Invariant Low-Dimensional Subspaces in Gradient Descent for Learning Deep Matrix Factorizations

Anonymous Author(s)
Affiliation
Address
email

Abstract

An extensively studied phenomenon of the past few years in training deep networks is the implicit bias of gradient descent towards parsimonious solutions. In this work, we further investigate this phenomenon by narrowing our focus to deep matrix factorization, where we reveal surprising low-dimensional structures in the learning dynamics when the target matrix is low-rank. Specifically, we show that the evolution of gradient descent starting from arbitrary orthogonal initialization only affects a minimal portion of singular vector spaces across all weight matrices. In other words, the learning process happens only within a small invariant subspace of each weight matrix, despite the fact that all parameters are updated throughout training. From this, we provide rigorous justification for low-rank training in a specific, yet practical setting. In particular, we demonstrate that we can construct compressed factorizations that are equivalent to full-width, deep factorizations throughout training for solving low-rank matrix completion problems efficiently.

1 Introduction

In recent years, deep learning has demonstrated remarkable success across a wide range of applications [1]. Many recent works attempt to explain the exceptional generalization capabilities of deep networks by studying the implicit bias of gradient-based methods, showing that deep networks trained with such algorithms tend to learn simple functions [2–6]. Similarly, it has been shown that gradient descent induces max-margin [6, 7] or low-rank solutions [8–12] in deep networks, to name a few.

In another vein, recent work has explored the increasingly important problem of training deep networks more efficiently via *low-rank training* [13–17], where the number of trainable parameters is effectively reduced by replacing the original network weights with low-rank factorizations. While such methods have shown promising empirical results for reducing training time, theoretical justifications for these approaches remain deficient. The aforementioned works on implicit bias characterize low-rank structure in the limit of gradient descent – they do not address whether the trajectory of the original overparameterized network (along with its generalization/convergence properties) is achievable via low-rank factorization *from initialization* throughout training, which is what low-rank training necessitates.

Contributions. In this work, we draw theoretical connections between the implicit bias of gradient descent in deep networks and the practice of low-rank training. Utilizing deep matrix factorizations as a testbed (commonly assumed for analyzing the complex optimization dynamics of deep networks [10, 18–20]) we demonstrate that for low-rank data, all weight matrices are only updated within a low-dimensional subspace that is *invariant throughout training*, which can be determined from their arbitrary orthogonal initialization. To our knowledge, this is the first work identifying such invariant structures in gradient descent dynamics from random initialization. From this, we show that we can construct a compressed, low-rank factorization that is nearly equivalent to the original

Submitted to the Mathematics of Modern Machine Learning Workshop at NeurIPS 2023. Do not distribute.

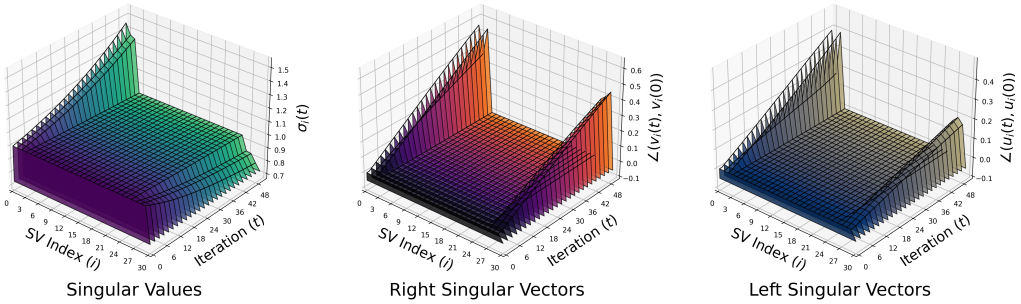

Figure 1: **Evolution of SVD of weight matrices.** We visualize the SVD dynamics of the first layer weight matrix of an $L = 3$ layer deep matrix factorization with $d = 30$, $r = 3$, $\sigma_l = 1$ throughout GD without weight decay. *Left*: Magnitude of the i-th singular value $\sigma_i(t)$ at iteration t. *Middle*: Angle $\angle(\boldsymbol{v}_i(t), \boldsymbol{v}_i(0))$ between the i-th right singular vector at iteration t and initialization. *Right*: Angle $\angle(\boldsymbol{u}_i(t), \boldsymbol{u}_i(0))$ between the i-th left singular vector at iteration t and initialization.

overparameterized network, thereby providing some rigorous foundations for low-rank training. Although deep matrix factorizations are mostly of theoretical interest, they are adopted for low-rank matrix sensing problems - therefore, we demonstrate that our theory can be applied (with slight modifications) towards accelerating practical low-rank deep matrix completion problems.

2 Analysis

Setup. We study the training dynamics of L-layer deep matrix factorizations $f(\boldsymbol{\Theta})$ given by

$$f(\boldsymbol{\Theta}) := \boldsymbol{W}_L \boldsymbol{W}_{L-1} \cdots \boldsymbol{W}_2 \boldsymbol{W}_1$$

where $\boldsymbol{\Theta} = (\boldsymbol{W}_l)_{l=1}^L$ are the parameters or weights with $\boldsymbol{W}_l \in \mathbf{R}^{d_l \times d_{l-1}}$ for $l \in [L]$. For a given target matrix $\boldsymbol{\Phi} \in \mathbf{R}^{d \times d}$, we learn parameters $\boldsymbol{\Theta}$ with $d_0 = d_1 = \cdots = d_L = d$ by minimizing the square loss

$$\ell(\boldsymbol{\Theta}) = \frac{1}{2} \|f(\boldsymbol{\Theta}) - \boldsymbol{\Phi}\|_F^2 \tag{1}$$

via gradient descent (GD) from scaled *orthogonal* initialization, i.e., we initialize parameters $\boldsymbol{\Theta}(0)$ such that all singular values of $\boldsymbol{W}_l(0)$ are equal to some $\sigma_l > 0$ for each $l \in [L]$. Then, we update all weights for $t = 0, 1, 2, \ldots$ as

$$\boldsymbol{W}_l(t+1) = (1 - \eta\lambda)\boldsymbol{W}_l(t) - \eta \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)),\ l \in [L] \tag{2}$$

where $\lambda \geq 0$ is an optional weight decay parameter and $\eta > 0$ is the learning rate.

Main Result. Under the setting described above, we show learning only occurs within an invariant low-dimensional subspace of the weight matrices, provided that the target matrix $\boldsymbol{\Phi}$ is low-rank.

Theorem 1. *Suppose $\boldsymbol{\Phi} \in \mathbf{R}^{d \times d}$ is at most rank r where $m := d - 2r > 0$. Then there exist orthogonal matrices $(\boldsymbol{U}_l)_{l=1}^L \subset \mathbf{R}^{d \times d}$ and $(\boldsymbol{V}_l)_{l=1}^L \subset \mathbf{R}^{d \times d}$ satisfying $\boldsymbol{V}_{l+1} = \boldsymbol{U}_l$ for $l \in [L-1]$, such that $\boldsymbol{W}_l(t)$ admits the decomposition*

$$\boldsymbol{W}_l(t) = \boldsymbol{U}_l \begin{bmatrix} \widetilde{\boldsymbol{W}}_l(t) & \mathbf{0} \\ \mathbf{0} & \rho_l(t) \boldsymbol{I}_m \end{bmatrix} \boldsymbol{V}_l^\top \tag{3}$$

for all $l \in [L]$ and $t \geq 0$, where $\widetilde{\boldsymbol{W}}_l(t) \in \mathbf{R}^{2r \times 2r}$ with $\boldsymbol{W}_l(0) = \sigma_l \boldsymbol{I}_{2r}$, and

$$\rho_l(t) = \rho_l(t-1) \cdot (1 - \eta\lambda - \eta \cdot \prod_{k \neq l} \rho_k^2(t-1)) \tag{4}$$

for all $l \in [L]$ and $t \geq 1$ with $\rho_l(0) = \sigma_l$.

We defer the proof of Theorem 1 to Appendix A.1. In the following, we discuss several implications of our result and its relationship to previous work.

- **SVD dynamics of weight matrices.** The decomposition (3) implies that $\boldsymbol{W}_l(t)$ has m identical singular values that follow the updates given in (4), whose corresponding singular vectors are stationary from initialization throughout GD – this is portrayed in Figure 1. By this, we can decompose the total space $\mathbf{R}^d$ into two invariant singular subspaces: a $2r$-dimensional space within which learning takes place, and an m-dimensional space corresponding to repeated singular values.

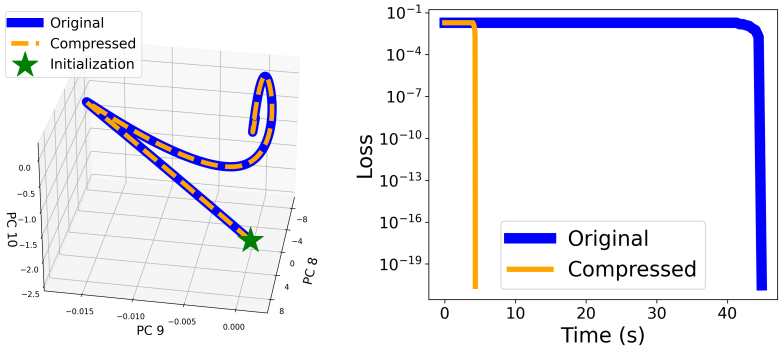

Figure 2: **Network compression for deep matrix factorization.** Comparison of trajectories for optimizing the original problem (1) vs. the compressed problem (6) with $L = 3$, $d = 1000$, $\widehat{r} = r = 5$, and $\sigma_l = 10^{-3}$. *Left*: Principal components of end-to-end GD trajectories. *Right*: Training loss vs. wall-time comparison.

- **Low-rank bias.** From (4), we see that the GD trajectory either remains or tends towards a rank of at most $2r$ when we employ implicit or explicit regularization respectively. Indeed, if we use small initialization $\sigma_l \approx 0$, then the fact that ρ_l is a decreasing sequence implies that $\boldsymbol{W}_l(t)$ can be no more than rank $2r$ throughout the entire trajectory; whereas if $\lambda > 0$, then $\rho_l(t) \to 0$ as $t \to \infty$, which forces $\boldsymbol{W}_l(t)$ towards a solution of rank at most $2r$, regardless of initialization.
- **Comparison to prior arts.** In contrast to existing work that demonstrates the tendency of GD to find low nuclear-norm solutions [9, 11], our result directly shows that GD tends to find low-rank solutions. Moreover, unlike previous work on implicit bias [11, 21–23], we carefully examine the effect of weight decay, which is commonly employed during the training of deep networks. We note that our analysis is distinct from that of [18, 19], where continuous time dynamics are studied with the special (separable) setting $\boldsymbol{W}_{L:1}(0) = \boldsymbol{U}\boldsymbol{V}^\top$ with $\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. In contrast, our result applies to discrete time GD and holds for initialization that is agnostic to the target matrix. We also note that our result is unrelated to balanced initialization (as in [24]), since the σ_l can be arbitrarily different from one another.

Compressed Deep Matrix Factorization. We now show that, as a consequence of Theorem 1, we can run gradient descent on dramatically fewer parameters to achieve a near *identical* end-to-end trajectory to the original (full-width) factorization. More specifically, given an initialization $\boldsymbol{\Theta}(0)$ of the original parameters and an upper bound on the rank $\widehat{r} \geq r$ such that $d - 2\widehat{r} > 0$, we define the *compressed* factorization

$$\widehat{f}(\widehat{\boldsymbol{\Theta}}, \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}) := \boldsymbol{U}_{L,1}\widehat{\boldsymbol{W}}_L\widehat{\boldsymbol{W}}_{L-1}\cdots\widehat{\boldsymbol{W}}_1\boldsymbol{V}_{1,1}^\top \tag{5}$$

where $\widehat{\boldsymbol{\Theta}} = (\widehat{\boldsymbol{W}}_l)_{l=1}^L$ are compressed weights with $\widehat{\boldsymbol{W}}_l \in \mathbf{R}^{2\widehat{r}\times 2\widehat{r}}$ and $\boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1} \in \mathbf{R}^{d\times 2\widehat{r}}$ are the first $2\widehat{r}$ columns of $\boldsymbol{U}_L, \boldsymbol{V}_1 \in \mathbf{R}^{d\times d}$ respectively from Theorem 1 (depends on $\boldsymbol{\Theta}(0)$ and $\boldsymbol{\Phi}$). Then, initializing $\widehat{\boldsymbol{\Theta}}(0)$ such that $\widehat{\boldsymbol{W}}_l(0) = \boldsymbol{U}_{l,1}^\top\boldsymbol{W}_l(0)\boldsymbol{V}_{l,1}$ for all $l \in [L]$ and running gradient descent on the loss

$$\widehat{\ell}(\widehat{\boldsymbol{\Theta}}) = \frac{1}{2}\|\widehat{f}(\widehat{\boldsymbol{\Theta}}, \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}) - \boldsymbol{\Phi}\|_F^2 \tag{6}$$

gives an almost equivalent network in the following sense.

Proposition 1. *For $\widehat{r} \geq r$ such that $\widehat{m} := d - 2\widehat{r} > 0$, running gradient descent on the compressed weights $\widehat{\boldsymbol{\Theta}}$ as described above for the loss* (6) *satisfies*

$$\left\|f(\boldsymbol{\Theta}(t)) - \widehat{f}(\widehat{\boldsymbol{\Theta}}(t), \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1})\right\|_F^2 \leq \widehat{m}\cdot\prod_{l=1}^L \sigma_l^2$$

for all iterates $t = 0, 1, 2, \ldots$.

We defer the proof of Proposition 1 to Appendix A.2. When we start from small initialization ($\sigma_l \approx 0$), Proposition 1 demonstrates that we only need to optimize $4L\cdot\widehat{r}^2$ many parameters as opposed to the original $L\cdot d^2$ number of parameters to achieve an almost identical end-to-end trajectory, see Figure 2 (left). Since it is often the case that $r \leq \widehat{r} \ll d$, this results in an order of magnitude reduction in time to reach an optimal solution compared to the original network, see Figure 2 (right). In the next section, we demonstrate how this idea can be leveraged (with slight modification) to accelerate a more practical problem.

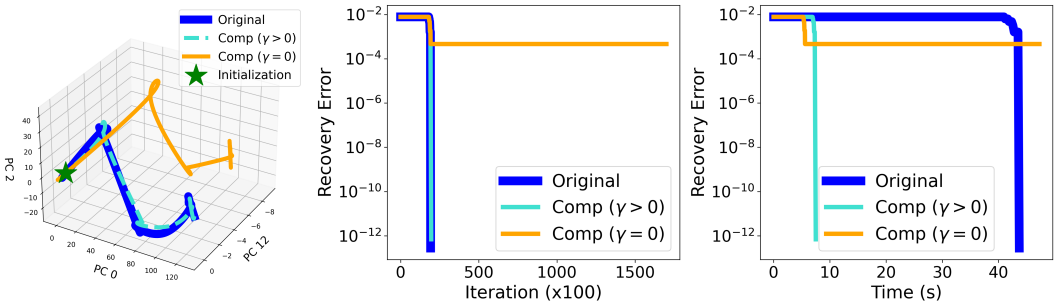

Figure 3: **Network compression for deep matrix completion.** Comparison of trajectories for optimizing the original problem (7) vs. the compressed problem (8) with γ discrepant updates ($\gamma = 0.01$) and ablating γ ($\gamma = 0$) with $L = 3$, $d = 1000$, $r = 5$, $\sigma_l = 10^{-3}$ and 20% of entries observed. *Left*: Principal components of end-to-end trajectories of each factorization. *Middle*: Recovery error vs. iteration comparison. *Right*: Recovery error vs wall-time comparison.

3 Application: Accelerating Deep Low-Rank Matrix Completion

Problem Setup. We consider the low-rank matrix completion problem [25–27] with ground-truth $\mathbf{\Phi} \in \mathbf{R}^{d\times d}$ with rank $r \ll d$, where the goal is to recover $\mathbf{\Phi}$ from only a few number of observations encoded by a mask $\mathbf{\Omega} \in \{0,1\}^{d\times d}$. Adopting a deep matrix factorization approach [11], we minimize the objective

$$\ell_{\text{mc}}(\mathbf{\Theta}) = \frac{1}{2}\|\mathbf{\Omega} \odot (f(\mathbf{\Theta}) - \mathbf{\Phi})\|_F^2 \tag{7}$$

which simplifies to (1) when $\mathbf{\Omega} = \mathbf{1}_d\mathbf{1}_d^\top$ in the full observation case. In practice, the true rank r is not known – instead, we assume to have an upper bound $\widehat{r}$ of the same order as r, i.e., $r \leq \widehat{r} \ll d$.

Compressed Deep Matrix Completion. In the setting described above, it is advantageous to *over-parameterize* along both the depth and width of the factorization, particularly for accelerating GD convergence to well-generalizing solutions – see Appendix B for a more detailed discussion alongside evidence. Nonetheless, the advantages of over-parameterization are hindered by the fact that depth and width incur much higher *per-iteration* costs – for an L-layer factorization of (full)-width d, we require $O(L \cdot d^3)$ multiplications to evaluate gradients, where d is often very large. However, using ideas from the previous section, we can effectively reduce the computation to $O(\widehat{r}^2 \cdot (L\widehat{r} + d))$ multiplications via a compressed factorization that emulates the trajectory of a (full)-width d network, thereby enjoying accelerated GD convergence with heavily reduced per-iteration computational cost.

In the full observation case ($\mathbf{\Omega} = \mathbf{1}_d\mathbf{1}_d^\top$), we have already seen via Proposition 1 that the compressed factorization (5) with small initialization stays close to the trajectory of the full-width factorization. However, applying this directly to the projection $\mathbf{\Omega} \odot \mathbf{\Phi}$ will result in the compressed factorization's trajectory diverging from that of the original – see the orange trace in Figure 3. Intuitively, this is because the factors $\boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}$ are initialized from incomplete measurement of $\mathbf{\Phi}$ – instead, we optimize the modified objective

$$\widehat{\ell}_{\text{mc}}(\widehat{\mathbf{\Theta}}, \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}) = \frac{1}{2}\|\mathbf{\Omega} \odot (\widehat{f}(\widehat{\mathbf{\Theta}}, \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}) - \mathbf{\Phi})\|_F^2 \tag{8}$$

where $\widehat{\mathbf{\Theta}}$ are updated with learning rate η while the $\boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}$ factors are updated with a *discrepant* learning rate $\gamma\eta$ where $\gamma > 0$ is small. While this results in an additional $2d\widehat{r}$ parameters to be tracked, the trajectory of this compressed factorization will ultimately align with that of the original while converging roughly $5\times$ faster w.r.t. wall-time, as demonstrated in Figure 3. Moreover, the accelerated convergence induced by the full-width trajectory results in the compressed factorization being $3\times$ faster than randomly initialized factorizations of similar width – see Appendix C for more details.

4 Conclusion

This paper offers novel insights into simple structures in gradient descent for learning deep matrix factorizations, by which we derive some rigorous justification for the practice of low-rank training. Through this work, we hope to ultimately inspire more principled approaches to designing efficient and effective deep models by exploiting low-dimensional aspects of their training dynamics.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[3] Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019.

[4] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[5] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[6] Daniel Kunin, Atsushi Yamamura, Chao Ma, and Surya Ganguli. The asymmetric maximum margin bias of quasi-homogeneous neural networks. *arXiv preprint arXiv:2210.03820*, 2022.

[7] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[8] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023.

[9] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

[12] Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning, 2020.

[13] Samuel Horvath, Stefanos Laskaridis, Shashank Rajput, and Hongyi Wang. Maestro: Uncovering low-rank structures via trainable decomposition, 2023.

[14] Jiawei Zhao, Yifei Zhang, Beidi Chen, Florian Schäfer, and Anima Anandkumar. Inrank: Incremental low-rank learning, 2023.

[15] Hongyi Wang, Saurabh Agarwal, Pongsakorn U-chupala, Yoshiki Tanaka, Eric P. Xing, and Dimitris Papailiopoulos. Cuttlefish: Low-rank model training without all the tuning, 2023.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[17] Albert Gural, Phillip Nadeau, Mehul Tikekar, and Boris Murmann. Low-rank training of deep neural networks for emerging memory technology, 2020.

[18] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[19] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[20] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018.

[21] Hancheng Min, Salma Tarmoun, René Vidal, and Enrique Mallada. Convergence and implicit bias of gradient flow on overparametrized linear networks. *arXiv preprint arXiv:2105.06351*, 2022.

[22] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. *arXiv preprint arXiv:1909.12051*, 2019.

[23] Gal Vardi and Ohad Shamir. Implicit regularization in relu networks with the square loss. In *Conference on Learning Theory*, pages 4224–4258. PMLR, 2021.

[24] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

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

[26] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[27] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

[28] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[29] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

[30] Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

[31] Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

[32] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

[33] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29, 2016.

[34] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[35] Qiuwei Li, Zhihui Zhu, and Gongguo Tang. The non-convex geometry of low-rank matrix optimization. *Information and Inference: A Journal of the IMA*, 8(1):51–96, 2019.

[36] Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

[37] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.

[38] Mahdi Soltanolkotabi, Dominik Stöger, and Changzhi Xie. Implicit balancing and regularization: Generalization and convergence guarantees for overparameterized asymmetric matrix sensing. *arXiv preprint arXiv:2303.14244*, 2023.

Appendix

Notation. Given any $L \in \mathbb{N}$, we use $[L]$ to denote the index set $\{1, \ldots, L\}$. We use $\boldsymbol{I}_n \in R^n$ to denote the identity matrix of size n, and $\mathbf{1}_n$ to denote a vector of length n with all entries equal to 1. We denote by $\|\boldsymbol{A}\|_F^2$ the squared *Frobenius* norm of matrix $\boldsymbol{A}$, i.e., the sum of squares of all entries of $\boldsymbol{A}$. For convenience, whenever $j > i$ we adopt the abbreviations $\boldsymbol{W}_{j:i} = \boldsymbol{W}_j \cdots \boldsymbol{W}_i$ and $\boldsymbol{W}_{j:i}^\top = \boldsymbol{W}_i^\top \cdots \boldsymbol{W}_j^\top$, whereas both are identity if $j < i$.

A Proofs in Section 2

Substituting the analytic form of the gradient into (2), we have the update rule

$$\boldsymbol{W}_l(t+1) = (1 - \eta\lambda)\boldsymbol{W}_l(t) - \eta \boldsymbol{W}_{L:l+1}^\top(t) \boldsymbol{E}(t) \boldsymbol{W}_{l-1:1}^\top(t),\ l \in [L] \tag{9}$$

for $t = 0, 1, 2, \ldots$, where $\boldsymbol{E}(t) = f(\boldsymbol{\Theta}(t)) - \boldsymbol{\Phi}$.

We first establish the following Lemma 1 – the claim in Theorem 1 then follows in a relatively straightforward manner. We note that all statements quantified by i in this section implicity hold for all $i \in [m]$ (as defined in Theorem 1) for the sake of notational brevity.

A.1 Proof of Theorem 1

Lemma 1. *Under the setting of Theorem 1, there exist orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m \subset \mathbf{R}^d$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m \subset \mathbf{R}^d$ for $l \in [L]$ satisfying $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$ such that the following hold for all $t \geq 0$:*

$$\begin{aligned}
\mathcal{A}(t) &: \boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} = \rho_l(t)\boldsymbol{u}_i^{(l)} \quad \forall l \in [L], \\
\mathcal{B}(t) &: \boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} = \rho_l(t)\boldsymbol{v}_i^{(l)} \quad \forall l \in [L], \\
\mathcal{C}(t) &: \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} = \mathbf{0} \quad \forall l \in [L], \\
\mathcal{D}(t) &: \boldsymbol{\Phi} \boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} = \mathbf{0} \quad \forall l \in [L],
\end{aligned}$$

where $\rho_l(t) = \rho_l(t-1) \cdot (1 - \eta\lambda - \eta \cdot \prod_{k \neq l} \rho_k(t-1)^2)$ for all $t \geq 1$ with $\rho_l(0) = \sigma_l > 0$.

Proof. Define $\boldsymbol{\Psi} := \boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Phi}$. Since the rank of $\boldsymbol{\Phi}$ is at most r, we have that the rank of $\boldsymbol{\Psi} \in \mathbf{R}^{d \times d}$ is at most r, which implies that $\dim \mathcal{N}(\boldsymbol{\Psi}) = \dim \mathcal{N}\left(\boldsymbol{\Psi}^\top\right) \geq d - r$. We define the subspace

$$\mathcal{S} := \mathcal{N}(\boldsymbol{\Psi}) \cap \mathcal{N}\left(\boldsymbol{\Psi}^\top \boldsymbol{W}_1(0)\right) \subset \mathbf{R}^d.$$

Since $\boldsymbol{W}_1(0) \in \mathbf{R}^{d \times d}$ is nonsingular, we have

$$\dim(\mathcal{S}) \geq 2(d - r) - d = m.$$

Let $\{\boldsymbol{v}_i^{(1)}\}_{i=1}^m$ denote an orthonormal set contained in $\mathcal{S}$ and set $\boldsymbol{u}_i^{(1)} := \boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)}/\sigma_1$, where $\sigma_1 > 0$ is the scale of $\boldsymbol{W}_1(0)$ – since $\boldsymbol{W}_1(0)/\sigma_1$ is orthogonal, $\{\boldsymbol{u}_i^{(1)}\}_{i=1}^m$ is also an orthonormal set. Then we trivially have $\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \sigma_1 \boldsymbol{u}_i^{(1)}$, which implies $\boldsymbol{W}_1^\top(0)\boldsymbol{u}_i^{(1)} = \sigma_1 \boldsymbol{v}_i^{(1)}$. It follows from $\boldsymbol{v}_i^{(1)} \in \mathcal{S}$ that $\boldsymbol{\Psi}\boldsymbol{v}_i^{(1)} = \mathbf{0}$ and $\boldsymbol{\Psi}^\top \boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \mathbf{0}$, which is equivalent to $\boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \mathbf{0}$ and $\boldsymbol{\Phi}^\top \boldsymbol{W}_{L:2}(0)\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \sigma_1 \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:2}(0)\boldsymbol{u}_i^{(1)} = \mathbf{0}$ respectively. Since $\boldsymbol{W}_{L:2}^\top(0)$ is full column rank, we further have that $\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \mathbf{0}$.

Now let $\mathcal{E}(l)$ be the event that we have orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m$ satisfying $\boldsymbol{W}_l(0)\boldsymbol{v}_i^{(l)} = \sigma_l \boldsymbol{u}_i^{(l)}$, $\boldsymbol{W}_l^\top(0)\boldsymbol{u}_i^{(l)} = \sigma_l \boldsymbol{v}_i^{(l)}$, $\boldsymbol{\Phi}^\top \boldsymbol{W}_{L:l+1}(0)\boldsymbol{u}_i^{(l)} = \mathbf{0}$, and $\boldsymbol{\Phi}\boldsymbol{W}_{l-1:1}^\top(0)\boldsymbol{v}_i^{(l)} = \mathbf{0}$. From the above arguments, we have that $\mathcal{E}(1)$ holds – now suppose $\mathcal{E}(k)$ holds for some $1 \leq k < L$.

Set $\boldsymbol{v}_i^{(k+1)} := \boldsymbol{u}_i^{(k)}$ and $\boldsymbol{u}_i^{(k+1)} := \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\sigma_{k+1}$. This implies that $\boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)} = \sigma_{k+1}\boldsymbol{u}_i^{(k+1)}$ and $\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)} = \sigma_{k+1}\boldsymbol{v}_i^{(k+1)}$. Moreover, we have

$$\begin{aligned}\boldsymbol{\Phi}^\top \boldsymbol{W}_{L:(k+1)+1}(0)\boldsymbol{u}_i^{(k+1)} &= \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:k+1}(0)\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)}/\sigma_{k+1}^2 \\ &= \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:k+1}(0)\boldsymbol{v}_i^{(k+1)}/\sigma_{k+1} \\ &= \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:k+1}(0)\boldsymbol{u}_i^{(k)}/\sigma_{k+1} = \boldsymbol{0},\end{aligned}$$

where the first two equalities follow from orthogonality and $\boldsymbol{u}_i^{(k+1)} = \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\sigma_{k+1}$, and the last equality is due to $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$. Similarly, we have

$$\begin{aligned}\boldsymbol{\Phi}\boldsymbol{W}_{(k+1)-1:1}^\top(0)\boldsymbol{v}_i^{(k+1)} &= \boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{v}_i^{(k+1)} \\ &= \boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} \\ &= \sigma_k\boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{v}_i^{(k)} = \boldsymbol{0},\end{aligned}$$

where the second equality follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and the third equality is due to $\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} = \sigma_k\boldsymbol{v}_i^{(k)}$. Therefore $\mathcal{E}(k+1)$ holds, so we have $\mathcal{E}(l)$ for all $l \in [L]$. As a result, we have shown the base cases $\mathcal{A}(0)$, $\mathcal{B}(0)$, $\mathcal{C}(0)$, and $\mathcal{D}(0)$.

Now we proceed by induction on $t \geq 0$. Suppose that $\mathcal{A}(t)$, $\mathcal{B}(t)$, $\mathcal{C}(t)$, and $\mathcal{D}(t)$ hold for some $t \geq 0$. First, we show $\mathcal{A}(t+1)$ and $\mathcal{B}(t+1)$. We have

$$\begin{aligned}\boldsymbol{W}_l(t+1)\boldsymbol{v}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l(t) - \eta\boldsymbol{W}_{L:l+1}^\top(t)\boldsymbol{E}(t)\boldsymbol{W}_{l-1:1}^\top(t)\right]\boldsymbol{v}_i^{(l)} \\ &= \left[(1-\eta\lambda)\boldsymbol{W}_l(t) - \eta\boldsymbol{W}_{L:l+1}^\top(t)\left(\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}\right)\boldsymbol{W}_{l-1:1}^\top(t)\right]\boldsymbol{v}_i^{(l)} \\ &= (1-\eta\lambda)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} - \eta\boldsymbol{W}_{L:l+1}^\top(t)\boldsymbol{W}_{L:1}(t)\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} \\ &= (1-\eta\lambda)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} - \eta\cdot\Big(\prod_{k\neq l}\rho_k^2(t)\Big)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} \\ &= \rho_l(t)\cdot\Big(1-\eta\lambda-\eta\cdot\prod_{k\neq l}\rho_k^2(t)\Big)\boldsymbol{u}_i^{(l)} = \rho_l(t+1)\boldsymbol{u}_i^{(l)}\end{aligned}$$

for all $l \in [L]$, where the first equality follows from (9), the second equality follows from definition of $\boldsymbol{E}(t)$, the third equality follows from $\mathcal{D}(t)$, and the fourth equality follows from $\mathcal{A}(t)$ and $\mathcal{B}(t)$ applied repeatedly along with $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$, proving $\mathcal{A}(t+1)$. Similarly, we have

$$\begin{aligned}\boldsymbol{W}_l^\top(t+1)\boldsymbol{u}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l^\top(t) - \eta\boldsymbol{W}_{l-1:1}(t)\boldsymbol{E}^\top(t)\boldsymbol{W}_{L:l+1}(t)\right]\boldsymbol{u}_i^{(l)} \\ &= \left[(1-\eta\lambda)\boldsymbol{W}_l^\top(t) - \eta\boldsymbol{W}_{l-1:1}(t)\left(\boldsymbol{W}_{L:1}^\top(t) - \boldsymbol{\Phi}^\top\right)\boldsymbol{W}_{L:l+1}(t)\right]\boldsymbol{u}_i^{(l)} \\ &= (1-\eta\lambda)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} - \eta\boldsymbol{W}_{l-1:1}(t)\boldsymbol{W}_{L:1}^\top(t)\boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} \\ &= (1-\eta\lambda)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} - \eta\cdot\Big(\prod_{k\neq l}\rho_k^2(t)\Big)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} \\ &= \rho_l(t)\cdot\Big(1-\eta\lambda-\eta\cdot\prod_{k\neq l}\rho_k^2(t)\Big)\boldsymbol{v}_i^{(l)} = \rho_l(t+1)\boldsymbol{v}_i^{(l)}\end{aligned}$$

for all $l \in [L]$, where the third equality follows from $\mathcal{C}(t)$, and the fourth equality follows from $\mathcal{A}(t)$ and $\mathcal{B}(t)$ applied repeatedly along with $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$, proving $\mathcal{B}(t+1)$. Now, we show $\mathcal{C}(t+1)$. For any $k \in [L-1]$, it follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and $\mathcal{A}(t+1)$ that

$$\boldsymbol{W}_{k+1}(t+1)\boldsymbol{u}_i^{(k)} = \boldsymbol{W}_{k+1}(t+1)\boldsymbol{v}_i^{(k+1)} = \rho_{k+1}(t+1)\boldsymbol{u}_i^{(k+1)}.$$

Repeatedly applying the above equality for $k = l, l+1, \ldots, L-1$, we obtain

$$\boldsymbol{\Phi}^\top\boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} = \left[\prod_{k=l}^{L-1}\rho_{k+1}(t)\right]\cdot\boldsymbol{\Phi}^\top\boldsymbol{u}_i^{(L)} = \boldsymbol{0}$$

which follows from $\mathcal{C}(t)$, proving $\mathcal{C}(t+1)$. Finally, we show $\mathcal{D}(t+1)$. For any $k \in \{2, \dots, L\}$, it follows from $\boldsymbol{v}_i^{(k)} = \boldsymbol{u}_i^{(k-1)}$ and $\mathcal{B}(t+1)$ that

$$\boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{v}_i^{(k)} = \boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{u}_i^{(k-1)} = \rho_{k-1}(t+1)\boldsymbol{v}_i^{(k-1)}.$$

Repeatedly applying the above equality for $k = l, l-1, \dots, 2$, we obtain

$$\boldsymbol{\Phi}\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} = \left[\prod_{k=2}^{l} \rho_{k-1}(t)\right] \cdot \boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$$

which follows from $\mathcal{D}(t)$. Thus we have proven $\mathcal{D}(t+1)$, concluding the proof. □

Proof of Theorem 1. By $\mathcal{A}(t)$ and $\mathcal{B}(t)$ of Lemma 1, there exists orthonormal matrices $\{\boldsymbol{U}_{l,2}\}_{l=1}^L \subset \mathbf{R}^{d\times m}$ and $\{\boldsymbol{V}_{l,2}\}_{l=1}^L \subset \mathbf{R}^{d\times m}$ for $l \in [L]$ satisfying $\boldsymbol{U}_{l+1,2} = \boldsymbol{V}_{l,2}$ for all $l \in [L-1]$ as well as

$$\boldsymbol{W}_l(t)\boldsymbol{V}_{l,2} = \rho_l(t)\boldsymbol{U}_{l,2} \quad \text{and} \quad \boldsymbol{W}_l(t)^\top\boldsymbol{U}_{l,2} = \rho_l(t)\boldsymbol{V}_{l,2} \tag{10}$$

for all $l \in [L]$ and $t \geq 0$, where $\rho_l(t)$ satisfies (4) for $t \geq 1$ with $\rho_l(0) = \sigma_l$. First, complete $\boldsymbol{V}_{1,2}$ to an orthonormal basis for $\mathbf{R}^d$ as $\boldsymbol{V}_1 = [\boldsymbol{V}_{1,1}\ \boldsymbol{V}_{1,2}]$. Then for each $l \in [L-1]$, set $\boldsymbol{U}_l = [\boldsymbol{U}_{l,1}\ \boldsymbol{U}_{l,2}]$ where $\boldsymbol{U}_{l,1} = \boldsymbol{W}_l(0)\boldsymbol{V}_{l,1}/\sigma_l$ and $\boldsymbol{V}_{l+1} = [\boldsymbol{V}_{l+1,1}\ \boldsymbol{V}_{l+1,2}]$ where $\boldsymbol{V}_{l+1,1} = \boldsymbol{U}_{l,1}$, and finally set $\boldsymbol{U}_L = [\boldsymbol{U}_{L,1}\ \boldsymbol{U}_{L,2}]$ where $\boldsymbol{U}_{L,1} = \boldsymbol{W}_L(0)\boldsymbol{V}_{L,1}/\sigma_L$. We note that $\boldsymbol{V}_{l+1} = \boldsymbol{U}_l$ for each $l \in [L-1]$. Then we have

$$\boldsymbol{U}_{l,1}^\top\boldsymbol{W}_l(t)\boldsymbol{V}_{l,2} = \rho_l(t)\boldsymbol{U}_{l,1}^\top\boldsymbol{U}_{l,2} = \boldsymbol{0} \tag{11}$$

for all $l \in [L]$, where the first equality follows from (10). Similarly, we also have

$$\boldsymbol{U}_{l,2}^\top\boldsymbol{W}_l(t)\boldsymbol{V}_{l,1} = \rho(t)\boldsymbol{V}_{l,2}^\top\boldsymbol{V}_{l,1} = \boldsymbol{0} \tag{12}$$

for all $l \in [L]$, where the first equality also follows from (10). Therefore, combining (10), (11), and (12) yields

$$\boldsymbol{U}_l^\top\boldsymbol{W}_l(t)\boldsymbol{V}_l = [\boldsymbol{U}_{l,1} \quad \boldsymbol{U}_{l,2}]^\top\boldsymbol{W}_l(t)\,[\boldsymbol{V}_{l+1,1} \quad \boldsymbol{V}_{l+1,2}] = \begin{bmatrix} \widetilde{\boldsymbol{W}}_l(t) & \boldsymbol{0} \\ \boldsymbol{0} & \rho_l(t)\boldsymbol{I}_m \end{bmatrix}$$

for all $l \in [L]$, where $\widetilde{\boldsymbol{W}}_l(0) = \sigma_l\boldsymbol{I}_{2r}$ by construction of $\boldsymbol{U}_{l,1}$. This directly implies (3), completing the proof. □

A.2 Proof of Proposition 1

Proof. First, it follows from Theorem 1 that for any $1 \leq i \leq j \leq L$ we have

$$\boldsymbol{W}_{j:i}(t) = \boldsymbol{U}_{j,1}\widetilde{\boldsymbol{W}}_{j:i}(t)\boldsymbol{V}_{i,1}^\top + (\prod_{k=i}^{j}\rho_k(t)) \cdot \boldsymbol{U}_{j,2}\boldsymbol{V}_{i,2}^\top \tag{13}$$

for all $t \geq 0$, where $\boldsymbol{U}_{l,1}, \boldsymbol{V}_{l,1} \in \mathbf{R}^{d\times 2\widehat{r}}$ and $\boldsymbol{U}_{l,2}, \boldsymbol{V}_{l,2} \in \mathbf{R}^{d\times \widehat{m}}$ are the first $2\widehat{r}$ and last $\widehat{m}$ columns of $\boldsymbol{U}_l, \boldsymbol{V}_l \in \mathbf{R}^{d\times d}$ respectively.

The key claim to be shown here is that $\widehat{\boldsymbol{W}}_l(t) = \widetilde{\boldsymbol{W}}_l(t)$ for all $l \in [L]$ and $t \geq 0$. Afterwards, it follows straightforwardly from (13) that

$$\begin{aligned}
&\left\| f(\boldsymbol{\Theta}(t)) - \widehat{f}(\widehat{\boldsymbol{\Theta}}(t), \boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1}) \right\|_F^2 \\
&= \left\| \boldsymbol{U}_{L,1}\widetilde{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{1,1}^\top + (\prod_{l=1}^{L}\rho_l(t)) \cdot \boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top - \boldsymbol{U}_{L,1}\widehat{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{L,1}^\top \right\|_F^2 \\
&= \left\| \boldsymbol{U}_{L,1}(\widetilde{\boldsymbol{W}}_{L:1}(t) - \widehat{\boldsymbol{W}}_{L:1}(t))\boldsymbol{V}_{1,1}^\top + (\prod_{l=1}^{L}\rho_l(t)) \cdot \boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top \right\|_F^2 \\
&= \left\| (\prod_{l=1}^{L}\rho_l(t)) \cdot \boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top \right\|_F^2 \leq \widehat{m} \cdot \prod_{l=1}^{L}\sigma_l^2.
\end{aligned}$$

We proceed by induction. For $t = 0$, we have that

$$\widehat{\boldsymbol{W}}_l(0) = \boldsymbol{U}_{l,1}^\top \boldsymbol{W}_l(0) \boldsymbol{V}_{l,1} = \widetilde{\boldsymbol{W}}_l(0)$$

for all $l \in [L]$ by (13) and choice of initialization.

Now suppose $\widehat{\boldsymbol{W}}_l(t) = \widetilde{\boldsymbol{W}}_l(t)$ for all $l \in [L]$. Comparing

$$\widehat{\boldsymbol{W}}_l(t+1) = (1-\eta\lambda)\widehat{\boldsymbol{W}}_l(t) - \eta \nabla_{\widehat{\boldsymbol{W}}_l} \widehat{\ell}(\widehat{\boldsymbol{\Theta}}(t))$$

with

$$\begin{aligned}
\widetilde{\boldsymbol{W}}_l(t+1) &= \boldsymbol{U}_{l,1}^\top \boldsymbol{W}_l(t+1) \boldsymbol{V}_{l,1} \\
&= \boldsymbol{U}_{l,1}^\top \left[(1-\eta\lambda) \boldsymbol{W}_l(t) - \eta \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)) \right] \boldsymbol{V}_{l,1} \\
&= (1-\eta\lambda)\widetilde{\boldsymbol{W}}_l(t) - \eta \boldsymbol{U}_{l,1}^\top \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)) \boldsymbol{V}_{l,1}
\end{aligned}$$

it suffices to show that

$$\nabla_{\widehat{\boldsymbol{W}}_l} \widehat{\ell}(\widehat{\boldsymbol{\Theta}}(t)) = \boldsymbol{U}_{l,1}^\top \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)) \boldsymbol{V}_{l,1}, \ \forall l \in [L] \tag{14}$$

to yield $\widehat{\boldsymbol{W}}_l(t+1) = \widetilde{\boldsymbol{W}}_l(t+1)$ for all $l \in [L]$. Computing the right hand side of (14), we have

$$\begin{aligned}
\boldsymbol{U}_{l,1}^\top \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)) \boldsymbol{V}_{l,1} &= \boldsymbol{U}_{l,1}^\top \boldsymbol{W}_{L:l+1}^\top(t) (\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}) \boldsymbol{W}_{l-1:1}^\top(t) \boldsymbol{V}_{l,1} \\
&= (\boldsymbol{W}_{L:l+1}(t) \boldsymbol{U}_{l,1})^\top (\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}) (\boldsymbol{V}_{l,1}^\top \boldsymbol{W}_{l-1:1}(t))^\top
\end{aligned}$$

where

$$\boldsymbol{W}_{L:l+1}(t)\boldsymbol{U}_{l,1} = \left(\boldsymbol{U}_{L,1} \widetilde{\boldsymbol{W}}_{L:l+1}(t) \boldsymbol{V}_{l+1,1}^\top + \Big(\prod_{k=l+1}^{L} \rho_k(t) \Big) \cdot \boldsymbol{U}_{L,2} \boldsymbol{V}_{l+1,2}^\top \right) \boldsymbol{U}_{l,1} = \boldsymbol{U}_{L,1} \widetilde{\boldsymbol{W}}_{L:l+1}(t)$$

by (13) and the fact that $\boldsymbol{U}_l = \boldsymbol{V}_{l+1}$, and similarly

$$\boldsymbol{V}_{l,1}^\top \boldsymbol{W}_{l-1:1}(t) = \boldsymbol{V}_{l,1}^\top \left(\boldsymbol{U}_{l-1,1} \widetilde{\boldsymbol{W}}_{l-1:1}(t) \boldsymbol{V}_{1,1}^\top + \Big(\prod_{k=1}^{l-1} \rho_k(t) \Big) \cdot \boldsymbol{U}_{l-1,2} \boldsymbol{V}_{1,2}^\top \right) = \widetilde{\boldsymbol{W}}_{l-1:1}(t) \boldsymbol{V}_{1,1}^\top.$$

We also have that

$$\begin{aligned}
\boldsymbol{U}_{L,1}^\top (\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}) \boldsymbol{V}_{1,1} &= \boldsymbol{U}_{L,1}^\top \left(\boldsymbol{U}_{L,1} \widetilde{\boldsymbol{W}}_{L:1}(t) \boldsymbol{V}_{1,1}^\top + \Big(\prod_{k=1}^{L} \rho_k(t) \Big) \cdot \boldsymbol{U}_{L,2} \boldsymbol{V}_{1,2}^\top - \boldsymbol{\Phi} \right) \boldsymbol{V}_{1,1} \\
&= \widetilde{\boldsymbol{W}}_{L:1}(t) - \boldsymbol{U}_{L,1}^\top \boldsymbol{\Phi} \boldsymbol{V}_{1,1}
\end{aligned}$$

so putting together the previous four equalities yields

$$\begin{aligned}
\boldsymbol{U}_{l,1}^\top \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t)) \boldsymbol{V}_{l,1} &= (\boldsymbol{W}_{L:l+1}(t) \boldsymbol{U}_{l,1})^\top (\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}) (\boldsymbol{V}_{l,1}^\top \boldsymbol{W}_{l-1:1}(t))^\top \\
&= \widetilde{\boldsymbol{W}}_{L:l+1}^\top(t) \boldsymbol{U}_{L,1}^\top (\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}) \boldsymbol{V}_{1,1} \widetilde{\boldsymbol{W}}_{l-1:1}^\top(t) \\
&= \widetilde{\boldsymbol{W}}_{L:l+1}^\top(t) (\widetilde{\boldsymbol{W}}_{L:1}(t) - \boldsymbol{U}_{L,1}^\top \boldsymbol{\Phi} \boldsymbol{V}_{1,1}) \widetilde{\boldsymbol{W}}_{l-1:1}^\top(t).
\end{aligned}$$

On the other hand, the left hand side of (14) gives

$$\begin{aligned}
\nabla_{\widehat{\boldsymbol{W}}_l} \widehat{\ell}(\widehat{\boldsymbol{\Theta}}(t)) &= \widehat{\boldsymbol{W}}_{L:l+1}(t)^\top \boldsymbol{U}_{L,1}^\top (\boldsymbol{U}_{L,1} \widehat{\boldsymbol{W}}_{L:1}(t) \boldsymbol{V}_{1,1}^\top - \boldsymbol{\Phi}) \boldsymbol{V}_{1,1} \widehat{\boldsymbol{W}}_{l-1:1}(t)^\top \\
&= \widehat{\boldsymbol{W}}_{L:l+1}(t)^\top (\widehat{\boldsymbol{W}}_{L:1}(t) - \boldsymbol{U}_{L,1}^\top \boldsymbol{\Phi} \boldsymbol{V}_{1,1}) \widehat{\boldsymbol{W}}_{l-1:1}(t)^\top
\end{aligned}$$

so (14) holds by the fact that $\widehat{\boldsymbol{W}}_l(t) = \widetilde{\boldsymbol{W}}_l(t)$ for all $l \in [L]$, completing the proof. □

B Benefits of Over-Parameterization in Deep Matrix Completion

In the setup described in Section 3, we claim that depth and width are beneficial for accelerating GD convergence to well-generalizing solutions, and therefore constructing more computationally efficient factorizations that share the same trajectory is a fruitful endeavour. Below, we give a more detailed explanation of these ideas:

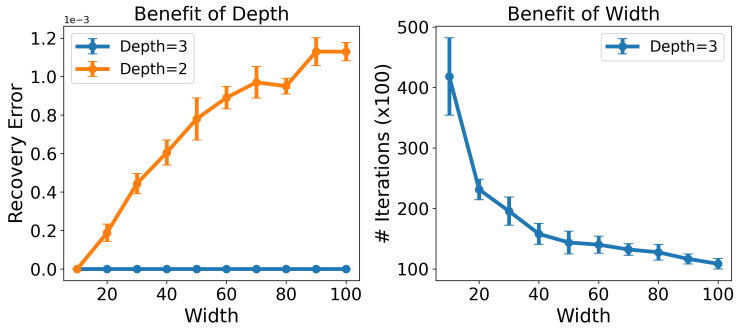

Figure 4: **Benefits of depth & width in overparameterized matrix completion** with $d = 100$, $r = 5$, $\sigma_l = 10^{-3}$ and 30% of entries observed. *Left*: Recovery error. *Right*: Number of GD iterations to converge to 10^{-10} error.

- **Benefits of depth.** When $L = 2$, (7) reduces to Burer-Monteiro factorization [28], whose global optimality and convergence under GD have been widely studied under various settings [9, 29–38]. However, it has been demonstrated [11] that in the over-parameterized regime $\widehat{r} > r$, deeper factorizations (starting from small random initialization) continue to generalize well beyond the exact parameterization $\widehat{r} = r$ unlike their shallow counterparts, see Figure 4 (left).
- **Benefits of width.** On the other hand, increasing the width $\widehat{r}$ of the deep factorization beyond r results in accelerated convergence of GD in terms of iterations, see Figure 4 (right).

C Compressed vs. Narrow Factorizations

We compare the training efficiency of a $2\widehat{r}$-compressed factorization (with trajectory equivalent to a wide factorization of width $d \gg \widehat{r}$) versus a narrow factorization with width $2\widehat{r}$ under different over-parameterized estimates $\widehat{r}$. As depicted in Figure 5 (left), the compressed factorization requires fewer iterations to reach convergence, and the number of iterations necessary is almost unaffected by $\widehat{r}$. Consequently, **training compressed factorizations is considerably more time-efficient than training narrow ones of the same size**, provided that $\widehat{r}$ is not significantly larger than r.

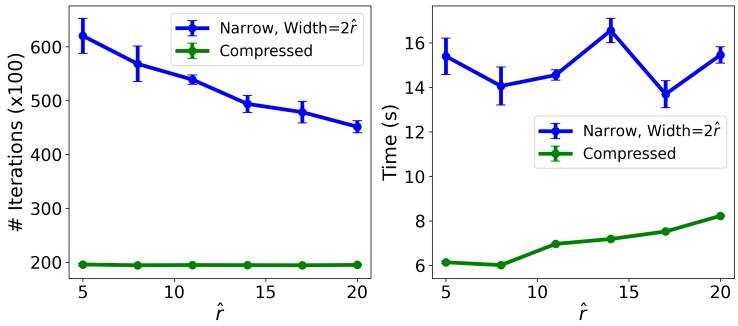

Figure 5: **Efficiency of compressed vs. narrow factorizations** for different overestimated $\widehat{r}$ with $L = 3$, $d = 1000$, $r = 5$, $\sigma_l = 10^{-3}$ and 20% of entries observed. *Left*: Number of iterations to converge to 10^{-10} error. *Right*: Time to converge.

The distinction between the compressed and narrow factorizations underscores the benefits of width, as previously demonstrated and discussed in Figure 4 (right), where increasing the width results in faster convergence. However, increasing the width alone also increases the number of parameters. By employing our compression methodology, we can achieve the best of both worlds.