A Learning-only Method for Multi-Cell Multi-User MIMO Sum Rate Maximization

Qingyu Song†, Juncheng Wang‡, Jingzong Li§, Guochen Liu¶, Hong Xu†

†The Chinese University of Hong Kong, ‡Hong Kong Baptist University,
§City University of Hong Kong, ¶Huawei

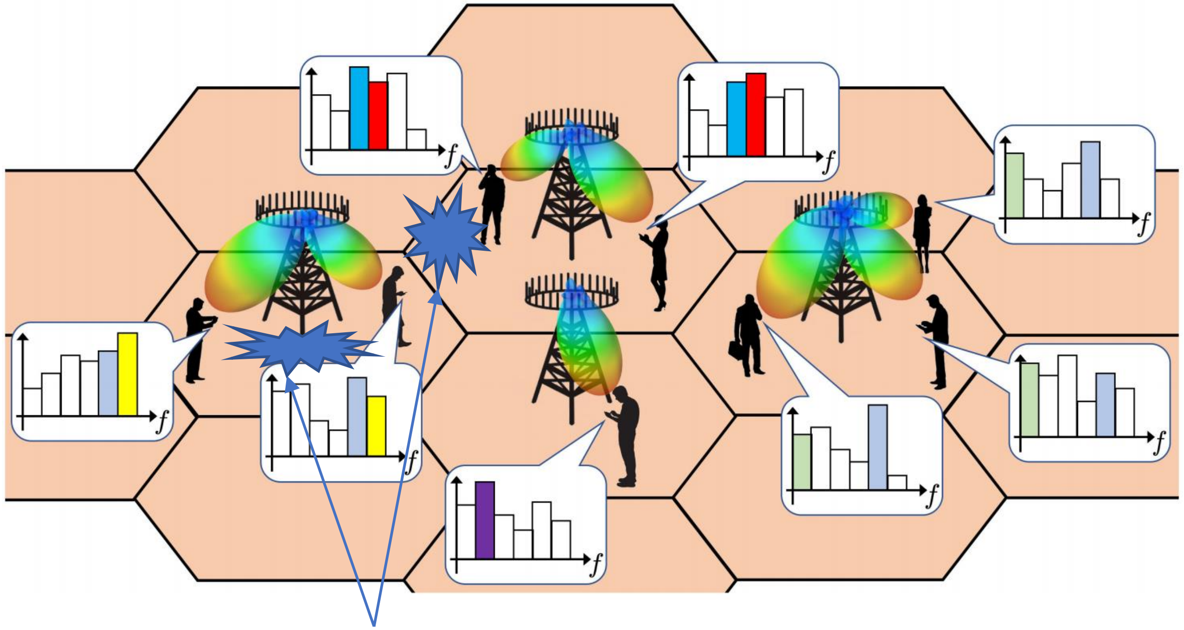
Interference in Multi-Cell Multi-User MIMO

Download-link in a cellular network:

- Sender: Base stations.
- Receiver: Users (equipment).
- Signal: y, x.
- Channel: $\boldsymbol{H}$.

Receiving signal on user side:

$$y_{1,1} = \boldsymbol{H}_{1,1} \cdot (x_{1,1} + \boxed{x_{1,2}})$$



Interference

Picture from K.-K. Wong, G. Liu, W. Cun, W. Zhang, M. Zhao, and Z. Zheng. Truly Distributed Multicell Multi-Band Multiuser MIMO by Synergizing Game Theory and Deep Learning. IEEE Access, vol. 9, pp. 30347-30358, 2021.

Classical Problem Formulation

Interference Reduction by **Precoding** $\boldsymbol{V}$

$$y_{1,1} = \boldsymbol{H}_{1,1} \cdot (x_{1,1} + x_{1,2})$$

⇩

$$y_{1,1} = \boldsymbol{H}_{1,1} \cdot (\boldsymbol{V}_{1,1} x_{1,1} + \boldsymbol{V}_{1,2} x_{1,2})$$

Signal-to-Interference-plus-Noise Ratio (SINR)

$$\text{SINR} = \frac{\textbf{Signal}}{\textbf{Interference + Noise}}$$

Formulation: Sum Rate (SINR) Maximization Problem

Non-triviality: Non-convex, NP-hard in some cases [IEEE JSTSP' 08].

Algorithms for Sum Rate Maximization

- Conventional Algorithms
  - Zero Forcing Scheme: Orthogonalization with SVD
    - Efficient (one shot).
    - Poor optimality in ill-conditioning cases.
  - WMMSE Algorithm [IEEE TSP’ 11]: Block Coordinate Descent
    - State-of-the-art optimality.
    - Auxiliary variables.
    - First order derivative condition for local optimality.
- Learning-related Methods
  - Learning-only Approaches
  - Learning-assisted Approaches

Timeline for Learning-related Methods

Learning-only Approaches

TSP-DNN (IEEE TSP' 17)
Vanilla DNN, Supervised learning

PCNets (IEEE JSAC' 20)
Ensembled DNN, Unsupervised learning

DDPG DQN (IEEE Access' 21)

HetGNN, PCGNN (IEEE TWC' 21, 22)
Message Passing Graph Neural Networks
Supervised + unsupervised learning

Degenerated SISO problems only.
Poor optimality on MIMO problems.
Efficient: Full parallel on GPU.

Learning-assisted Approaches

IAIDNN (IEEE TWC' 21)
Approximate matrix inversion in WMMSE.

GCNWMMSE (IEEE TWC' 23)
Heuristically approximate matrix operations in WMMSE.

SOTA on optimality over all learning related approaches.
Inefficient: Serially update users.

Can we keep both optimality and efficiency?

- We need to combine the two kinds of learning-related methods.

- What?
  - We try to improve the **learning-only method.**
- Why?
  - Efficiency is more important.
  - Learning-assisted approaches are limited by the backboned algorithm (vanilla WMMSE).
- How?
  - Observation: Existing learning-only methods are black-box.
  - We try to learn from the SOTA non-learning algorithm (WMMSE) and the optimization problem itself.

Learn from WMMSE: Structural Solution Update

- Q: Can we make a more white-box layer in learning-only model?
- Observation: In each iteration, WMMSE computation can be transferred into one-line form.

$$
\begin{cases}
\mathbf{U}_{b,u} = \Big(\sum_{(b,u)} \mathbf{H}_{b,u}\mathbf{V}_{b,u}\mathbf{V}_{b,u}^H\mathbf{H}_{b,u}^H + \sigma_{b,u}^2\mathbf{I}\Big)^{-1}\mathbf{H}_{b,u}\mathbf{V}_{b,u}, \\
\mathbf{W}_{b,u} = (\mathbf{I} - \mathbf{U}_{b,u}^H\mathbf{H}_{b,u}\mathbf{V}_{b,u})^{-1}, \\
\mathbf{V}_{b,u} = \Big(\sum_{(b,u)} \mathbf{H}_{b,u}^H\mathbf{U}_{b,u}\mathbf{W}_{b,u}\mathbf{U}_{b,u}^H\mathbf{H}_{b,u} + \mu_k^*\mathbf{I}\Big)^{-1} \mathbf{H}_{b,u}^H\mathbf{U}_{b,u}\mathbf{W}_{b,u},
\end{cases}
$$

$$
\underbrace{\left(\sum_{(b,u)} \mathbf{H}_{b,u}^H\mathbf{U}_{b,u}\mathbf{W}_{b,u}\mathbf{U}_{b,u}^H\mathbf{H}_{b,u} + \mu_k^*\mathbf{I}\right)^{-1} \mathbf{H}_{b,u}^H \left(\sum_{(b,u)} \mathbf{H}_{b,u}\mathbf{V}_{b,u}\mathbf{V}_{b,u}^H\mathbf{H}_{b,u}^H + \sigma_{b,u}^2\mathbf{I}\right)^{-1} \mathbf{H}_{b,u}}_{①} \boxed{\mathbf{V}_{b,u}} \underbrace{\mathbf{W}_{b,u}}_{②}, \quad u \in \mathcal{U}, b \in \mathcal{B}.
$$

Per Iteration Structural Solution Update

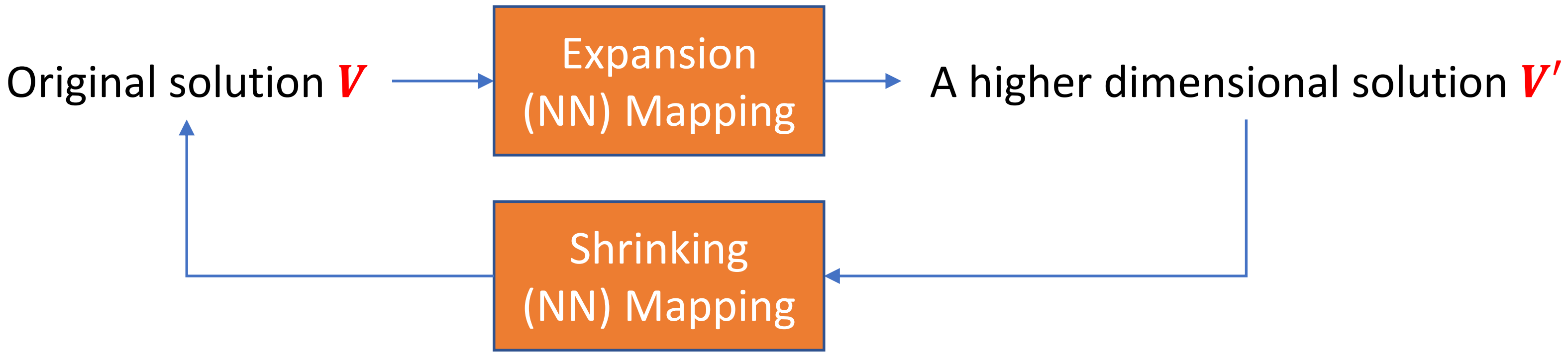
- Unrolling:
  - Sophisticated computations in ① ② with two learnable parameter matrices (motivated by [Liu et al. ICML' 23]).

$$\underbrace{\left(\sum_{(b,u)} \mathbf{H}_{b,u}^H \mathbf{U}_{b,u} \mathbf{W}_{b,u} \mathbf{U}_{b,u}^H \mathbf{H}_{b,u} + \mu_k^* \mathbf{I}\right)^{-1} \mathbf{H}_{b,u}^H \left(\sum_{(b,u)} \mathbf{H}_{b,u} \mathbf{V}_{b,u} \mathbf{V}_{b,u}^H \mathbf{H}_{b,u}^H + \sigma_{b,u}^2 \mathbf{I}\right)^{-1} \mathbf{H}_{b,u}}_{①\ \mathbf{W}_L} \mathbf{V}_{b,u} \underbrace{\mathbf{W}_{b,u}}_{②\ \mathbf{W}_R}, \quad u \in \mathcal{U}, b \in \mathcal{B}.$$

- Modeling:
  - Generate $\mathbf{W}_L$ and $\mathbf{W}_R$ from $\mathbf{V}$ and $\mathbf{H}$.
  - Construct powerful neural network models.

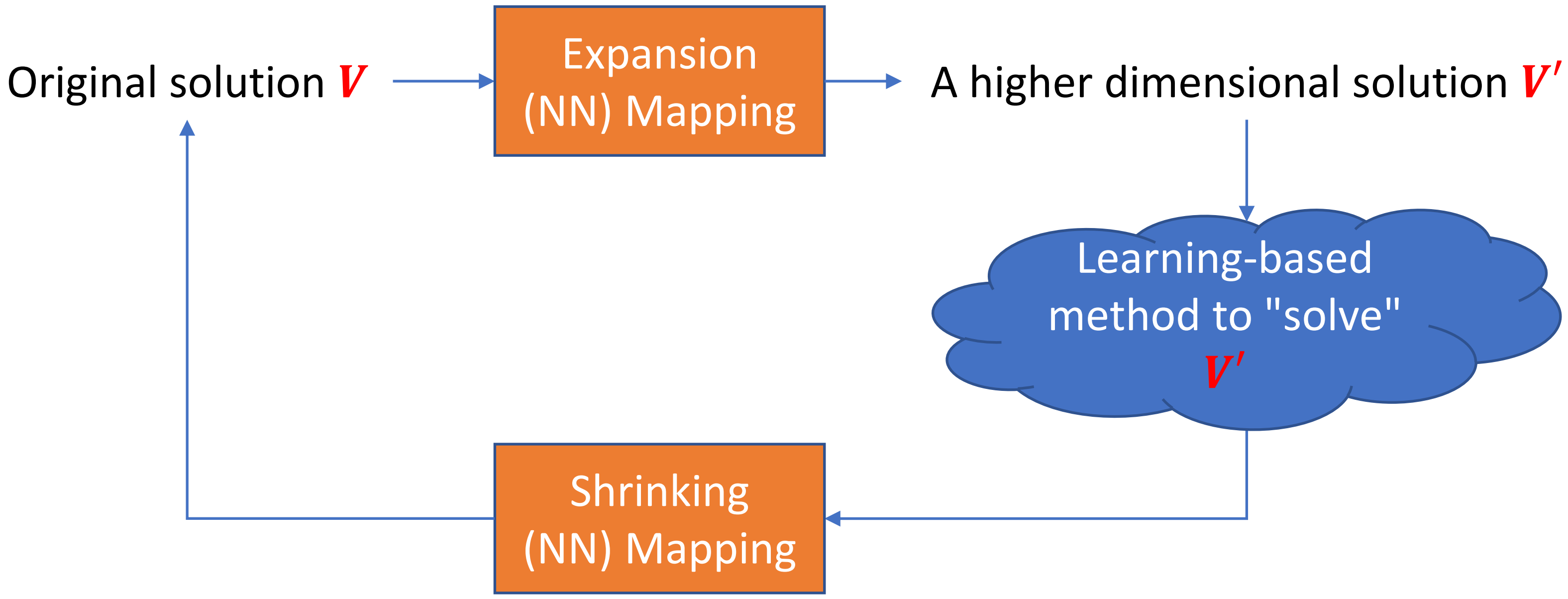
Dimension Expansion and Shrink in Learning-Only Methods

- Observation: <u>Benefits of solving problem in a higher dimensional space</u>
  - The problem is trivially solvable in a higher dimensional space.
    - More transmitting antennas facilitates solvability: Use distinct antenna to serve each user.
  - Increase number of learnable parameters in neural network.
- Workflow
  - Round-trip (neural network) mapping: Get solution of original space.



Requirement of Round-trip Mapping

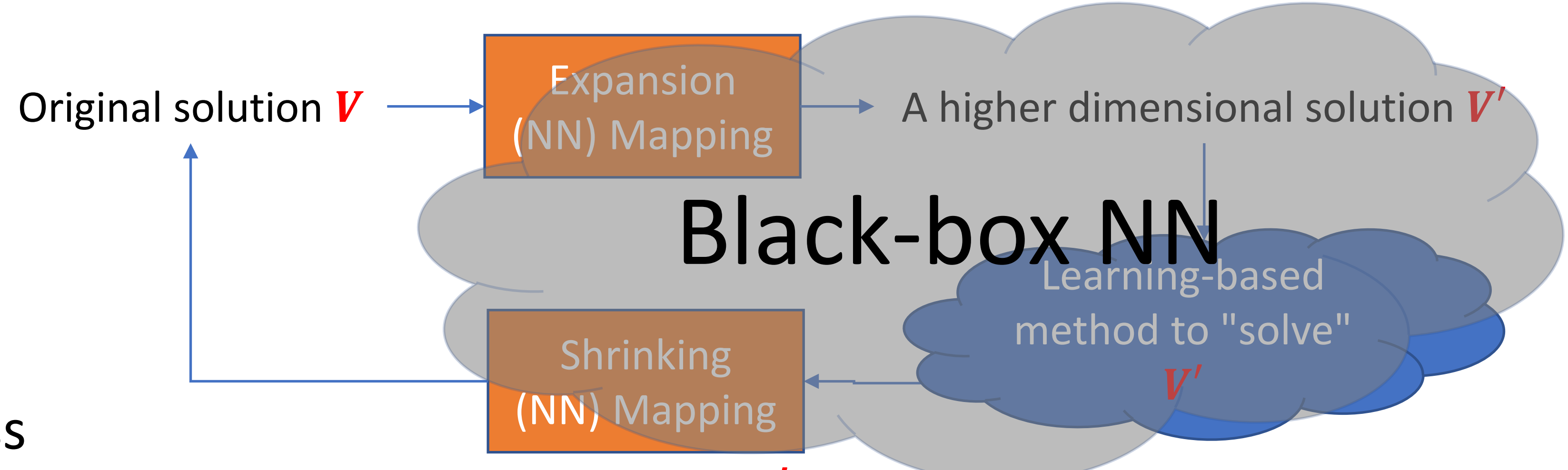
- Equivalence between V and V'
  - [Ideal] On optimality: Solution of V' (**<=**) **=>** solution of V
    - Optimality of V' is also non-deterministic (rely on training) in learning-only framework.
  - <u>[Relaxed] On feasibility: V' is feasible (**<=**) **=>** V is feasible</u>



Existing Mapping Methods in Learning-only Framework

- **Normalization** after Neural Network Mapping
  - Since the constraints are quadratic
  - No equivalence between $\boldsymbol{V}$ and $\boldsymbol{V'}$

$$V_T^{b,u} = \frac{V_T^{b,u}}{\sum_i \sqrt{Tr\left(V_T^{b,i}\left(V_T^{b,i}\right)^H\right)}} \cdot \sqrt{P_k}, u \in \{1, \ldots, U\}$$



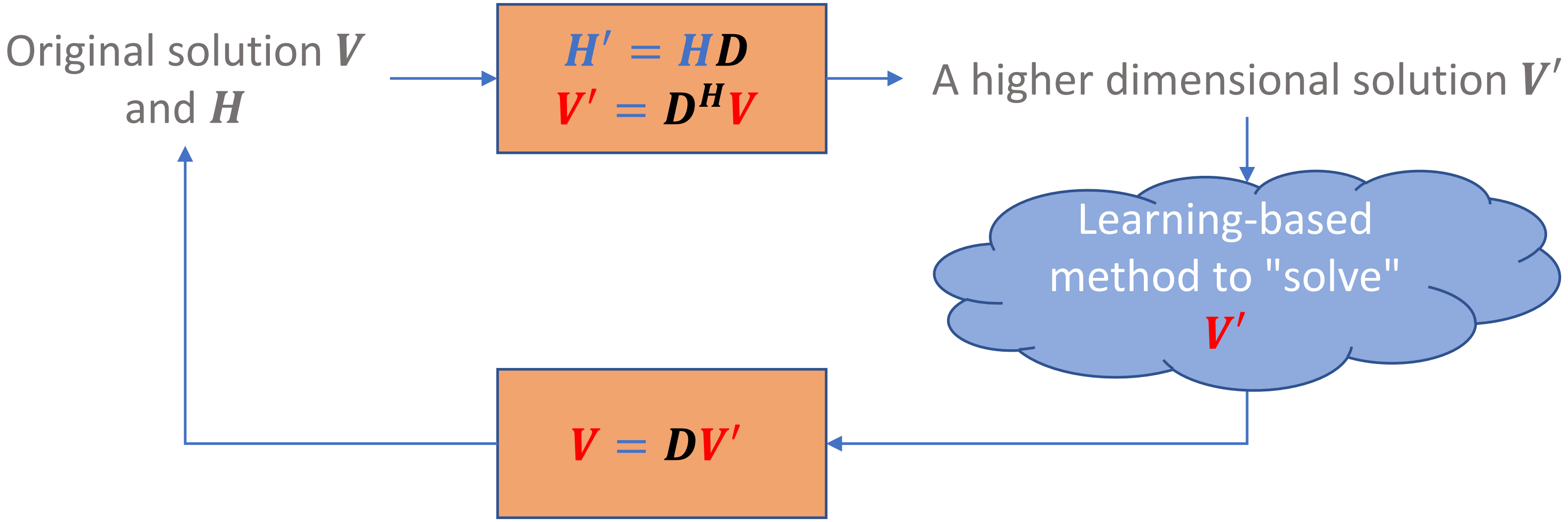
- Weakness
  - No equivalence **<=>** Black-box model **<=>** $\boldsymbol{V'}$ is infeasible.
- We propose a learnable unitary matrix
  - $\boldsymbol{V'}$ is infeasible - > $\boldsymbol{V}$ is infeasible
  - Shared among all iterations

Learn from Sum Rate Maximization Problem: Learnable Unitary Matrix

- Lemma (Lemma 1 in paper): Any unitary matrix ensures feasibility of SINR maximization problem.
  - Proved by von Neumann's trace inequality.

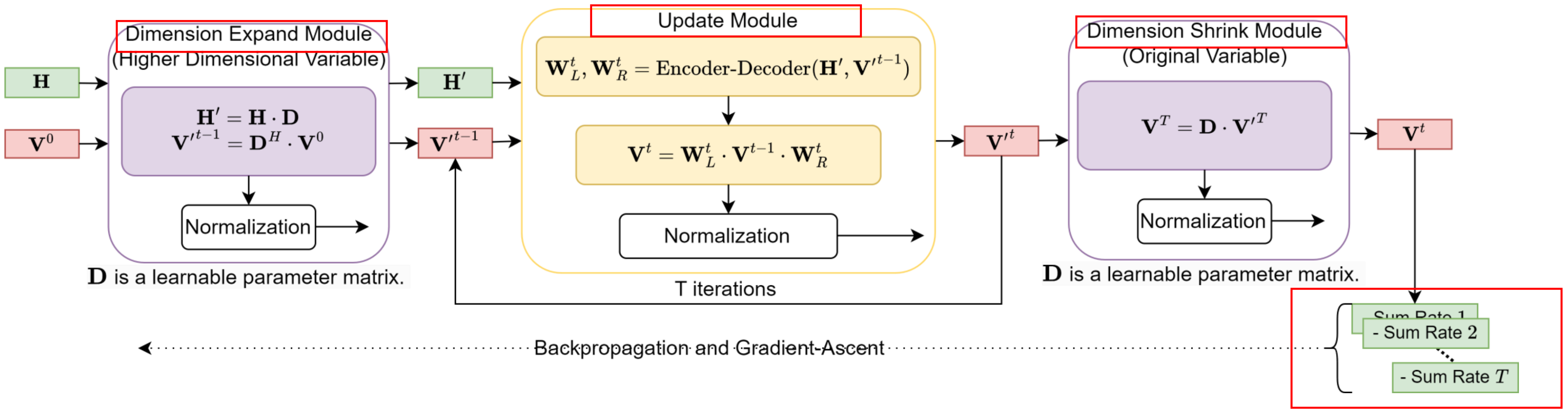
$$\boldsymbol{H}_{b,u}\boldsymbol{V}_{b,u} = \boldsymbol{H}_{b,u}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{H}}\boldsymbol{V}_{b,u}, \text{ if } \boldsymbol{D}\boldsymbol{D}^{\boldsymbol{H}} = \boldsymbol{I}, \forall b, u$$

- Workflow: with a learnable $\boldsymbol{D}$



System Overview

- Two main modules:
  - Update
  - Dimension Transformation: Expand and Shrink
- Fixed-Number (T) Iterations



Models Construction: Overview

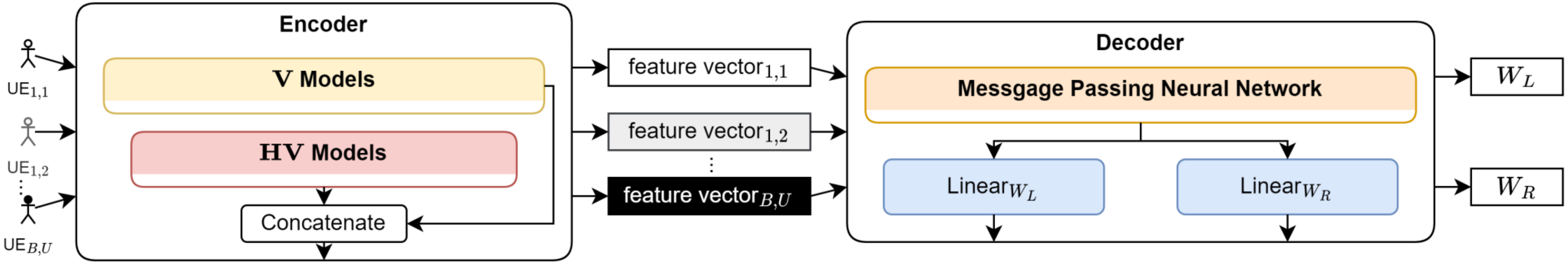
- Multi-agent System:
  - Users (equipment) are regarded as agents.
  - Channel $\mathbf{H}$ are shared among users.
- Feature selection from optimization problem:
  - Sum rate maximization with per-cell power budget constraints

$$\mathbf{P}:\max_{\mathbf{V}} \quad \sum_{u\in\mathcal{U}}\sum_{b\in\mathcal{B}} \log_2 \det\left(\mathbf{I} + \left(\mathbf{H}_{b,u}\mathbf{V}_{b,u}\right)\left(\mathbf{H}_{b,u}\mathbf{V}_{b,u}\right)^H \left(\sum_{\substack{\tilde{b}\in\mathcal{B},\tilde{u}\in\mathcal{U}\\(\tilde{b},\tilde{u})\neq(b,u)}} \left(\mathbf{H}_{\tilde{b},u}\mathbf{V}_{\tilde{b},\tilde{u}}\right)\left(\mathbf{H}_{\tilde{b},u}\mathbf{V}_{\tilde{b},\tilde{u}}\right)^H + \sigma_{b,u}^2\mathbf{I}\right)^{-1}\right)$$

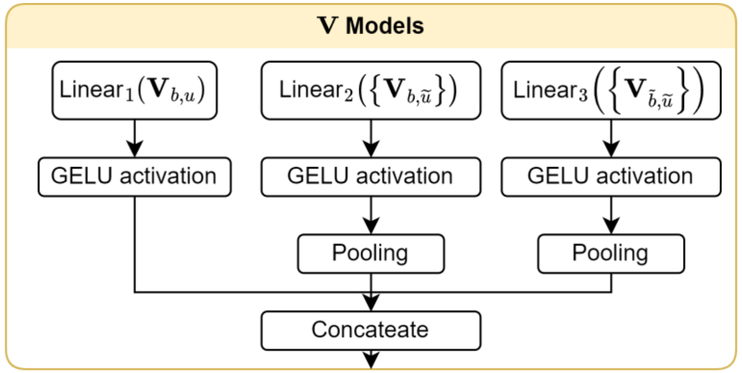
$$\text{s.t.} \quad \sum_{u\in\mathcal{U}} \mathrm{Tr}\left(\mathbf{V}_{b,u}\mathbf{V}_{b,u}^H\right) \leq P, \quad b\in\mathcal{B}$$

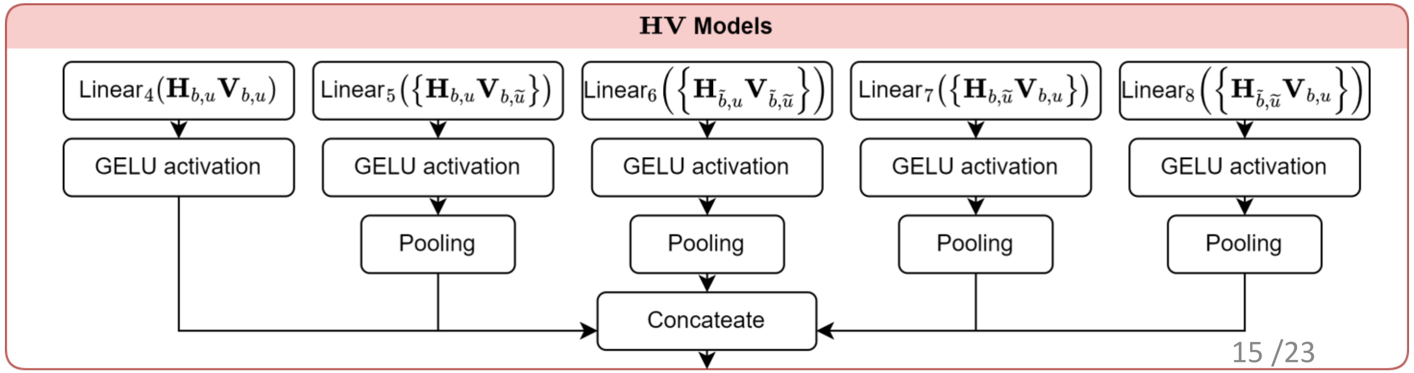
  - For each user b, u, we choose $\mathbf{V}$ and $\mathbf{HV}$.

Encoder-Decoder Framework



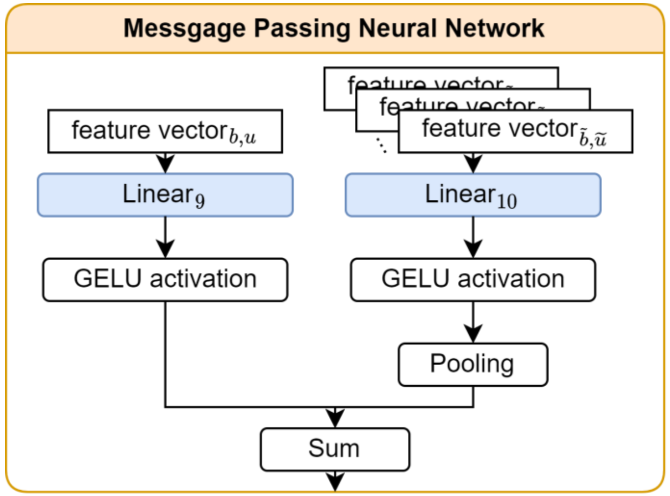
- Different models for **V** and **HV** features extractions
  - Group features by physical meaning:
    - Target, intra-cell interference, inter-cell interference, interference to intra-cell users, interference to inter-cell users.





Decoder Model: Message Passing Neural Network

- A graph neural network on complete graph
  - Explicit feature exchange among users.



Overal Complexity Analysis

- Workflow
  - Calculate all matrix multiplications: $\mathbf{HV}$ computation and neural networks.
- Basic result (after parallelism over all users): $\mathcal{O}\left(3d * N_r * N_t^3\right)$
  - d is a hyperparameter (related to the width of neural networks),
  - N_r N_t are numbers of receiving antenna (one user) and transmitting antenna (one base station), respectively.
  - $N_r \ll N_t$
- Complexity of WMMSE: $\mathcal{O}\left((BU)^2 * N_t^3\right)$
  - After parallelism: $\mathcal{O}\left(BU * N_t^3\right)$
- Improved efficiency: $\mathcal{O}\left(3d * N_r * N_t^3\right) < \mathcal{O}\left((BU)^2 * N_t^3\right)$
  - $N_r \ll BU$
  - d is set to be 2 in all simulations.

Implementation and Simulation Construction

- Complex Neural Network
  - Two models for real part and imaginary part
- Use zero-forcing scheme for initialization
- Competitors:
  - WMMSE
  - Learning-related baselines:
    - Two learning-assisted WMMSE
    - One learning-only graph neural network model
- Environment configuration:
  - Synthetic cellular network, randomly sampled user positions
  - Randomly sampled channel with path loss

Three Scenarios

- B: Number of base station
- U: Number of user
- N_t: Number of transmitting antenna
- N_r: Number of receiving antenna

TABLE II: Scenarios Settings

Properties	**Small Scale**	**Moderate Scale**	**Large Scale**
B	2	2	2
U	4	8	16
N_t	8	16	32
N_r	2	2	2
Data Size	100,000	120,000	180,000

Results: Optimality

- Sum rate over WMMSE

TABLE IV: Sum rate over WMMSE.

Models	Small		Moderate		Large	
StructuralMPNN	98.6%	96.9%	96.9%	95.7%	93.0%	91.5%
GCNWMMSE[5]	100.4%	100.5%	100.2%	100.5%	100.2%	100.3%
IAIDNN[7]	93.2%	94.0%	91.2%	87.9%	92.0%	89.7%
PCGNN[10]	12.7%	12.7%	6.5%	6.5%	-	
StructualMPNN-B	58.7%	60.7%	46.9%	48.7%	-	
StructualMPNN-O	88.3%	86.8%	78.8%	78.6%	-	

Results: Efficiency

- Speedup over WMMSE

TABLE V: Speedup over WMMSE on CPU.

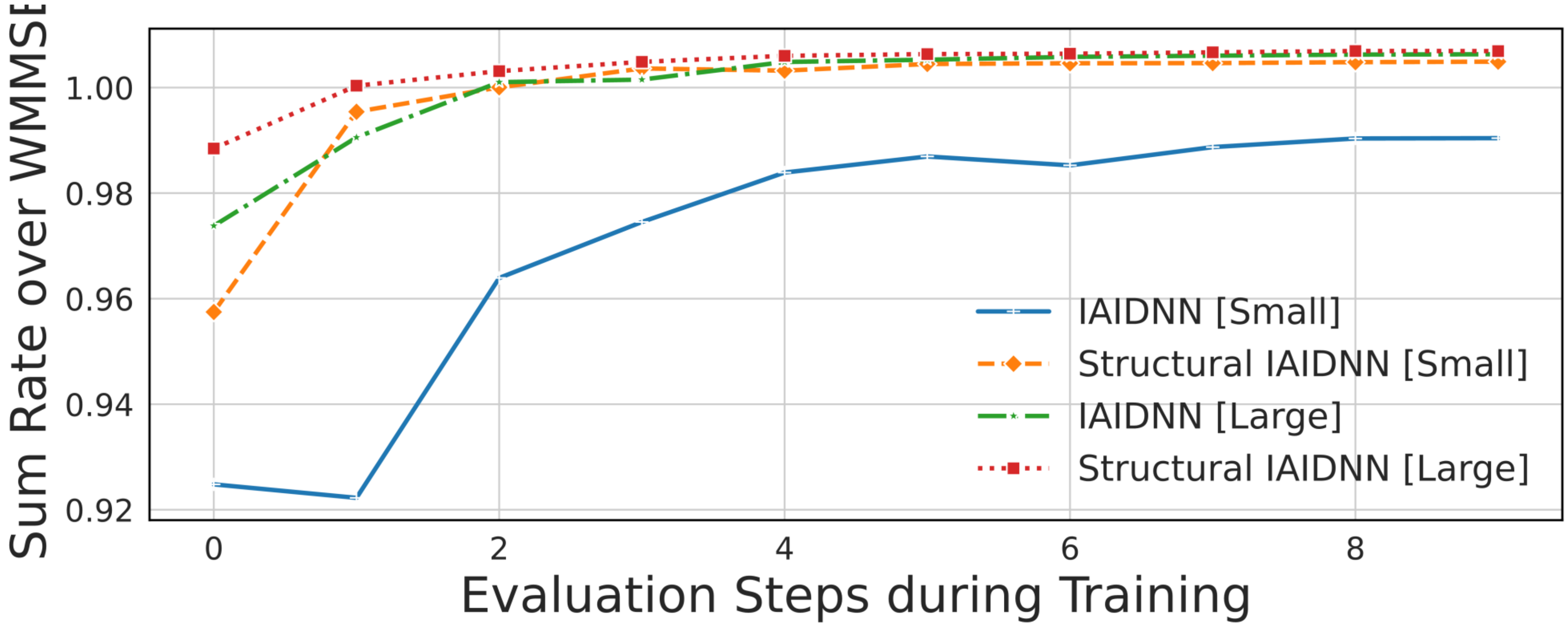
Scale:	Small		Moderate		Large	
Interference:	Small	Large	Small	Large	Small	Large
StructuralMPNN	**3.16**	**4.21**	**4.5**	**4.2**	**6.62**	**6.83**
GCNWMMSE[5]	0.76	1.04	1.16	1.07	3.27	3.33
IAIDNN[7]	1.01	1.39	1.47	1.36	4.04	4.12

TABLE VI: Speedup over WMMSE on GPU.

Scale:	Small		Moderate		Large	
Interference:	Small	Large	Small	Large	Small	Large
StructuralMPNN	**3.11**	**4.26**	**8.45**	**7.77**	**46.82**	**47.85**
GCNWMMSE[5]	0.39	0.55	0.61	0.57	1.66	1.71
IAIDNN[7]	0.57	0.8	0.82	0.77	2.22	2.19

Learnable Unitary Matrix on An Existing Work

- Backbone learning-assisted WMMSE: IAIDNN [IEEE TWC’ 21]



Conclusion

- First learning-only approach for MIMO sum rate maximization.

- We propose two schemes:
  - Structural update
  - Dimension transformation

- Achievement:
  - Up to 98% optimality.
  - Up to 47x acceleration.

Thank You!