# Implicit Regularization in Matrix Factorization and Neural Networks with Diagonal Layers


Yikun Hou
Umeå University

Suvrit Sra
Technical University of Munich

Alp Yurtsever
Umeå University


## Matrix Factorization

① Motivation

Matrix Sensing Problem:

- ▶ Recover a *Positive Semidefinite* (PSD) matrix $X \in \mathbb{S}_+^{d\times d}$.
  - ➢ Symmetric measurement matrices $A_1, A_2, \dots, A_n \in \mathbb{S}^{d\times d}$
  - ➢ $\mathcal{A} : \mathbb{R}^{d\times d} \to \mathbb{R}^n$ and $n \ll d^2$
  - ➢ $b = \mathcal{A}(X) = [\langle A_1, X\rangle \cdots \langle A_n, X\rangle]^T \in \mathbb{R}^n$

⇔ Rank-Constrained Optimization Problem:

$$\min_{X\in\mathbb{S}_+^{d\times d}} f(X) \coloneqq \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 \;\; s.t. \;\; \mathrm{rank}(X) \le r$$

② Problem Formulation & Methods

Conventional Method: *Burer-Monteiro (BM) Factorization*

- ▶ Reparametrize $X = UU^T$, and formulate the problem:

$$\min_{U\in\mathbb{R}^{d\times r}} \frac{1}{2}\|\mathcal{A}(UU^T) - b\|_2^2$$

- ▶ Update by gradient descent:

$$U_{k+1} = U_k - \eta\nabla_U f(U_k U_k^T)$$
$$\nabla_U f(UU^T) = 2\nabla f(UU^T)U = 2\mathcal{A}^T(\mathcal{A}(UU^T) - b)U$$

  - ➢ Iteration k = 1, 2,…; Step-size $\eta > 0$.

Proposed Method:

- ▶ Reparametrize $\boldsymbol{X} = \boldsymbol{UDU^T}$ with constraints, and formulate the problem:

$$\min_{\substack{U\in\mathbb{R}^{d\times r}\\ D\in\mathbb{R}^{r\times r}}} \frac{1}{2}\|\mathcal{A}(UDU^T) - b\|_2^2$$
$$s.t. \|U\|_F \le 1,\; D_{ii} \ge 0, D_{ij} = 0, \forall i \text{ and } \forall j \ne i$$

- ▶ Update by projected-gradient descent:

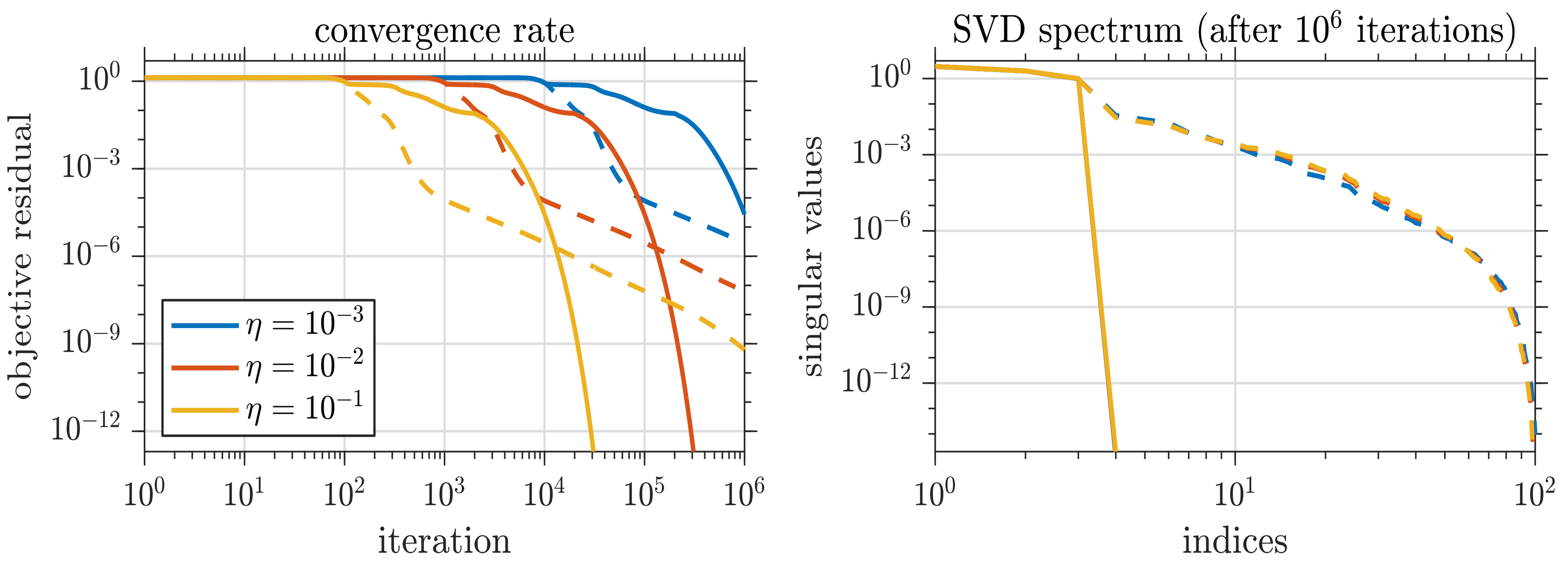
$$U_{k+1} = \Pi_U\left(U_k - \eta\nabla_U f(U_k D_k U_k^T)\right)$$
$$D_{k+1} = \Pi_D\left(D_k - \eta\nabla_D f(U_k D_k U_k^T)\right)$$
$$\nabla_U f(UDU^T) = 2\nabla f(UDU^T)UD, \;\; \nabla_D f(UDU^T) = \mathrm{U}^T\nabla f(UDU^T)U$$

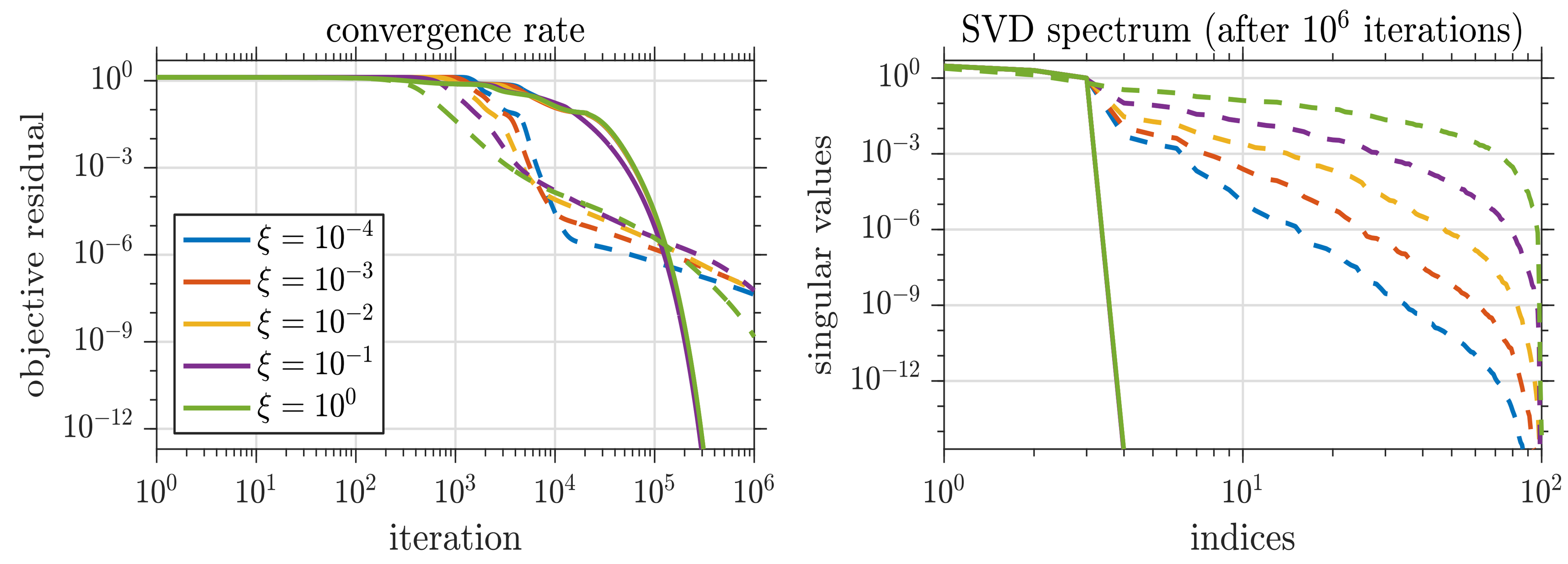
③ Numerical Experiments & Results

- ▶ Recover a PSD matrix $X_\natural = U_\natural U_\natural^T \in \mathbb{R}^{100\times 100}$ from $b \in \mathbb{R}^{900}$
  - ➢ Initialization: $U_\natural \in \mathbb{R}^{100\times 3}$, $\widehat{U}_0 \in \mathbb{R}^{100\times 100}$, $U_0 = \xi \frac{\widehat{U}_0}{\|\widehat{U}_0\|_F}$
  - ➢ Noise setting: $b = b_\natural + \omega$; Noiseless if $\omega = 0$
  - ➢ Impact of initialization $\xi > 0$ and step-size $\eta > 0$

✯ A strong implicit bias toward truly low-rank solution

✯ Regardless of initialization and step-size, both in the settings with consistent and inconsistent measurements.




## Neural Network Application

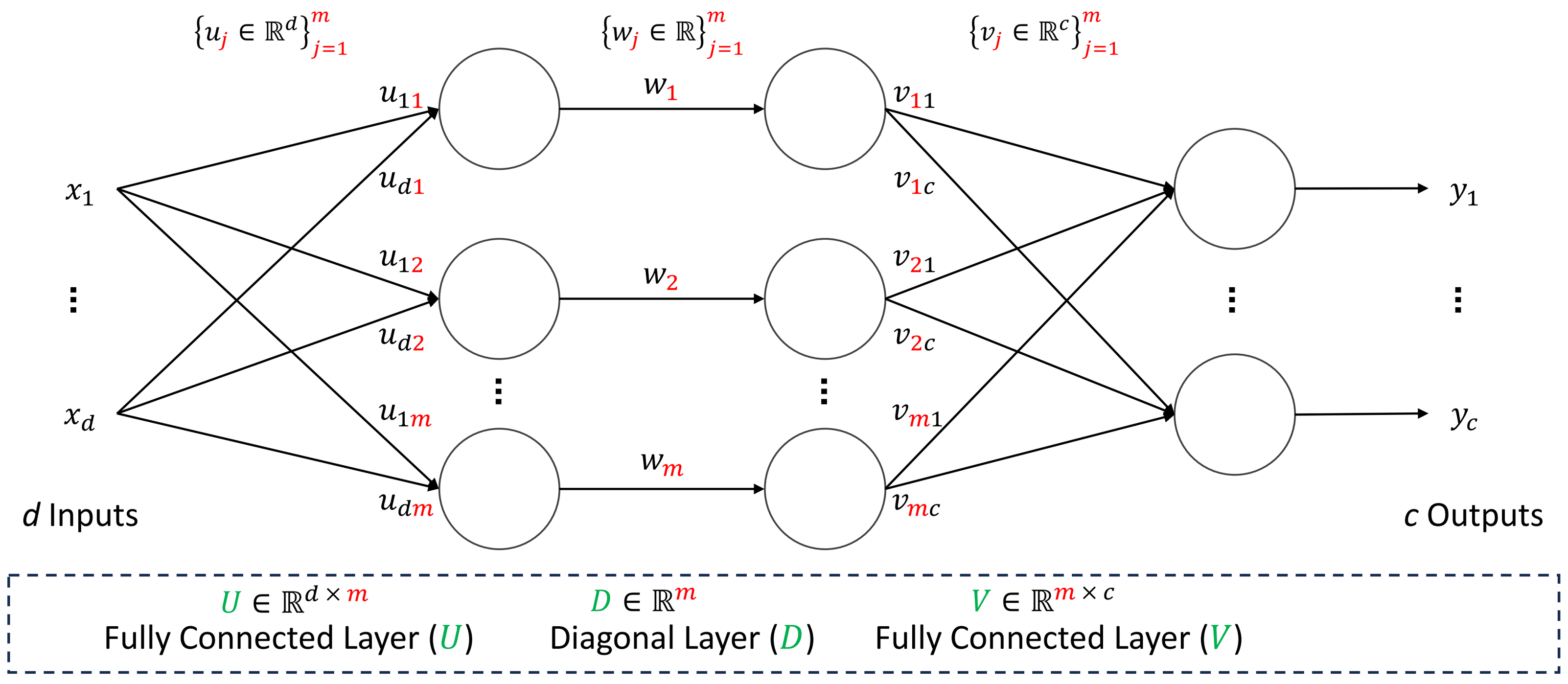
① Problem Formulation & Methods

Training Problem:

- ▶ Generalize $\boldsymbol{UDU^T} \Longrightarrow \boldsymbol{UDV}$:

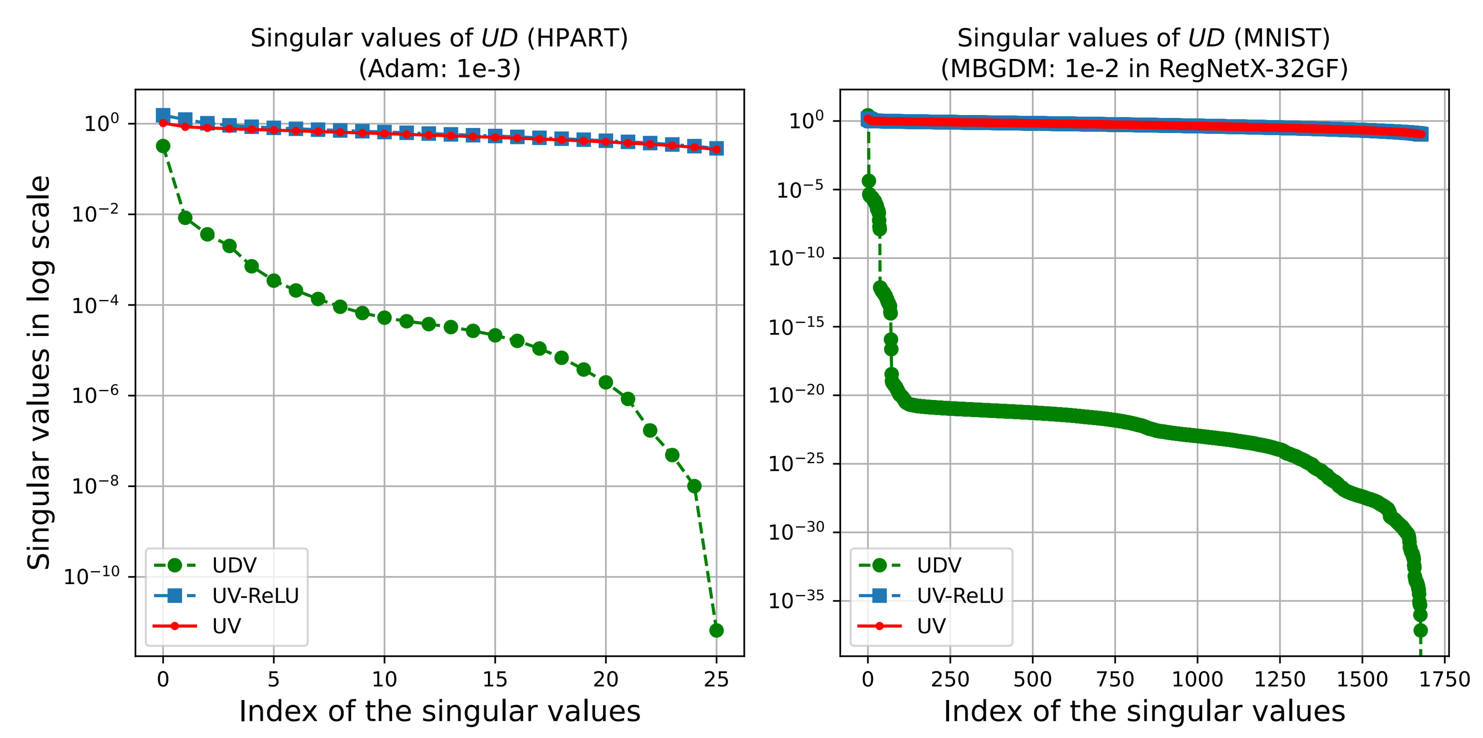
$$\min_{u_j\in\mathbb{R}^d,\, v_j\in\mathbb{R}^c,\, w_j\in\mathbb{R}} \frac{1}{2n}\sum_{i=1}^{n}\left\|\sum_{j=1}^{m} v_j\, w_j\, u_j^T\, x_i - y_i\right\|_2^2$$
$$s.t. \;\; \textstyle\sum_{j=1}^m \|u_j\|_2^2 \le 1, \;\; \sum_{j=1}^m \|v_j\|_2^2 \le 1, \text{ and } w_j \ge 0; \;\; j = 1,2,\dots m$$

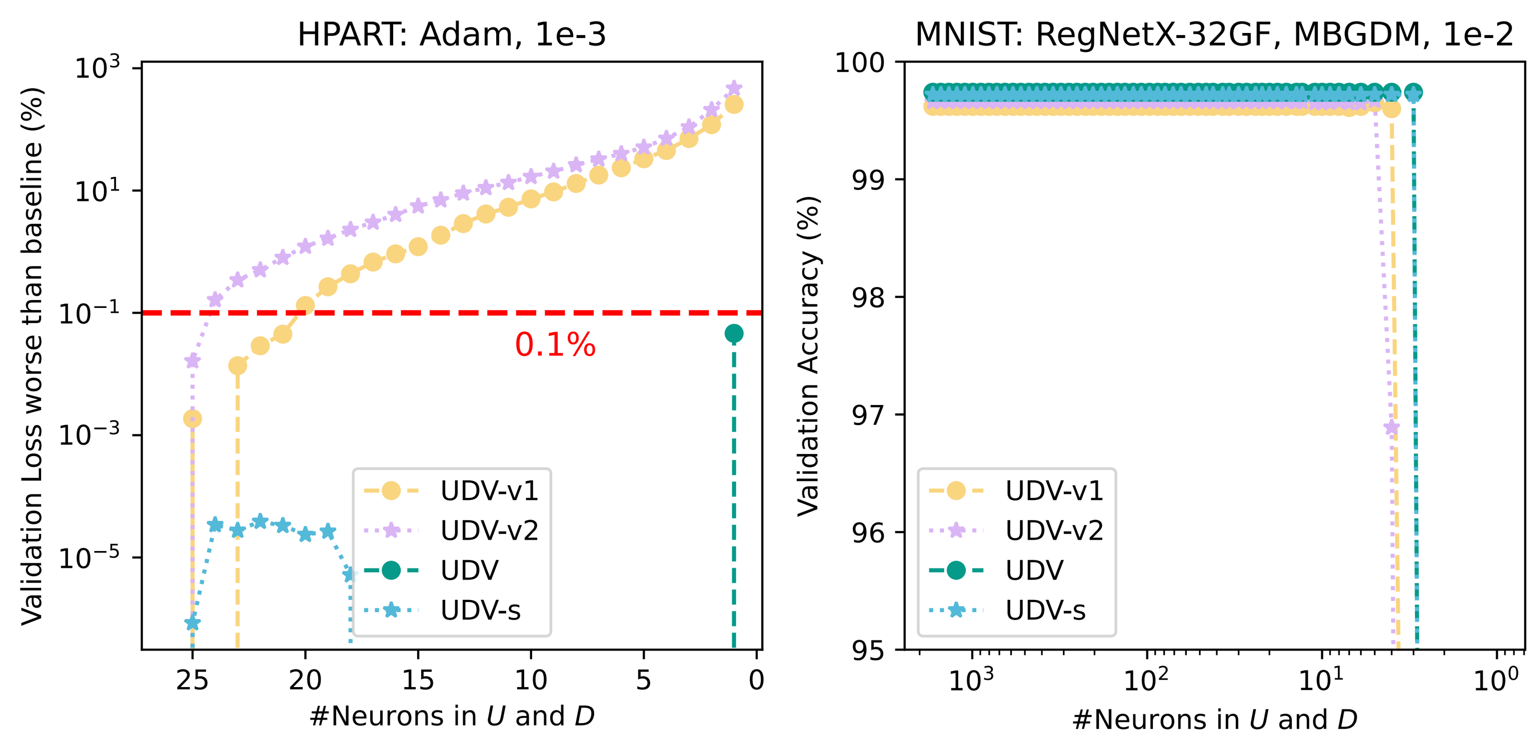
- ▶ UDV classifier in neural networks




② Numerical Experiments & Results

- ▶ Regression / Classification*
- ▶ Singular Value Decomposition (SVD) based pruning




✯ Low-rank solution leads compact and lightweight networks

✯ Competitive performance compared to ReLU networks


UMEÅ UNIVERSITY

WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM