Efficiency Calibration of Implicit Regularization in Deep Networks via Self-paced Curriculum-Driven Singular Value Selection

Zhe Li[1] , **Shuo Chen**[2] , **Jian Yang**[1] and **Lei Luo**[1*]
[1]PCA Lab, Nanjing University of Science and Technology
[2]RIKEN
{zheli, csjyang}@njust.edu.cn, shuo.chen.ya@riken.jp, luoleipitt@gmail.com

Abstract

The generalization of neural networks has been a major focus of research in deep learning. It is often interpreted as an implicit bias towards solutions with specific properties. Especially, in practical applications, it has been observed that linear neural networks (LNN) tend to favor low-rank solutions for matrix completion tasks. However, most existing methods rely on increasing the depth of the neural network to enhance the low rank of solutions, resulting in higher complexity. In this paper, we propose a new explicit regularization method that calibrates the implicit bias towards low-rank trends in matrix completion tasks. Our approach automatically incorporates smaller singular values into the training process using a self-paced learning strategy, gradually restoring matrix information. By jointly using both implicit and explicit regularization, we effectively capture the low-rank structure of LNN and accelerate its convergence. We also analyze how our proposed penalty term interacts with implicit regularization and provide theoretical guarantees for our new model. To evaluate the effectiveness of our method, we conduct a series of experiments on both simulated and real-world data. Our experimental results clearly demonstrate that our method has better robustness and generalization ability compared with other methods.

1 Introduction

In deep learning, the number of parameters in a network is often significantly larger than the amount of available training data. This phenomenon is known as overparameterization. Despite this, stochastic gradient descent can still yield satisfactory generalization results on test data without the need for explicit regularization constraints [Zhang *et al.*, 2021; Hardt *et al.*, 2016]. This is often attributed to the implicit regularization [Neyshabur *et al.*, 2017; Neyshabur *et al.*, 2015; Vardi, 2023; Li *et al.*, 2023] inherent in deep networks, which guides them towards solutions with strong generalization properties. However, there is currently no consensus on the exact reasons behind the effectiveness of this implicit regularization and how it can be further enhanced to improve the representation capabilities of networks.

Significant progress has been made in the study of implicit regularization in nonlinear networks [Zhang *et al.*, 2021; Belkin *et al.*, 2019; Williams *et al.*, 2019; Smith *et al.*, 2021]. However, the complexity of these networks makes it challenging to describe them in a formulaic manner. To facilitate research and eliminate the impact of changes in network expressive power due to increasing depth [Raghu *et al.*, 2017; Arora *et al.*, 2018a], the focus has shifted to exploring linear neural networks (LNN) for traditional machine learning tasks such as matrix completion and factorization. In particular, [Gunasekar *et al.*, 2017] proposes the hypothesis of implicit regularization based on experimental results on matrix factorization. According to this hypothesis, under sufficiently small initialization and learning rates, gradient descent will converge to low-rank solutions, and the implicit regularization can be interpreted as minimizing the nuclear norm (the sum of singular values of a matrix). Further investigation into LNN [Arora *et al.*, 2018b] has revealed that depth could be viewed as a preprocessor, accelerating network convergence.

However, doubts are raised by [Arora *et al.*, 2019; Razin and Cohen, 2020; Li *et al.*, 2021] regarding the theoretical conclusions of [Gunasekar *et al.*, 2017]. Based on experimental reasoning, it has been pointed out that simple mathematical formulas may not accurately describe implicit regularization. Furthermore, [Arora *et al.*, 2019] deduces that increased depth accelerates network convergence by hastening the separation of large and small singular values, enhancing the low-rank properties of deep networks. Building upon [Arora *et al.*, 2019], [Zhao, 2022] proposes a ratio penalty term that improves and utilizes implicit regularization more effectively, achieving stronger low-rank constraints. This penalty term ensures network convergence even in 1-layer networks. Additionally, other researchers [Wu and Su, 2023; Wang *et al.*, 2022; Orvieto *et al.*, 2022; Wu *et al.*, 2022] have explored the dynamic stability, momentum, and noise of networks, offering diverse perspectives for comprehending and studying implicit regularization.

Current research has shown that explicit regularization can improve the performance of neural networks. However, most studies have focused on implicit regularization or adding penalty terms to strengthen constraints, without considering

*Correpsonding Author

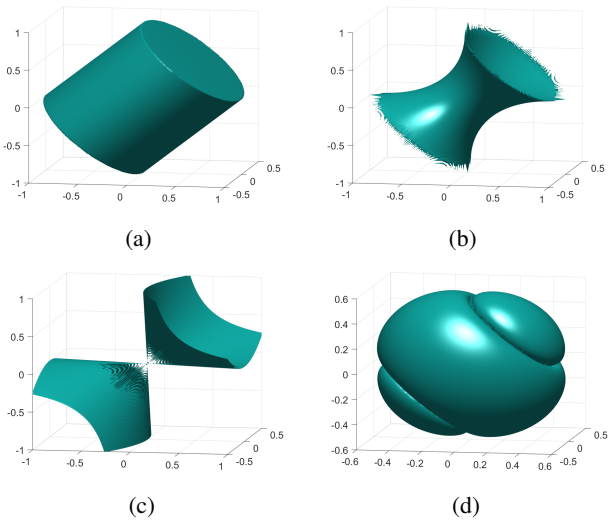


Figure 1: Manifolds of various penalty terms for symmetric 2×2 matrices $M = [a, b; b, c]$. Here, (a) denotes the nuclear norm, (b) refers to the Schatten-p norm [Nie *et al.*, 2012], (c) indicates the ratio norm [Zhao, 2022], and (d) corresponds to the proposed new regularization.

the impact of network depth. Network depth is a crucial factor in designing neural networks, as there needs to be a balance between depth and performance. Additionally, as network depth increases, adding penalty terms can lead to a higher computational burden. Therefore, further research is necessary to determine if implicit regularization is already strong enough to yield adequate low-rank constraints at different network depths and if there are more efficient methods related to explicit regularization to calibrate implicit regularization toward desired properties. The answers to these questions will greatly impact the design and optimization of deep neural networks. To this end, we conduct research using LNN on matrix completion tasks, making the following specific contributions:

- This paper explores the connections between network depth with implicit and explicit regularization. It is observed that as the network depth increases, the addition of explicit regularization does not always bring about performance improvements. This phenomenon helps us design more efficient networks.
- We propose an effective regularizer that takes advantage of self-paced learning and the nuclear norm, resulting in a more precise restoration of matrix information. To our knowledge, this is the first instance of applying self-paced learning to the selection of singular values.
- We theoretically analyze the rationality of our approach. Meanwhile, a series of experiments are conducted on simulated data with varying ranks and real-world data to further validate the reliability of our model.

2 Related Work

Regularization. During the training of neural networks, regularization terms are usually added to the loss function [Zaremba *et al.*, 2014; Krizhevsky *et al.*, 2017] to improve network generalization and prevent overfitting. However, explicit regularization is not always necessary. Studies have shown that implicit regularization [Vardi, 2023] can be obtained by employing optimization algorithms [Hardt *et al.*, 2016; Wu and Su, 2023; Wang *et al.*, 2022] or adjusting the network structure [Li *et al.*, 2023], resulting in strong generalization abilities. In practice, implicit regularization often produces impressive results, making it crucial to analyze its role. This paper investigates the relationship between the proposed penalty term and implicit regularization, as well as the impact of network depth on this interaction.

Matrix Completion. Matrix completion and matrix sensing are essential techniques for research in recommendation systems, information retrieval, and related applications. The commonly used approaches can be categorized into three types: nuclear norm minimization, low-rank factorization [Yan *et al.*, 2013], and minimal rank approximation [Wang *et al.*, 2015]. Traditional algorithms often use nuclear norm minimization to reconstruct matrices. However, the standard nuclear norm, which is one of the earliest methods used [Cai *et al.*, 2010; Candes and Recht, 2012], does not always adequately resolve issues by minimizing all singular values. Subsequent research [Hu *et al.*, 2012] introduced the truncated nuclear norm based on matrix properties, imposing constraints solely on smaller singular values. However, this approach neglects the constraints on larger singular values and applies the same penalization to smaller ones, thus failing to adequately constrain the matrix. [Gu *et al.*, 2014] introduced the Weighted Nuclear Norm (WNN), which assigns different weights to singular values based on their magnitudes for more flexible constraints. Following this, [Kim *et al.*, 2015; Peng *et al.*, 2015; Yang *et al.*, 2018] introduced various adaptations of weighted nuclear norms. Compared with previous methods, the penalty terms we introduce not only impose flexible constraints on singular values but also show superior performance in coping with matrix noise. Additionally, our approach considers the relative magnitudes of the singular values rather than their absolute values when imposing constraints on the matrix.

Self-Paced Learning. The design philosophy behind self-paced learning [Kumar *et al.*, 2010] emulates the human tendency to learn knowledge from simple to complex concepts, representing a branch of curriculum learning [Zhou *et al.*, 2020]. During training, self-paced learning selects the subset with the lowest loss as the easiest part to train. As the number of iterations increases, it gradually expands to cover the entire dataset. Self-paced regularization involves varying weights and can be categorized into three types: Hard [Kumar *et al.*, 2010], Linear [Jiang *et al.*, 2014], and Mixture [Zhao *et al.*, 2015]. Each type is suitable for different scenarios, and incorporating prior knowledge [Zhang *et al.*, 2017; Zhang *et al.*, 2019; Jiang *et al.*, 2015] may enhance performance. Research [Meng *et al.*, 2017; Gong *et al.*, 2016] has shown that self-paced learning can lead to convergence to flatter minima and reduce the impact of noise. In this paper, a new penalty term is proposed based on self-paced learning to impose stronger low-rank constraints, and the matrix comple-

Regularizers	$R(M)$
Nuclear Norm	$\sum_{i=1}^{n} \sigma_i$
Schatten-p Norm [Nie *et al.*, 2012]	$(\sum_{i=1}^{n} \sigma_i^p)^{\frac{1}{p}}$
Weighted Nuclear Norm [Gu *et al.*, 2014]	$\sum_{i=1}^{n} w_i \sigma_i$
Ratio Norm [Zhao, 2022]	$\sum_{i=1}^{n} \frac{\sigma_i}{\|M\|_F}$

Table 1: Commonly used low-rank regularizers for a matrix.

tion task is used to examine the effects of implicit and explicit regularization at different depths.

3 Methodology

In this section, we will provide a detailed explanation of our motivation, methodology, and theoretical justification.

Setup. In the experimental setup, we use an LNN with parameters that can be interpreted as a matrix factorization. Specifically, an N-layer LNN can be formulated as $M = M_N M_{N-1} \dots M_2 M_1$, where M represents the predicted values, and M_i refers to the parameters of the i-th layer of the network. In matrix completion tasks, for a true matrix $\hat{M} \in \mathbb{R}^{m \times n}$, we use $\mathrm{P}_\Omega(\hat{M}) \in \mathrm{R}^{m \times n}$ represents the projection of $\hat{M}$ on the observed dataset Ω, this notation means that the values of the data in the Ω set remain unchanged, while the rest are set to zero. And $\|\cdot\|_F$ denotes the Frobenius norm. Without loss of generality, we assume the matrix dimensions satisfy $n \leq m$. The optimization objective for matrix completion can be written as:

$$\min_M \mathcal{L}(M) = \left\| \mathrm{P}_\Omega(\hat{M}) - \mathrm{P}_\Omega(M) \right\|_F^2 .$$

Motivation. There is a prior assumption in matrices that the magnitude of the singular values represents the amount of information they contain, with larger singular values encompassing the primary information of the matrix, while smaller ones often correspond to noise signals. For example, in movie rating datasets, inaccurate ratings by individuals might introduce a certain degree of noise into the data matrix. Table 1 illustrates the commonly used low-rank norms in matrix completion tasks, where $\sigma_i (i = 1, 2, \dots, n)$ is the singular value of M. Although these methods are somewhat effective, they do not manage to flexibly constrain the singular values while effectively dealing with the matrix noise. To address these issues, we propose a new low-rank regularizer.

According to the definition of self-paced learning [Zhao *et al.*, 2015], at the beginning stages of the training process, parts of the data that contribute to larger losses (i.e., noisy data) are suppressed to minimize their detrimental impact on training. As the training progresses, noisy data is gradually introduced, but due to the initial suppression, the overall convergence direction of the network has been largely determined. At this point, the noise is unlikely to have a substantial effect. The entire training process is continuously guided by useful priors, mitigating the likelihood of falling into unreasonable local optima and enhancing the model's robustness to noise. Utilizing this characteristic, we have adapted its training strategy to the constraint of singular values, suppressing the impact of smaller singular values. In Figure 2, we compared the effects of various penalty terms on handling singular values of different magnitudes. The results show that while other methods shrink singular values to varying extents, our method precisely restores any singular value, thus better preserving the properties of the matrix. For visual comprehension, we exhibit the manifolds of different norm spheres with respect to a 2×2 matrix in Figure 1.

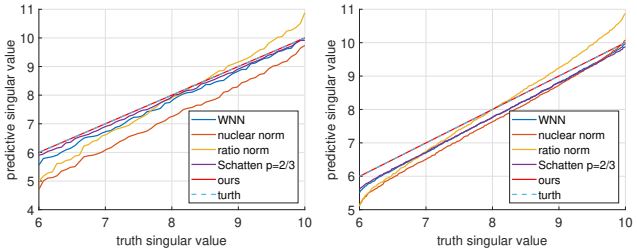


Figure 2: Comparison of different methods on the restoration of singular values of varying magnitudes using simulated data. A network depth of 2 is employed, utilizing matrix dimensions of 100×100 (left) and 400×400 (right). To highlight the distinct characteristics of different methods, matrices of higher rank are used, specifically ranks of 60 and 240, with singular values ranging from 6 to 10 and the observation data ratio of 90%.

Self-Paced Weighted Nuclear Norm. Building on the aforementioned analysis, we propose a novel regularization method, namely the Self-Paced Weighted Nuclear Norm (SPWNN), which can impose more effective low-rank constraints on the network. Here, the definition of the self-paced regularizer of SPWNN is given as follows, and for ease of notation, $r_i(M) := w_i \sigma_i / \|M\|_F$.

Definition 1. *The function $g(c_i, \lambda)$ is referred to as the self-paced regularizer of SPWNN, satisfying the conditions:*

1. *$g(c_i, \lambda)$ is a convex function, $c_i \in [0, 1]$.*
2. *The self-paced learning weight $c_i(r_i(M), \lambda)$ is monotonically decreasing with respect to r_i, and $\lim_{r_i \to \infty} c_i(r_i(M), \lambda) = 0$, $\lim_{r_i \to 0} c_i(r_i(M), \lambda) = 1$.*
3. *The self-paced learning weight $c_i(r_i(M), \lambda)$ is monotonically increasing with respect to λ, and $\lim_{\lambda \to 0} c_i(r_i(M), \lambda) = 0$, $\lim_{\lambda \to \infty} c_i(r_i(M), \lambda) \leq 1$.*

where

$$c_i(r_i, \lambda) = \arg\min_{c_i} \; c_i r_i + g(c_i, \lambda) . \tag{1}$$

The specific form of SPWNN is as follows:

$$R(M) = \sum_{i=1}^{n} c_i\left(r_i(M), \lambda\right) r_i(M) + g(c_i(r_i(M), \lambda), \lambda)$$

$$c_i\left(r_i(M), \lambda\right) = \begin{cases} 1, & \text{if } l_i < \lambda \\ 0, & \text{if } l_i \geq \lambda \end{cases} ,$$

where $\lambda > 0$ represents the growth factor of self-paced regularization. σ_i represents the i-th singular value arranged in descending order. w_i denotes the weight of the singular value σ_i, where $w_i = 1/{\sigma_i}^p$ and $p \in (0, 1)$. The term $c_i(r_i\left(M\right), \lambda)$ indicates the weight of self-paced learning. The function $g\left(c_i(r_i\left(M\right), \lambda), \lambda\right) = -\lambda \sum_{i=1}^{n} c_i(r_i\left(M\right), \lambda)$ acts as the self-paced regularizer.

Specifically, guided by the prior of progressing from easier to more challenging tasks from self-paced learning, we gradually incorporate smaller singular values into the training process to restore the characteristics of the matrix M. Since the larger singular values often contain the primary information of a matrix, we prioritize the restoration of larger singular values before moving on to the smaller ones. Meanwhile, we employ the weighted nuclear norm that applies different weight constraints based on the magnitude of the singular values, offering flexible control over the importance of each direction. In practical applications, this allows for assigning greater weight to factors representing user preferences. The normalization by $\|M\|_F$ ensures that the importance of each direction in the matrix is dependent not on the absolute magnitude of the singular values but on their relative magnitude.

Upon incorporating the penalty term, the final optimization target is expressed as follows:

$$\min_{M} \tilde{\mathcal{L}}(M) := \left\| \mathrm{P}_{\Omega}(\hat{M}) - \mathrm{P}_{\Omega}(M) \right\|_F^2 + \mu R(M), \quad (2)$$

where the penalty term $R(M)$ is our SPWNN and $\mu > 0$ is the balance parameter.

4 Theoretical Analysis

Efficiency Calibration of Implicit Regularization. We analyze the trajectory of singular values and the matrix M to show how the proposed SPWNN calibrates the implicit regularization. The following theorem elucidates the changes in singular values and matrix trajectories without the inclusion of any explicit regularizers:

Lemma 1 (Theorem 1 from [Zhao, 2022])**.** *Under the assumptions specified in [Arora* et al.*, 2019], without any explicit regularization, for a depth greater than 2 and when using the Adam optimizer, the trajectory of singular values and the matrix can be described as follows:*

$$vec(\dot{M}) = -Q_{M,G} vec(\nabla \mathcal{L}(M))$$
$$\dot{\sigma}_i = -vec(v_i u_i^T)^T Q_{M,G} vec(\nabla \mathcal{L}(M)),$$

where

$$Q_{M,G} = \sum_{j}^{N} ((MM^T)^{\frac{N-j}{N}} \otimes (M^T M)^{\frac{j-1}{N}}) G_j. \quad (3)$$

In the above, $Q_{M,G}$ is positive semi-definite, $\dot{M}$ denotes the derivative of M with respect to time t, and $\dot{\sigma}$ represents the derivative of the singular values σ with respect to time t. The Kronecker product is symbolized by $\otimes$, j indexes the network layers, and vec denotes the vectorization of a matrix by column order. $G_j = diag(vec(S_j))$ defines a matrix with the elements of $vec(S_j)$ on the diagonal, and zeros elsewhere. Here, S_j is a matrix with values $(\nabla_{M_j} \mathcal{L}(M)^2 + s_j^2)^{-1/2}$, where $\nabla_{M_j} \mathcal{L}(M) = \partial \mathcal{L}(M) / \partial M_j$ represents the derivative of the loss function with respect to the parameters of the j-th layer, $s_j^2 = var(\nabla_{M_j} \mathcal{L}(M))$, u_i and v_i respectively represent the left and right singular value vectors, corresponding to the i-th singular value σ_i of the matrix M.

The function $Q_{M,G}$ is the result of the combined effect of the Adam optimizer and the depth. When the depth is greater than or equal to 2, $Q_{M,G}$ acts as a preprocessor that facilitates the separation of large and small singular values, thereby accelerating network convergence and promoting low-rank representation within the network. Based on the expression (3) for $Q_{M,G}$, It is evident that the magnitude of $Q_{M,G}$ is related to the depth N, indicating that this implicit regularization increases with the depth, showcasing the effect of implicit regularization in the network. Comparing the trajectories of gradient singular values with matrix changes [Arora *et al.*, 2018b], we observe that this preprocessor benefits from an additional term G_j. The Adam optimizer utilizes information on gradient variance and past momentum to facilitate a fast and precise convergence of the network during the optimization process.

For the convenience, let $C = [c_1, c_2, \ldots, c_n]^T$ and $W = [w_1, w_2, \ldots, w_n]^T \in R^n$. For the matrix M, we perform singular value decomposition as $M = USV^T$. The following theorem describes the trajectories of singular values and matrix changes:

Theorem 1. *In the case of incorporating SPWNN and using the Adam optimizer, the trajectories of singular values and matrix changes are as follows:*

$$vec(\dot{M}) = -Q_{M,G} vec(\nabla \mathcal{L}(M) + \frac{\lambda}{\|M\|_F^2} U\hat{S}V^{\mathrm{T}})$$
$$\dot{\sigma}_i = -vec(v_i u_i^T)^T Q_{M,G} vec(\nabla \mathcal{L}(M) + \frac{\lambda}{\|M\|_F^2} U\hat{S}V^{\mathrm{T}}),$$

where

$$\hat{S} = CW^T \|M\|_F - \frac{S}{\|M\|_F} tr(WC^T S).$$

We give complete proof in Appendix A. Comparing the trajectory expression in Lemma 1, the inclusion of an additional term $U\hat{S}V^T$ offers supplementary acceleration toward low rank for matrix M during the training process.

When the depth is set to 1, $Q_{M,G}$ degenerates to G, resulting in the loss of the acceleration effect on the network. Despite this loss, the additional penalty term still serves its purpose in separating large and small singular values, allowing for convergence even with a depth of 1. Our experiments also show that without explicit regularization, an LNN cannot converge. However, the introduction of explicit regularization leads to significant improvement in the results, highlighting its dominant role in this scenario.

When the depth is within an appropriate range, the effect of $Q_{M,G}$ allows for convergence with just a LNN. Additionally, the convergence speed increases as the depth increases, indicating the start of implicit regularization. When explicit regularization is incorporated, it interacts with implicit regularization, resulting in faster convergence and stronger low-rank constraints compared to the original scenario. The penalty term provided by $Q_{M,G}$ also enhances the acceleration effect as the depth increases. By observing the trajectory after adding the penalty term, it can be seen that $\hat{S}$ acts as a scaling factor on the spectrum of S, affecting the entire evolution trajectory from a frequency domain perspective. Looking at

the singular values, the self-paced regularization guides the training process by selectively incorporating singular values based on their magnitudes, allowing for less noise interference and convergence in the correct direction. Furthermore, by assigning different weights to each singular value, more flexible constraints can be imposed on the matrix. The introduction of $\|M\|_F$ modifies the effect of singular values on training by considering their relative magnitudes instead of just their absolute magnitudes. This strengthens the impact of larger singular values on optimization while reducing the influence of smaller singular values, further promoting low-rank constraints. At this point, the implicit and explicit regularization work together to promote low-rank properties in the network, achieving a balanced state.

When the depth is deep enough and the rank is high, we have noticed a trend in experiments where using only a LNN yields better results than incorporating a regularization term. This suggests that the implicit regularization is strong enough to serve as an effective regularizer, achieving satisfactory low-rank constraints without the need for extra regularization terms. This highlights the fact that explicit and implicit regularization do not always have a mutually beneficial impact. In this case, the implicit regularization of the network takes on a dominant role.

Convergence analysis. We will analyze the convergence of our method from the view of Majorization Minimization (MM) Algorithm [Sun *et al.*, 2017]. For the convenience, let's define $f\left(r\left(M\right)\right) := \int_0^{r(M)} cdr\left(M\right)$. First, we present the following theorem to establish an upper bound for the function $f\left(r\left(M\right)\right)$.

Theorem 2. *Given a parameter M^*, the function $f\left(r\left(M\right)\right)$ and the self-paced weight c have the following relationship:*

$$f\big(r(M)\big) \leq f(r(M^*)\big) + c(r(M) - r(M^*)\big).$$

The proof is given in Appendix A. Let's assume that the optimization objective has reached the k-th iteration and use M^k to represent the parameters at the k-th iteration.

Majorization: The objective of this stage is to obtain an upper bound for the optimization objective. According to Theorem 1, we can derive the upper bound for $f\left(r_i\left(M\right)\right)$:

$$h^i(M|M^k) := f(r_i(M^k)) + c_i(r_i(M) - r_i(M^k)). \quad (4)$$

Furthermore, it can be established that:

$$\sum_{i=1}^{n} f(r_i(M)) \leq \sum_{i=1}^{n} h^i(M|M^k). \quad (5)$$

The computation of $h^i\left(M|M^k\right)$ can be obtained at each iteration using equations (4) and (1).

Minimization: The objective of this stage is to optimize the upper bound function $h\left(M|M^k\right)$ for $f\left(r\left(M\right)\right)$ and obtain the optimized parameters. Based on (2), (4) and (5), we can derive the optimization objective as follows:

$$\begin{aligned} M^{k+1} &= \arg\min_M \mathcal{L}(M) + \sum_{i=1}^{n} f\left(r_i(M^k)\right) \\ &\quad + (r_i(M) - r_i(M^k))c_i. \\ &= \arg\min_M \mathcal{L}(M) + \sum_{i=1}^{n} c_i r_i(M). \end{aligned} \quad (6)$$

Observing that this optimization objective (6) is consistent with the original objective (2), it indicates that our proposed algorithm is equivalent to the MM Algorithm. This proves that our algorithm has reliable convergence properties.

Complexity Analysis. In this section, we will analyze the time complexity of the algorithm. To simplify our discussion, we will assume that the matrix completion uses a matrix with dimensions of $n \times n$ and that the depth of the network is denoted as h.

For the LNN, each layer has n inputs and n outputs. With a depth of h, the corresponding time complexity is $O(hn^2)$. Regarding the penalty term $R(M)$, during each training process, the matrix needs to undergo singular value decomposition, which has a time complexity of $O(2n^3)$. Computing the Frobenius norm of the matrix requires a time complexity of $O(n^2)$. Similarly, calculating the self-paced regularizer $g(c_i(r_i(M), \lambda), \lambda)$ and the weight w_i corresponding to the singular values both have a time complexity of $O(n)$. Therefore, the overall time complexity of the algorithm is $O(hn^2 + 2n^3 + n^2 + 2n)$. Since the dimension of the matrix is typically much larger than the depth of the network, the dominant time complexity is $O(n^3)$. Similarly, for the ratio penalty term, since it also involves computing the singular values, its corresponding time complexity is also $O(n^3)$.

Through the analysis of time complexity, it can be observed that the SPWNN algorithm improves network performance without significantly increasing computational complexity compared to other methods, highlighting the superiority of our algorithm.

5 Experiments

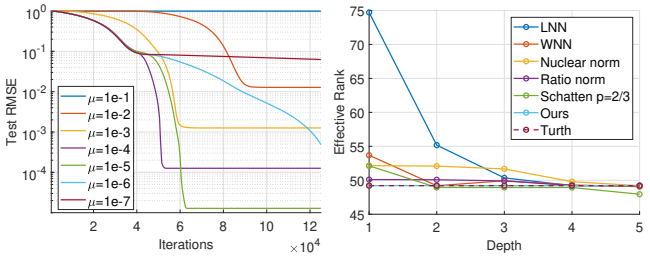
In this section, we will compare our method with other methods listed in Table 1, using both synthetic and real datasets. The evaluation criterion will be the root mean square error (RMSE) of the test data, with a lower RMSE indicating better results. All experiments will be conducted using the Adam optimizer. Furthermore, we will utilize the effective rank (denoted by $e\_rank(\cdot)$) [Arora *et al.*, 2019] to track and measure the rank of the matrix. Specifically, $e\_rank(M) = exp\{H(a_1, \ldots, a_n)\}$, where $a_i = \sigma_i / ||\sigma||_1$, H is the Shannon entropy, and $\|\cdot\|_1$ denotes the L_1 norm. More experiments are found in the supplementary materials.

5.1 Implementations

The results of the experiments are obtained by taking the average of multiple runs. The stopping condition for the experiment is when the iteration reaches 4×10^5 or when the test results remain unchanged after 2×10^4 iterations. For the experimental parameter settings, the networks used in this study do not include bias terms. The network initialization method is the same as [Arora *et al.*, 2019], where the weights are initialized to 10^{-3}. In the case of simulated data with a depth greater than or equal to 2, the learning rate is set to 10^{-5}, while for a depth of 1, the learning rate is set to 10^{-3}. The penalty coefficient μ is set to 10^{-5}, and its impact on the experiments is depicted in Figure 3a. In the case of real data with a depth greater than or equal to 2, the learning rate is set to 10^{-4}, and for a depth of 1, the learning rate is set to

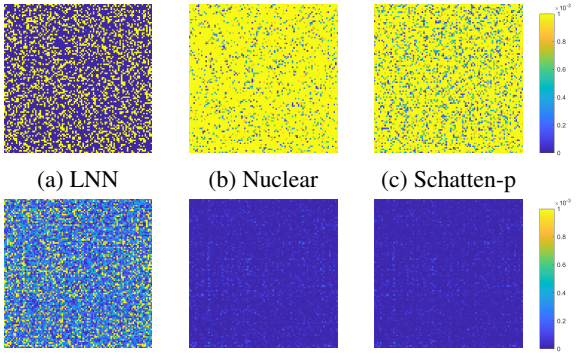
		5	10	20	30	40	50	60
Depth=1	LNN	1.00e+00	1.05e+00	1.00e+00	1.01e+00	9.82e-01	9.91e-01	9.87e-01
	Schatten-p	5.23e-01	5.79e-01	6.84e-01	7.43e-01	7.87e-01	8.33e-01	8.34e-01
	Ratio	2.52e-06	5.15e-06	1.27e-05	2.99e-05	5.72e-05	1.66e-04	1.61e-01
	Weight	3.04e-02	2.01e-02	1.86e-02	3.74e-02	1.67e-01	3.57e-01	4.60e-01
	Nuclear	1.33e-03	4.85e-03	1.10e-02	2.50e-02	1.64e-02	1.69e-02	1.12e-01
	Ours	**2.21e-06**	**4.11e-06**	**1.03e-05**	**2.49e-05**	**4.88e-05**	**1.43e-04**	**5.49e-04**
Depth=2	LNN	3.00e-02	4.88e-02	8.36e-02	1.66e-01	2.56e-01	3.83e-01	5.65e-01
	Schatten-p	3.06e-04	7.21e-04	2.06e-03	4.14e-03	9.60e-03	2.12e-02	7.10e-02
	Ratio	2.98e-06	4.13e-06	1.57e-05	2.87e-05	6.32e-05	1.49e-04	1.69e-01
	Weighted	5.12e-05	9.15e-05	1.98e-04	3.40e-04	7.11e-04	1.48e-03	5.45e-03
	Nuclear	1.36e-03	2.24e-03	4.47e-03	7.23e-03	1.49e-02	9.77e-02	9.77e-02
	Ours	**2.36e-06**	**3.29e-06**	**1.27e-05**	**2.40e-05**	**5.40e-05**	**1.30e-04**	**5.37e-04**
Depth=5	LNN	4.15e-05	4.92e-05	5.12e-05	6.03e-05	6.46e-05	**9.79e-05**	**3.00e-04**
	Schatten-p	2.98e-04	7.20e-04	2.06e-03	4.13e-03	9.60e-03	2.12e-02	7.37e-01
	Ratio	3.58e-05	2.02e-05	3.09e-05	4.20e-05	8.25e-05	1.64e-04	9.11e-04
	Weighted	4.91e-04	9.07e-04	1.98e-03	3.39e-03	7.03e-03	1.43e-02	4.60e-02
	Nuclear	1.36e-04	2.26e-04	4.59e-04	7.34e-04	1.48e-03	3.03e-03	7.74e-02
	Ours	**1.68e-05**	**1.79e-05**	**2.05e-05**	**3.04e-05**	**5.88e-05**	1.33e-04	5.50e-04

Table 2: The outcomes of different methods at varying depths on simulated data, utilizing matrix dimensions of 100×100, with rank ranging from 5-60, and an observation data proportion of 90%.



(a) (b)

Figure 3: (a) depicts the impact of hyperparameters μ at a depth of 2; (b) shows the variation of effective rank restoration under different depths and methods. The experiments utilized matrices of 100×100 dimensions with a 90% observation ratio.



(a) LNN (b) Nuclear (c) Schatten-p

(d) Weighted (e) Ratio (f) Ours

Figure 4: Performance comparison of various methods on simulated data using a 2-layer network. The matrix dimension employed is 100×100, with the ratio of observed data at 70%, and a rank of 30.

10^{-2}. Due to significant differences in data structures, the penalty coefficient μ varies with different real data. The initial value of the threshold for self-regularization is set to 1, and the growth factor for the threshold is set to 1.08.

5.2 Experiments on the Simulated Data

Simulated data are generated using a Gaussian distribution. Specifically, a matrix W of a designated rank is first constructed, followed by computation of WW^T to obtain the true data matrix $\hat{W}$. In the experiments, the chosen matrix dimensions are 100×100, with tests carried out at various depths and ranks. The specific results are provided in Table 2. Furthermore, robustness tests were conducted on matrices of different dimensions and diverse proportions of observed data, with detailed results presented in Appendix B.

The experiments conducted on synthetic data consistently demonstrate the same outcome: our penalty term shows a significant improvement over previous methods at any depth. This results in better restoration of the intrinsic properties of the matrices, as shown in Figure 4. Additionally, we visualize the experimental outcomes in Figure 5, where the data restored by our approach more closely matches the distribution of the original data. Furthermore, at a depth of 1, the network only converges after the inclusion of the penalty term. In the case of a depth of 2, the network can still converge to satisfactory results even without explicit regularization, due to the effect of implicit regularization. However, the performance of the network is significantly enhanced with the addition of explicit regularization. These findings suggest that at an appropriate depth, the interplay between explicit and implicit regularization leads to a stronger low-rank constraint, thus confirming the validity of our theory.

However, as the rank increases and the depth grows, even with explicit regularization, the performance of matrix completion may not surpass that of a LNN. This phenomenon varies depending on the required depth, which is influenced

		Depth=1			Depth=2			Depth=5		
		ML100K	ML1M	FilmTrust	ML100K	ML1M	FilmTrust	ML100K	ML1M	FilmTrust
0.7	LNN	3.695	3.748	3.149	0.933	0.923	1.022	**0.943**	**0.885**	**1.026**
	Schatten-p	0.986	0.921	1.052	0.926	0.915	1.033	0.959	0.915	1.029
	Ratio	0.940	0.868	1.018	0.922	0.850	1.021	0.945	0.886	1.057
	Weighted	0.947	0.890	1.087	0.924	0.858	1.043	0.946	0.913	1.033
	Nuclear	0.992	0.903	1.124	0.925	0.851	1.053	0.944	0.920	1.036
	Ours	**0.937**	**0.862**	**1.011**	**0.911**	**0.845**	**1.022**	0.944	0.885	1.029
0.9	LNN	3.720	3.740	3.120	0.916	0.842	0.990	**0.925**	**0.874**	**0.963**
	Schatten-p	0.972	0.910	1.036	0.917	0.843	0.975	0.947	0.911	0.973
	Ratio	0.915	0.857	0.942	0.906	0.837	0.958	0.926	0.883	0.967
	Weighted	0.932	0.869	1.031	0.910	0.846	0.970	0.926	0.882	0.970
	Nuclear	0.963	0.884	1.097	0.909	0.849	0.991	0.928	0.911	0.974
	Ours	**0.910**	**0.852**	**0.935**	**0.895**	**0.831**	**0.951**	0.925	0.877	0.964

Table 3: The experiments conducted on the ML100K, ML1M, and FilmTrust datasets utilized observed data ratios of 70% and 90%.

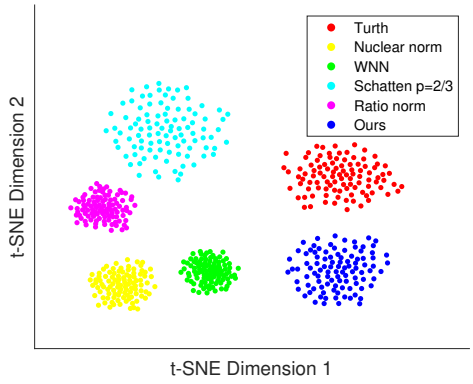


Figure 5: Comparison of t-SNE Visualization Methods at a Network Depth of 2, with an Observation Data Proportion of 90%. The matrix dimensions employed is 100×100; for ease of comparison, we present the varied data in a dispersed format.

by the structure of the data set. As shown in Figure 3b, the change in effective rank indicates that as the depth increases, the implicit regularization becomes more effective in constraining the rank, eventually achieving results comparable to or even better than those with explicit regularization. The figure also demonstrates that our proposed method maintains a consistently strong low-rank constraint at any depth.

The phenomena described above suggest that at shallower depths, the implicit regularization provided by the network itself may not serve as an effective regularizer, necessitating explicit regularization for constraint. However, as the depth increases, the implicit regularization of the network becomes stronger and can lead to excellent results without the need for explicit regularization. However, a deeper network structure will lead to higher computational complexity, which is obviously impractical in the real world. Thus, compared with those methods that only have implicit regularization, our method may be a better choice for practical applications.

5.3 Experiments on the Real-World Data

The real datasets employed are ML100K [Harper and Konstan, 2016], ML1M, and FilmTrust [Guo *et al.*, 2013]. Unlike simulated data, real data consists of discrete values, and they tend to have a high rank close to full rank. Specifically, the ML100K dataset contains 100,000 ratings from 943 users for 1,682 movies, while the ML1M dataset is composed of ratings from 6,400 users for 3,900 movies, totaling one million ratings. In the MovieLens series, rating values range from 1 to 5 and are integer values. The FilmTrust dataset includes 35,497 ratings from 1,508 users for 2,071 movies, with a rating scale of 0.5 to 4, and increments in multiples of 0.5. We conducted similar experiments on the real datasets. The specific results are presented in Table 3.

The experiments demonstrate that the results are consistent for both real and simulated data sets. However, at a depth of 1, the explicit regularization is necessary for convergence. On the other hand, when the depth is optimal, explicit regularization significantly improves network performance. Interestingly, when the depth is large enough, the results with explicit regularization and those with only an LNN show minimal differences, with a slight advantage for the latter. While there may be slight deviations from the simulated data results, this also highlights the power of the network's implicit regularization at sufficient depths and with higher ranks, rendering explicit regularization unnecessary. Additionally, our introduced regularization term remains superior to the current best regularizer across all depths.

6 Conclusion

This paper introduces the SPWNN, which not only enables more flexible constraints on singular values compared to previous methods but also exhibits stronger robustness to noise. We have validated its effectiveness and convergence through experiments and theoretical analysis. Moreover, we have investigated the relationship between explicit and implicit regularization as a function of depth. Our results show that at shallow depths, the addition of explicit regularization significantly promotes network convergence. However, at greater depths with higher rank, the inclusion of explicit regularization does not contribute to performance enhancement.

Acknowledgments

This work was supported in part by the National Science Fund of China under Grant 62276135 and Grant 62361166670.

References

[Arora *et al.*, 2018a] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *ICLR*. OpenReview.net, 2018.

[Arora *et al.*, 2018b] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *ICML*, volume 80, pages 244–253. PMLR, 2018.

[Arora *et al.*, 2019] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *NIPS*, 32:7411–7422, 2019.

[Belkin *et al.*, 2019] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[Cai *et al.*, 2010] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[Candes and Recht, 2012] Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[Gong *et al.*, 2016] Tieliang Gong, Qian Zhao, Deyu Meng, and Zongben Xu. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data and Information Analytics*, 1(1):111–127, 2016.

[Gu *et al.*, 2014] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[Gunasekar *et al.*, 2017] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *NIPS*, 30:6151–6159, 2017.

[Guo *et al.*, 2013] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *IJCAI*, pages 2619–2625, 2013.

[Hardt *et al.*, 2016] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*, volume 48, pages 1225–1234. PMLR, 2016.

[Harper and Konstan, 2016] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.

[Hu *et al.*, 2012] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2117–2130, 2012.

[Jiang *et al.*, 2014] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 547–556, 2014.

[Jiang *et al.*, 2015] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 29, pages 2694–2700, 2015.

[Kim *et al.*, 2015] Eunwoo Kim, Minsik Lee, and Songhwai Oh. Elastic-net regularization of singular values for robust subspace learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 915–923, 2015.

[Krizhevsky *et al.*, 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[Kumar *et al.*, 2010] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *NIPS*, 23:1189–1197, 2010.

[Li *et al.*, 2021] Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *ICLR*. OpenReview.net, 2021.

[Li *et al.*, 2023] Jiangyuan Li, Thanh V. Nguyen, Chinmay Hegde, and Raymond K. W. Wong. Implicit regularization for group sparsity. In *ICLR*, 2023.

[Meng *et al.*, 2017] Deyu Meng, Qian Zhao, and Lu Jiang. A theoretical understanding of self-paced learning. *Information Sciences*, 414:319–328, 2017.

[Neyshabur *et al.*, 2015] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *ICLR*, 2015.

[Neyshabur *et al.*, 2017] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *NIPS*, 30:5947–5956, 2017.

[Nie *et al.*, 2012] Feiping Nie, Heng Huang, and Chris H. Q. Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, pages 655–661. AAAI Press, 2012.

[Orvieto *et al.*, 2022] Antonio Orvieto, Hans Kersting, Frank Proske, Francis Bach, and Aurelien Lucchi. Anti-correlated noise injection for improved generalization. In *ICML*, volume 162, pages 17094–17116. PMLR, 2022.

[Peng *et al.*, 2015] Chong Peng, Zhao Kang, Huiqing Li, and Qiang Cheng. Subspace clustering using log-determinant rank approximation. In *Proceedings of the 21th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 925–934, 2015.

[Raghu *et al.*, 2017] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *ICLR*, pages 2847–2854. PMLR, 2017.

[Razin and Cohen, 2020] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *NIPS*, 33:21174–21187, 2020.

[Smith *et al.*, 2021] Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *ICLR*. OpenReview.net, 2021.

[Sun *et al.*, 2017] Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Trans. Signal Process.*, 65(3):794–816, 2017.

[Vardi, 2023] Gal Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66(6):86–93, 2023.

[Wang *et al.*, 2015] Zheng Wang, Ming-Jun Lai, Zhaosong Lu, Wei Fan, Hasan Davulcu, and Jieping Ye. Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM Journal on Scientific Computing*, 37(1):A488–A514, 2015.

[Wang *et al.*, 2022] Bohan Wang, Qi Meng, Huishuai Zhang, Ruoyu Sun, Wei Chen, Zhi-Ming Ma, and Tie-Yan Liu. Does momentum change the implicit regularization on separable data? *NIPS*, 35:26764–26776, 2022.

[Williams *et al.*, 2019] Francis Williams, Matthew Trager, Daniele Panozzo, Claudio Silva, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. *NIPS*, 32:8376–8385, 2019.

[Wu and Su, 2023] Lei Wu and Weijie J. Su. The implicit regularization of dynamical stability in stochastic gradient descent. In *ICML*, volume 202, pages 37656–37684, 2023.

[Wu *et al.*, 2022] Lei Wu, Mingze Wang, and Weijie Su. The alignment property of sgd noise and how it helps select flat minima: A stability analysis. *NIPS*, 35:4680–4693, 2022.

[Yan *et al.*, 2013] Ming Yan, Yi Yang, and Stanley Osher. Exact low-rank matrix completion from sparsely corrupted entries via adaptive outlier pursuit. *Journal of Scientific Computing*, 56:433–449, 2013.

[Yang *et al.*, 2018] Shuyuan Yang, Zhixi Feng, Min Wang, and Kai Zhang. Self-paced learning-based probability subspace projection for hyperspectral image classification. *IEEE transactions on neural networks and learning systems*, 30(2):630–635, 2018.

[Zaremba *et al.*, 2014] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[Zhang *et al.*, 2017] Dingwen Zhang, Le Yang, Deyu Meng, Dong Xu, and Junwei Han. Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4429–4437, 2017.

[Zhang *et al.*, 2019] Dingwen Zhang, Junwei Han, Long Zhao, and Deyu Meng. Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework. *IJCV*, 127:363–380, 2019.

[Zhang *et al.*, 2021] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[Zhao *et al.*, 2015] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander Hauptmann. Self-paced learning for matrix factorization. In *AAAI*, volume 29, pages 3196–3202, 2015.

[Zhao, 2022] Dan Zhao. Combining explicit and implicit regularization for efficient learning in deep networks. *NIPS*, 35:3024–3038, 2022.

[Zhou *et al.*, 2020] Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. Curriculum learning by dynamic instance hardness. *NIPS*, 33:8602–8613, 2020.