020, pp. 1–6.

Reference I

Adler, A., Boublil, D., & Zibulevsky, M. (2017). Block-based compressed sensing of images via deep learning. In *IEEE MMSP* (pp. 1–6).

Candes, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, *346*, 589–592.

Chen, Z., Sohrabi, F., Liu, Y.-F., & Yu, W. (2019). Covariance based joint activity and data detection for massive random access with massive mimo. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (pp. 1–6). IEEE.

Chen, Z., Sohrabi, F., & Yu, W. (2018). Sparse activity detection for massive connectivity. *IEEE Trans. Signal Process.*, *66*, 1890–1904. doi:10.1109/TSP.2018.2795540.

Reference II

Donoho, D. L., Maleki, A., & Montanari, A. (2009). Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, *106*, 18914–18919.

Eldar, Y. C., Kuppinger, P., & Bolcskei, H. (2010). Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Trans. Signal Process.*, *58*, 3042–3054.

Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *ICML* (pp. 399–406).

Haghighatshoar, S., Jung, P., & Caire, G. (2018). Improved scaling law for activity detection in massive MIMO systems. In *IEEE ISIT* (pp. 381–385). doi:10.1109/ISIT.2018.8437359.

He, H., Wen, C.-K., Jin, S., & Li, G. Y. (2018). Deep learning-based channel estimation for beamspace mmwave massive mimo systems. *IEEE Wireless Commun. Lett.*, *7*, 852–855.

Reference III

Jiang, D., & Cui, Y. (2020). Ml estimation and map estimation for device activity in grant-free massive access with interference. In *IEEE WCNC* (pp. 1–6).

Ke, M., Gao, Z., Wu, Y., Gao, X., & Schober, R. (2020). Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive mimo. *IEEE Transactions on Signal Processing*, *68*, 764–779.

Kingma, D., Ba, L. et al. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Li, S., Zhang, W., Cui, Y., Cheng, H., & Yu, W. (2019). Joint design of measurement matrix and sparse support recovery method via deep auto-encoder. *IEEE Signal Process. Lett*, (pp. 1–1). doi:10.1109/LSP.2019.2945683.

Reference IV

Liu, L., Larsson, E. G., Yu, W., Popovski, P., Stefanovic, C., & De Carvalho, E. (2018). Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things. *IEEE Signal Process. Mag.*, *35*, 88–99.

Liu, L., & Yu, W. (2018). Massive connectivity with massive MIMO-Part I: Device activity detection and channel estimation. *IEEE Trans. Signal Process.*, *66*, 2933–2946. doi:10.1109/TSP.2018.2818082.

Mousavi, A., Dasarathy, G., & Baraniuk, R. G. (2017). Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks. In *55th Allerton* (pp. 744–744). doi:10.1109/ALLERTON.2017.8262812.

Reference V

Nguyen, D. M., Tsiligianni, E., & Deligiannis, N. (2017). Deep learning sparse ternary projections for compressed sensing of images. In *IEEE GlobalSIP* (pp. 1125–1129).

Pal, P., & Vaidyanathan, P. P. (2015). Pushing the limits of sparse support recovery using correlation information. *IEEE Trans. Signal Process.*, *63*, 711–726. doi:10.1109/TSP.2014.2385033.

Qin, Z., Scheinberg, K., & Goldfarb, D. (2013). Efficient block-coordinate descent algorithms for the group lasso. *Mathematical Programming Computation*, *5*, 143–169.

Senel, K., & Larsson, E. G. (2018). Grant-free massive mtc-enabled massive mimo: A compressive sensing approach. *IEEE Trans. Commun.*, *66*, 6124–6175. doi:10.1109/TSP.2018.2818082.

Taha, A., Alrabeiah, M., & Alkhateeb, A. (2019). Enabling large intelligent surfaces with compressive sensing and deep learning. *arXiv preprint arXiv:1904.10136*, .

Reference VI

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*, 267–288.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Trans. Inf. Theory*, *55*, 2183–2202. doi:10.1109/TIT.2009.2016018.

Wei, C., Liu, H., Zhang, Z., Dang, J., & Wu, L. (2016). Approximate message passing-based joint user activity and data detection for noma. *IEEE Communications Letters*, *21*, 640–643.

Wen, C., Shih, W., & Jin, S. (2018). Deep learning for massive MIMO CSI feedback. *IEEE Commun. Lett*, *7*, 748–751. doi:10.1109/LWC.2018.2818160.

Reference VII

Wu, S., Dimakis, A., Sanghavi, S., Yu, F., Holtmann-Rice, D., Storcheus, D., Rostamizadeh, A., & Kumar, S. (2019). Learning a compressed sensing measurement matrix via gradient unrolling. In *ICML* (pp. 6828–6839).

Yang, Y., Sun, J., Li, H., & Xu, Z. (2018). Admm-csnet: A deep learning approach for image compressive sensing. *IEEE transactions on pattern analysis and machine intelligence*, .

Yao, S., Zhao, Y., Zhang, A., Su, L., & Abdelzaher, T. (2017). Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*.

Zhang, J., He, H., Wen, C.-K., Jin, S., & Li, G. Y. (2019). Deep learning based on orthogonal approximate message passing for cp-free ofdm. In *IEEE ICASSP* (pp. 8414–8418).

Reference VIII

Ziniel, J., & Schniter, P. (2012). Efficient high-dimensional inference in the multiple measurement vector problem. *IEEE Trans. Signal Process.*, *61*, 340–354.