ENSAE Paris

Understanding training dynamics and low-rank bias in NMF and Deep NMF

Advanced Machine Learning Project

Malo DAVID, Thomas LAMBELIN, Maxime CHANSAT

Janvier 2026

Abstract

Nonnegative Matrix Factorization (NMF) and its deep extensions are widely used for interpretable representation learning, yet their optimization dynamics and implicit regularizations remain only partially understood. In this work, we study NMF and Deep NMF on the MNIST handwritten digits dataset through a combined theoretical and empirical approach. We reimplement standard algorithms from the literature in a modular PyTorch framework, enabling controlled and reproducible experiments. Building on the recent analysis of gradient descent dynamics for deep matrix factorization by Hung-Hsu Chou and al. [1], we focus on the low-rank bias exhibited by NMF models trained with gradient-based methods. We partially extend this line of work to the nonnegative, non-symmetric setting with random initialization, and complement it with an empirical study on MNIST. By tracking the effective rank of the reconstruction during training, we observe a robust plateau structure, suggesting that model complexity increases in discrete stages that correlate with the emergence of more structured features. Finally, we compare different loss formulations, as suggested by the work of De Handschutter and al. [2], and highlight their qualitative impact on the learned archetypes. The full codebase is available at https://github.com/thomasalensae/NMF.

Contents

1 Introduction **2**
- 1.1 Motivations and context . . . 2
- 1.2 Our project's contribution . . . 2

2 Theoretical framework of NMF **3**
- 2.1 Principle and motivation . . . 3
- 2.2 Mathematical formulation . . . 3
- 2.3 Optimization aspects . . . 3
- 2.4 Implementation . . . 4
- 2.5 The low-rank bias . . . 4

3 Original contributions: low-rank bias in Deep NMF **6**
- 3.1 Tackling the low rank bias . . . 6
- 3.2 Dynamics of the singular values at the beginning of training . . . 7

4 Empirical results and observations **11**
- 4.1 Qualitative dynamics of archetypes and effective rank during training . . . 11
- 4.2 Comparison of different losses . . . 12

A The Adam optimizer **17**

B Archetypal images of H **18**

C Archetypal images for layer loss, $r_1 = 20$ and $r_2 = 10$ **19**

D Metrics evolution **20**

E Bound for the error **21**

1 Introduction

Nonnegative Matrix Factorization (NMF) is a classical representation learning technique that decomposes a nonnegative data matrix into the product of two (lower-)dimensional nonnegative factors. Since its popularization in machine learning and signal processing, NMF has been widely used in applications where interpretability is essential, including topic modeling, image analysis, bioinformatics, and audio source separation. The nonnegativity constraint induces a parts-based representation, often leading to features that are sparse, localized, and semantically meaningful ([3]).

1.1 Motivations and context

Despite the apparent simplicity of its formulation (see Section 2), NMF defines a highly nonconvex optimization problem, and its behavior depends strongly on the choice of loss function, optimization algorithm, initialization, and implicit regularization induced by training dynamics. As a result, understanding what NMF actually learns in practice can be difficult.

More recently, several works have proposed *Deep NMF* architectures, in which the factorization is decomposed across multiple layers (cf. Section 2). These models can be interpreted as hierarchical extensions of NMF, inspired by deep learning, and aim to capture more complex, organic or multi-scale structure in the data while preserving nonnegativity and interpretability. While Deep NMF has shown empirical promise in some settings, its theoretical properties and training dynamics are even less well understood than those of shallow NMF (see, for instance, the Conclusion of [4]).

A recurring empirical observation in matrix factorization models is the presence of an implicit *low-rank bias*: solutions tend to favor (effective) low-rank representations, even in the absence of explicit regularization during training. Similar phenomena have been extensively studied in linear networks, matrix sensing, and deep learning more broadly, where optimization dynamics bias solutions toward simpler or lower-complexity structures (cf. [1]). In the context of NMF, this bias raises natural questions: how does rank emerge during training, how does it evolve over time, and how is it reflected in the learned features?

1.2 Our project's contribution

In this project, we aim to contribute to this understanding through a focused both theoretical and empirical study of NMF and Deep NMF on the MNIST handwritten digits dataset. We reimplement existing methods in a modular PyTorch framework and systematically explore their behavior under different loss functions and architectural choices. A central emphasis is placed on the evolution of the effective rank during training and its correspondence with qualitative changes in the learned archetypal features.

2 Theoretical framework of NMF

2.1 Principle and motivation

The emergence of high-dimensional data processing has made classical statistical methods difficult to apply, particularly when the ratio between sample size and dimension is no longer negligible. Dimensionality reduction techniques, including NMF, address this challenge while preserving the interpretability of results.

2.2 Mathematical formulation

NMF consists in approximating a data matrix $X \in M_{n,m}(\mathbb{R}_+)$ by the product of two reduced-dimension matrices $W \in M_{n,r}(\mathbb{R}_+)$ and $H \in M_{r,m}(\mathbb{R}_+)$, where $r \ll \min(n, m)$. Formally, given a distance d on the matrix space, the problem can be written as:

$$\min_{\substack{W \in M_{n,r}(\mathbb{R}_+) \\ H \in M_{r,m}(\mathbb{R}_+)}} d(X, WH) \tag{1}$$

Since the Frobenius norm is the most commonly used metric, the optimization problem becomes:

$$\min_{\substack{W \geq 0 \\ H \geq 0}} \frac{1}{2} \|X - WH\|_F^2 \tag{2}$$

Other divergences, notably the Kullback-Leibler divergence, are also employed (cf. [5]), although we focus here on the Frobenius norm.

2.3 Optimization aspects

The objective function $\ell : (W, H) \mapsto \|X - WH\|_F^2$ is not convex, which complicates reaching the global minimum. Nevertheless, its continuity guarantees the existence of local minima. NMF algorithms therefore seek to converge toward satisfactory local minima.

[5] proved the convergence of multiplicative update (MU) algorithms to a local minimum. For projected gradient descent (PGD), although the decrease of the cost function is guaranteed, the convergence of the sequences $(W^{(k)})_k$ and $(H^{(k)})_k$ is not immediate. Indeed, the invariance under change of basis ($WPP^{-1}H = WH$ for any $P \in \mathrm{GL}_r(\mathbb{R})$) introduces non-uniqueness of the solution.

However, since $M_{m,n}(\mathbb{R}_+)$ is a closed convex set, and the gradient of ℓ is Lipschitz continuous, PGD with an appropriate learning rate (following for instance the Armijo rule) and under constraints on the norms of W and H, can guarantee convergence (as discussed by [6]).

2.4 Implementation

Although Python libraries such as `scikit-learn` provide NMF implementations, we developed our own version in PyTorch, in order to make it more modifiable. This approach allows:

- A better understanding of the internal mechanisms
- Flexibility to implement variants (Deep NMF, regularizations, different losses)
- Exploitation of GPU acceleration via CUDA (cf. [7])
- Use of the Adam Optimizer

The projected gradient descent algorithm is written, with a learning rate $\eta > 0$ and an initialization $(W^{(0)}, H^{(0)})$:

$$\begin{cases} W^{(t+1)} = \max(0, W^{(t)} - \eta \nabla_W \ell(W^{(t)}, H^{(t)})) \\ H^{(t+1)} = \max(0, H^{(t)} - \eta \nabla_H \ell(W^{(t)}, H^{(t)})) \end{cases} \tag{3}$$

where the gradients are given by:

$$\begin{cases} \nabla_W \ell(W, H) = (WH - X)H^\top \\ \nabla_H \ell(W, H) = W^\top (WH - X) \end{cases} \tag{4}$$

The function $\max(0, \cdot)$ (ReLU operator applied component-wise) ensures compliance with the non-negativity constraint featured by NMF.

Regarding initialization, we primarily use random matrices uniformly distributed on $[0, 1]$. Other methods exist, notably NNDSVD (Non-Negative Double Singular Value Decomposition) initialization, based on singular value decomposition, which can improve convergence in certain cases (cf. [8, 9]).

We also used the Adam optimizer, which we thought would help reaching better local minima, in particular because of the non-convex setup, and because the training was costly in time on our local GPUs. We also thought it would be a practical idea to implement that was rather close to the lecture notes and its chapter. The details can be found in Appendix A.

2.5 The low-rank bias

To quantify the notion of rank in a continuous manner, we employ the concept of effective rank introduced in [10]. For a given matrix A, the effective rank is defined as the ratio between the nuclear norm and the operator norm:

$$\mathrm{ER}(A) = \frac{\|A\|_{\text{nuclear}}}{\|A\|_{\text{op}}}$$

These norms are expressed in terms of the singular values of A. Denoting $\{\sigma_1, \sigma_2, \ldots, \sigma_{n\wedge m)}\}$ the collection of singular values of $A \in M_{m,n}(\mathbb{R})$ (with multiplicity), we have:

$$\begin{cases} \|A\|_{\text{op}} = \max_i \sigma_i \\ \|A\|_{\text{nuclear}} = \sum_i \sigma_i \end{cases}$$

Note that singular values are necessarily non-negative, as they are defined as the square roots of the eigenvalues of $A^\top A$, which is a symmetric positive semi-definite matrix.

The low-rank bias in our setting arises from the fact that the singular values are learned at different convergence rates under gradient descent, even though the loss function does not explicitly encode such a preference. This phenomenon becomes apparent when plotting the effective rank during training, where one observes characteristic plateaus as well as phases of increase or even temporary decrease (see Section 4.1).

To our knowledge, this phenomenon has not been described regarding NMF or Deep NMF. [1] presents a literature review that this phenomenon has mostly been documented when the ground-truth is symmetric. Because of that, the eigenvalues and the singular values of the ground-truth are identical in virtue of the spectral theorem.

3 Original contributions: low-rank bias in Deep NMF

3.1 Tackling the low rank bias

Several things differ from our experiments and [1]. First of all, [1], in a literature review, states that most articles tackle that problem assuming a symmetric or even PSD ground-truth. In our case, X has for only restriction to have positive coefficients. Moreover, most of the time, initializations are identities, whereas our initializations are random uniformly. Finally, we used the optimizer Adam, unlike vanilla gradient descent that is used in [1]. Therefore, the fact that we still see that phenomenon arises hints at the fact it is linked to gradient descent. We will only develop the idea with the Frobenius loss and three matrices W_1, W_2, H, but it can also be derived for more matrices:

$$\mathcal{L}(W_1, W_2, H) = \frac{1}{2}\|X - W_1 W_2 H\|_F^2.$$

A possible direction regarding an eventual proof is to study the gradient flow associated to Deep NMF, defined as the solution to the differential equation obtained (assuming it exists) when the learning rate η converges to 0.

Gradient computation. Using the chain rule, the gradients of $\mathcal{L}$ with respect to W_1, W_2 and H are

$$\begin{aligned}
\nabla_{W_1}\mathcal{L} &= (W_1 W_2 H - X)(W_2 H)^\top, \\
\nabla_{W_2}\mathcal{L} &= W_1^\top (W_1 W_2 H - X) H^\top, \\
\nabla_{H}\mathcal{L} &= W_2^\top W_1^\top (W_1 W_2 H - X).
\end{aligned}$$

Gradient descent and continuous-time limit. Recall our gradient descent verifies:

$$W_1^{(k+1)} = W_1^{(k)} - \eta \nabla_{W_1}\mathcal{L}(W_1^{(k)}, W_2^{(k)}, H^{(k)}),$$

and analogously for $W_2^{(k)}$ and $H^{(k)}$. But, by setting $k = \eta t$ and dividing by η, we get,

$$\lim_{\eta \to 0} \frac{W_1(t+\eta) - W_1(t)}{\eta} = \frac{dW_1}{dt},$$

Therefore,

$$\frac{dW_1}{dt} = -\nabla_{W_1}\mathcal{L}(W_1(t), W_2(t), H(t)).$$

Gradient flow for Deep NMF. Therefore, we get:

$$\begin{cases}
\dfrac{dW_1}{dt} = -(W_1 W_2 H - X)(W_2 H)^\top, \\
\dfrac{dW_2}{dt} = -W_1^\top (W_1 W_2 H - X) H^\top, \\
\dfrac{dH}{dt} = -W_2^\top W_1^\top (W_1 W_2 H - X).
\end{cases}$$

3.2 Dynamics of the singular values at the beginning of training

We also propose a qualitative analysis regarding the trajectories of the singular values of $M = W_1W_2H$ at the beginning of the gradient descent.Precisely in this subsection, we empirically set W_1, W_2, and H to be uniformly initialized in $[0, 10^{-2}]$ such that $W_1W_2H(0) = O(10^{-6}r_1r_2)$ entry-wise, ensuring that $W_1W_2H(0) \ll X = O(0.1)$ in terms of magnitude. Moreover, we will discuss a vanilla gradient descent instead of one optimized through Adam.

Please note that this subsection has not been scrutinized by independent peers therefore it might lack details and sometimes rigor, but we believe it still offers an interesting qualitative view of the phenomenon and a formal proof might arise from it.

Step 1 – Linearization of the gradient flow We consider the gradient flow equations:

$$\dot{W}_1 = -(M - X)(W_2H)^\top, \qquad \dot{W}_2 = -W_1^\top(M - X)H^\top, \qquad \dot{H} = -W_2^\top W_1^\top(M - X),$$

where $M(t) := W_1(t)W_2(t)H(t)$.

Using the Leibniz rule, the derivative of $t \mapsto M(t)$ is:

$$\dot{M} = \dot{W}_1W_2H + W_1\dot{W}_2H + W_1W_2\dot{H}.$$

Substituting the derivatives, we obtain:

$$\dot{M} = -(M - X)(W_2H)^\top W_2H - W_1W_1^\top(M - X)H^\top H - W_1W_2W_2^\top W_1^\top(M - X).$$

where the initial dynamics matrices are defined as:

- $A(0) = (W_2H)^\top(W_2H)$
- $B(0) = W_1W_1^\top$
- $C(0) = H^\top H$
- $D(0) = W_1W_2(W_1W_2)^\top$

Step 2 – Approximation using small initialization Because we initialize in $[0, 0.01]$, by continuity of the coefficients of M, for small enough t, we can neglect the $ReLU$ operator. The initial reconstruction error is dominated by the ground-truth matrix, such that $M(0) - X \approx -X$. Under this assumption, the gradient flow at $t = 0$ simplifies to:

$$\dot{M}(0) \approx XA(0) + B(0)XC(0) + D(0)X$$

where $A(0), B(0), C(0)$, and $D(0)$ are the weight-dependent matrices defined previously. Using the first-order Taylor expansion of $M(t)$ near $t = 0$, we have:

$$M(t) = M(0) + t\dot{M}(0) + o(t)$$

Since $M(0)$ is negligible compared to the magnitude of the update for small t, we can write:

$$\boxed{M(t) \approx t\Big(XA(0) + B(0)XC(0) + D(0)X\Big) + o(t)}$$

This expression shows that the early-stage evolution of M is driven by a linear combination of transformations of X. Specifically, each term "filters" the ground-truth matrix X through the initial random weights. Because this evolution is directly proportional to X, the directions corresponding to the largest singular values of X receive the strongest initial signal, providing a theoretical basis for the observed low-rank bias.

Step 3 – Link to the singular values In particular, we denote by $(\sigma_i(X))_{i=1}^r$ and $(\sigma_i(M(t)))_{i=1}^r$ the singular values of X and $M(t)$, respectively, ordered in decreasing order.

$$X = \sum_{i=1}^{r} \sigma_i(X)\, u_i v_i^\top, \qquad u_i^\top u_j = v_i^\top v_j = \delta_{ij}, \qquad \sigma_1(X) > \sigma_2(X) \geq \cdots \geq \sigma_r(X) > 0.$$

Thus,

$$\boxed{M(t) \approx \sum_{i=1}^{r} (t\sigma_i)\Big[u_i(A(0)^\top v_i)^\top + (B(0)u_i)(C(0)v_i)^\top + (D(0)u_i)v_i^\top\Big]}$$

We can thus write more concisely,

$$M(t) \approx t \sum_{i=1}^{r} \sigma_i(X)\Gamma_i(0) + o(t) \tag{5}$$

where each $\Gamma_i(0)$ represents the "filtered" contribution of the i-th singular component of X through the initial weights:

$$\Gamma_i(0) := u_i(A(0)^\top v_i)^\top + (B(0)u_i)(C(0)v_i)^\top + (D(0)u_i)v_i^\top \tag{6}$$

Step 4 – Statistical link between singular values To establish a precise link between $\sigma_k(M(t))$ and $\sigma_k(X)$, we consider the behavior of the directional matrices $\Gamma_i(0)$. Because of the random initialization, the expected projection of $\Gamma_i(0)$ onto the original singular directions (u_i, v_i) becomes a constant factor.

Due to the random uniform initialization of the weights, and the fact that $A(0), B(0), C(0), D(0)$ are PSD, we assume:

$$\mathbb{E}[u_i^\top \Gamma_i(0) v_j] \approx \delta_{i,j}\bar{\alpha}$$

where $\bar{\alpha} > 0$ is a scalar depending on the variance of the initial weights. Specifically, in expectation:

$$\mathbb{E}[\sigma_k(M(t))] \approx t \cdot \sigma_k(X) \cdot \bar{\alpha} + o(t) \tag{7}$$

To formally link the singular values of $M(t)$ and of X, we use another definition of the largest singular value:

$$\sigma_1(M(t)) = \sup_{\|u\|=1, \|v\|=1} u^\top M(t) v \tag{8}$$

By choosing the specific test vectors $u = u_1$ and $v = v_1$ (the singular vectors associated with $\sigma_1(X)$), we establish a lower bound for the growth of the first singular value, reusing the expression of $M(t)$ written before and the fact that the u_i and v_i are orthonormal and using the properties on the supremum on the expectation:

$$\mathbb{E}[\sigma_1(M(t))] \geq \sup_{\substack{\|u\|=1 \\ \|v\|=1}} \mathbb{E}[u^\top M(t) v] \geq \mathbb{E}[u_1^\top M(t) v_1] \approx t \cdot \sigma_1(X) \cdot \delta_{1,1} \cdot \bar{\alpha} + o(t) \tag{9}$$

This inequality ensures that the magnitude of $\mathbb{E}[\sigma_1(M(t))]$ is at least proportional to $\sigma_1(X)$.

Furthermore, since the sets of singular vectors $\{u_i\}_{i=1}^r$ and $\{v_i\}_{i=1}^r$ are orthonormal, we can extend this reasoning to the subsequent singular values. We use the Courant-Fischer min-max theorem applied to the SVD:

$$\sigma_k(M) = \max_{\substack{\mathcal{V} \subseteq \mathbb{R}^n \\ \dim(\mathcal{V}) = k}} \min_{\substack{v \in \mathcal{V} \\ \|v\|=1}} \|Mv\|_2 \tag{10}$$

Using that theorem, $\sigma_k(M(t))$ is defined by optimizing over the subspace orthogonal to the first $k-1$ singular vectors. Because of this orthogonality:

1. The contribution of $\sigma_1(X)$ vanishes when we project $M(t)$ onto the subspace spanned by $\{u_k, v_k\}$ for $k > 1$, since $u_k^\top u_1 = 0$ and $v_k^\top v_1 = 0$.
2. Consequently, $\sigma_k(M(t))$ depends primarily on $\sigma_k(X)$ and is "protected" from the dominance of $\sigma_1(X)$.

This decoupling confirms that the hierarchy $\sigma_1(X) \gg \sigma_2(X) \ldots$ is faithfully transmitted in expectation to $M(t)$ at $t \approx 0$. The first singular value of $M(t)$ grows the fastest (in expectation), while the subsequent ones remain small and independent of $\sigma_1(X)$, effectively maintaining $M(t)$ in a low-rank state during the initial phase of gradient descent. This theoretical prediction is consistent with our experimental observations, where the singular values of $M(t)$ emerge sequentially according to their importance in X (cf. Figure 1).

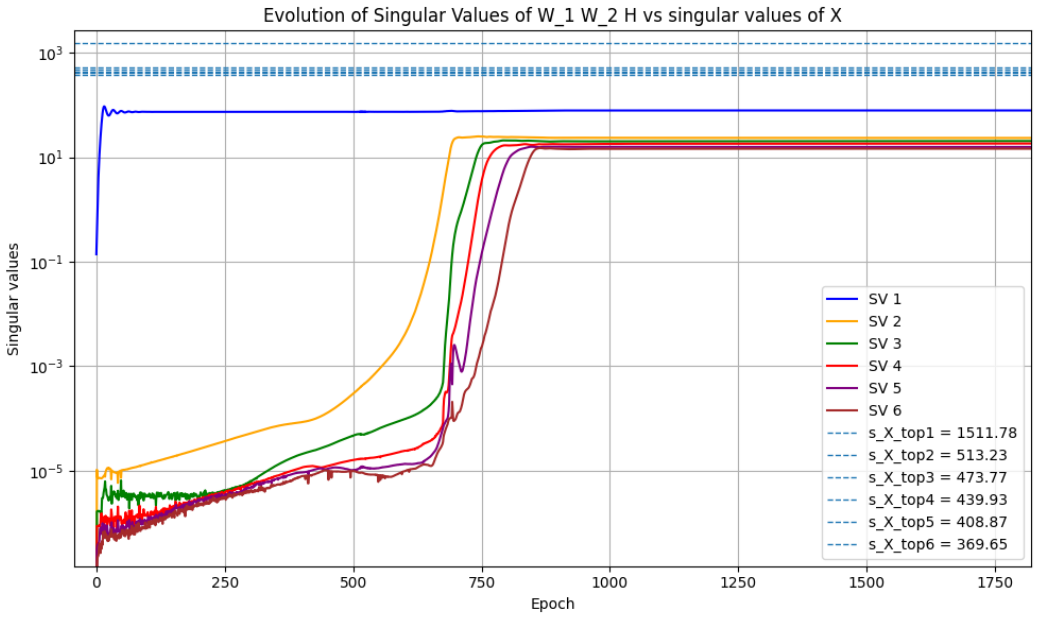


Figure 1: Evolution of the 6 first singular values of M compared to the singular values of X.

Step 5 – Link to the effective rank We now come back to the definition of the effective rank (see Section 2.5):

$$ER(M(t)) = \frac{\sigma_1(M(t))}{\sum_i \sigma_i(M(t))} \tag{11}$$

Recall our results lied on $\mathbb{E}[\sigma_i(M(t))]$. Because in general for two random variables Y, Z, $\mathbb{E}[Y/Z] \neq \mathbb{E}[Y]/\mathbb{E}[Z]$, our results do not extend that easily to the effective rank. However, because our matrices are bounded coefficient-wise, and each coefficients are chosen independently, they are sub-Gaussian (as seen in the lecture notes). Therefore, it appears feasible to use concentration inequalities to explicitly estimate $\mathbb{E}[ER(M(t))]$. Thus, we have provided a theoretical explanation (although qualitative, relying on heavy assumptions and incomplete) of the low rank bias.

4 Empirical results and observations

4.1 Qualitative dynamics of archetypes and effective rank during training

Beyond reconstruction error, a key appeal of Nonnegative Matrix Factorization (NMF) and its deep variants lies in the interpretability of the learned representations. In this section, we adopt an empirical and exploratory perspective to investigate how the low-rank bias observed during gradient-based training manifests itself in the evolution of interpretable archetypes.

Rather than focusing on a formal theoretical explanation (discussed in 3.1 and partially undertook in 3.2), we examine whether changes in the effective rank of the reconstruction correlate with qualitative changes in the learned features.

Experimental setup

We consider Deep NMF with the following objective:

$$(W_1, \dots, W_k, H) = \underset{\substack{W_i \in M_{r_{i-1}, r_i}(\mathbb{R}),\ i=1,\dots,k,\\ H \in M_{r_k, m}(\mathbb{R})\\ W_i \geq 0,\ H \geq 0}}{\arg\min} \frac{1}{2} \left\| X - \left(\prod_{i=1}^{k} W_i \right) H \right\|_F^2 .$$

As detailed in 2, the rows of H correspond to archetypal images, while the matrices W_i encode how these archetypes are combined to reconstruct the data.

During training, we periodically record the reconstructed matrix $(\prod_{i=1}^{k} W_i)H$, its effective rank computed from the singular value spectrum (cf. 2.5), and the corresponding archetypal images. These quantities are visualized at fixed intervals to track their evolution over training.

Motivation

Prior work highlights an implicit low-rank bias in matrix factorization models trained by gradient-based methods, whereby dominant low-rank structures are learned first and higher-rank components emerge later. While this phenomenon is typically analyzed through spectral quantities, its connection to the semantic structure of the learned representations remains unclear.

In the context of NMF, where features are directly interpretable, this naturally raises the question of whether transitions in effective rank correspond to the emergence of qualitatively new or more refined archetypal features. This section addresses this question empirically.

Qualitative evolution of archetypes

Figure 5 (in Appendix B) shows the evolution of the archetypal images (rows of H) during training, to be compared with the effective rank trajectory reported in Figure 9, which exhibits a clear plateau structure with two marked jumps around 12,000 and 14,000 epochs.

Before the first jump, the archetypes evolve from unstructured noise to simple and highly redundant patterns. In particular, several archetypes take the form of roughly circular shapes reminiscent of the digit *0*, while a significant fraction of the components remain dominated by noise, with no clear visual structure. At this stage, the learned representation appears limited to a small number of coarse, low-rank patterns.

Around the first effective-rank jump (approximately 12,000 epochs, see Figure 5h), two distinct qualitative changes are observed. First, a new archetype emerges from a previously noise-dominated component, becoming visually structured. Second, one of the redundant circular archetypes undergoes a sharp transformation and becomes recognizable as the digit *6*. These changes occur abruptly and coincide temporally with the increase in effective rank.

After the second jump (around 14,000 epochs, see Figure 5i), no further denoising effects are observed: all but one archetype are already well-structured, and the remaining noise-dominated component remains largely unchanged. However, a final qualitative transition occurs, as the last redundant circular archetype is suddenly transformed into a distinct digit, namely *2*. Beyond this point, the archetypes remain visually stable.

Overall, these observations indicate that increases in effective rank are associated not with gradual refinements, but with discrete and interpretable qualitative transitions in the learned archetypes, such as the emergence of new components or the resolution of redundant patterns. Note that those findings are consistent across different hyperparameter combinations (size of the dataset, dimensions of H, learning rate etc.)

4.2 Comparison of different losses

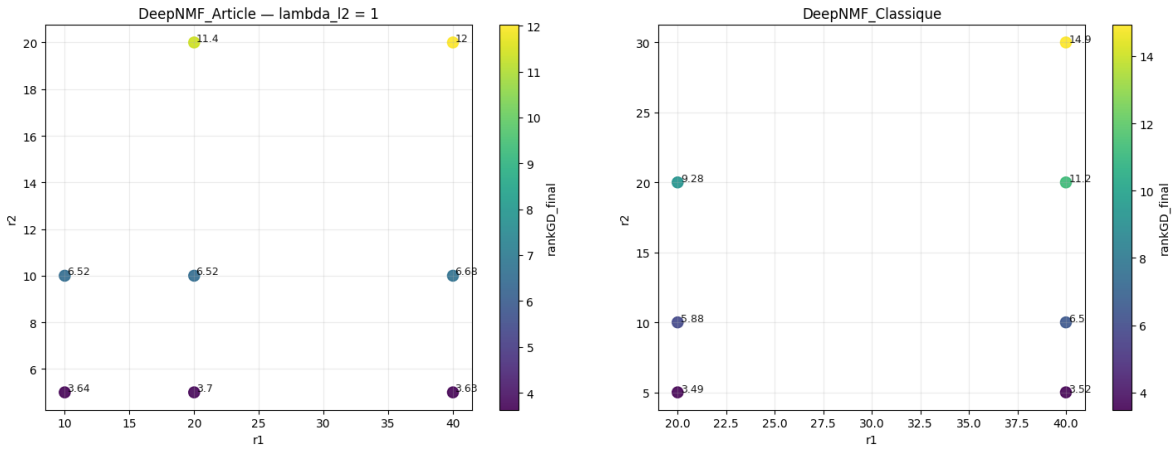
In our experiments, we consider two different training objectives for Deep NMF, which differ in the way reconstruction errors are enforced across layers.

Global reconstruction loss (classical formulation). The first approach corresponds to the classical end-to-end formulation commonly used in Deep NMF. Given a data matrix $A \in \mathbb{R}_+^{m \times n}$, and the decomposition matrices $W_1 \in \mathbb{R}_+^{m \times r_1}$, $W_2 \in \mathbb{R}_+^{r_1 \times r_2}$, and $H \in \mathbb{R}_+^{r_2 \times n}$. The parameters are learned by minimizing a single global reconstruction loss:

$$\mathcal{L}_{\text{global}} = \|A - W_1 W_2 H\|_F^2$$

where $\|\cdot\|_F$ denotes the Frobenius norm. All factors are optimized jointly using gradient-based methods under non-negativity constraints. This formulation does not explicitly constrain the quality of intermediate representations and allows the different layers to compensate for each other as long as the final reconstruction error is minimized.

Layer-wise reconstruction loss (proposed in [2]). The second approach follows the layer-wise loss formulation introduced by De Handschutter and al. [2]. In this setting, an explicit



(a) Layer-wise loss.

(b) Classical global loss.

Figure 2: Final effective rank obtained for different choices of (r_1, r_2). Each point corresponds to a trained Deep NMF model and is colored according to the final effective rank. We used both lambdas equal to 1. *Note: the two scales aren't identical.*

intermediate representation $H_{\text{mid}} \in \mathbb{R}_+^{r_1 \times n}$ is introduced, and the deep factorization is defined through the successive approximations:

$$A \approx W_1 H_{\text{mid}}, \qquad H_{\text{mid}} \approx W_2 H.$$

The corresponding loss function is given by:

$$\mathcal{L}_{\text{layer}} = \lambda_1 \|A - W_1 H_{\text{mid}}\|_F^2 + \lambda_2 \|H_{\text{mid}} - W_2 H\|_F^2,$$

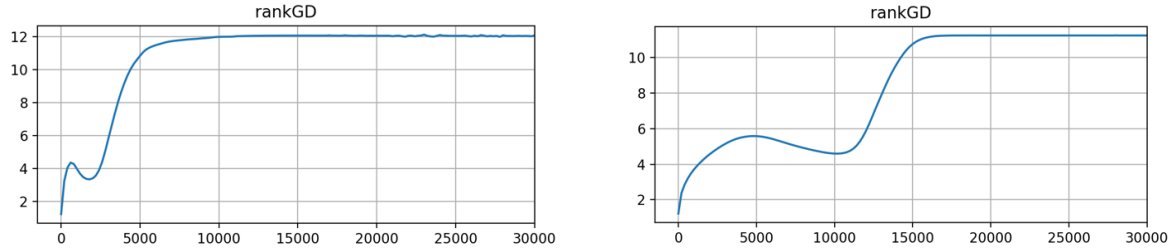
where λ_1 and λ_2 balance the contribution of each reconstruction term. By explicitly enforcing reconstruction quality at each layer, this formulation reduces the degrees of freedom of the deep model and promotes structured intermediate representations.

Results from our experiments

All our results are computed considering the same subset of images from MNIST (2000 images), the same seed and the same number of epochs (40000). It enables us to have reproducible but more essentially comparable results.

Comparison of effective rank across loss formulations. To analyze how different loss formulations impact the degrees of freedom exploited by Deep NMF models, we evaluate the effective rank (ER) of the reconstructed matrix $W_1 W_2 H$. Figure 2 reports the effective rank obtained for different choices of (r_1, r_2). Be aware that the scales differ between the two plots.

Across all configurations, the layer-wise loss consistently yields higher effective ranks than the classical global loss. This indicates that enforcing reconstruction constraints at



(a) Layer-wise loss case.

(b) Classical global loss case.

Figure 3: Evolution on the effective rank across epochs for both losses for $r_1 = 40, r_2 = 20$.

intermediate layers prevents the model from collapsing to overly low-rank solutions and encourages a more distributed use of the available spectral directions. However, the dominant factor governing the effective rank remains the model dimensions, in particular $\min(r_1, r_2)$. As $\min(r_1, r_2)$ increases, the effective rank increases markedly under both loss formulations, while the difference induced by the choice of loss remains secondary in comparison.

Figure 3 highlights a marked difference in the evolution of the effective rank during training under the two loss formulations. With the layer-wise loss, the effective rank increases rapidly and smoothly, reaching its final value early in the optimization process. In contrast, the classical global loss exhibits a pronounced low-rank regime: the effective rank initially rises to a small value, then remains nearly constant for a large number of iterations before undergoing a delayed transition toward a higher-rank solution. This behavior suggests that the global loss is strongly biased toward low-rank solutions during the early and intermediate stages of training. By enforcing reconstruction constraints at intermediate layers, the layer-wise loss avoids this low-rank collapse and promotes a more balanced and progressive use of the model's spectral capacity.

Comparison of Error across loss formulations.

To reduce the impact of the subset on the error computed, we consider the error normalized by the Frobenius norm of the subset of MNIST.

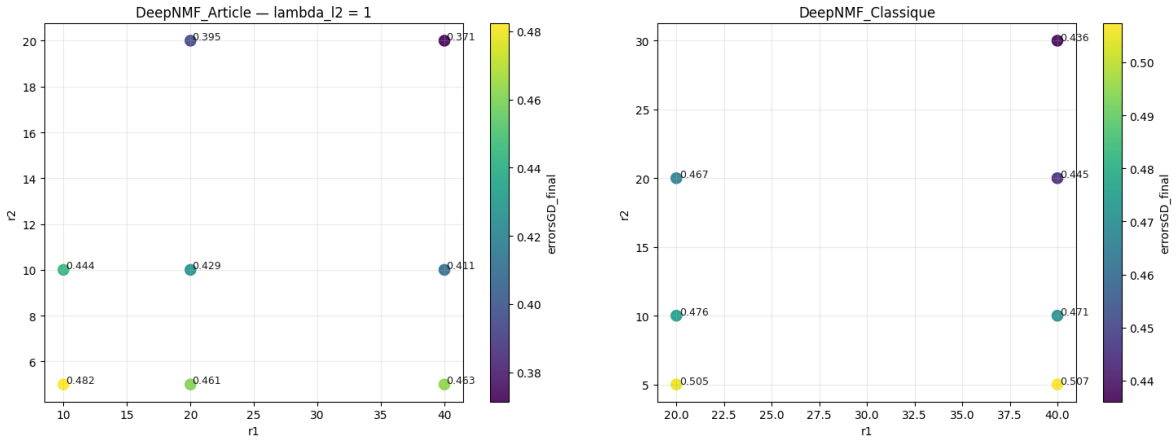
$$\mathrm{Err}_{\mathrm{A}}(W_1 W_2 H) = \frac{\|A - W_1 W_2 H\|_F^2}{\|A\|_F^2}$$

Figure 4 highlights the improvement in relative reconstruction error achieved by the layer-wise loss compared to the classical formulation. These observations suggest that the layer-wise loss enables the model to make a more effective use of its available capacity.

These empirical findings are consistent with the inconsistency issues highlighted in [2], and illustrate how explicitly constraining intermediate representations can mitigate the implicit low-rank bias of global loss formulations.

While our analysis does not aim at establishing theoretical guarantees, it provides practical insights into the interplay between loss design, optimization dynamics, and effective model capacity in Deep NMF.

Since this result is not directly related to the low-rank bias, it is therefore deferred to Appendix E. Nevertheless, it provides an estimate of the reconstruction error induced by the non-negativity constraints on the factorized matrices.



(a) Layer-wise loss.

(b) Classical global loss.

Figure 4: Final normalized error obtained for different choices of (r_1, r_2). Each point corresponds to a trained Deep NMF model and is colored according to the normalized error. We used both lambdas equal to 1. *Note: the two scales aren't identical.*

References

[1] Hung-Hsu Chou et al. “Gradient Descent for Deep Matrix Factorization: Dynamics and Implicit Bias towards Low Rank”. In: *Applied and Computational Harmonic Analysis* 68.101595 (2024). DOI: 10.1016/j.acha.2023.101595.

[2] Pierre De Handschutter and Nicolas Gillis. “A Consistent and Flexible Framework for Deep Matrix Factorizations”. In: *Pattern Recognition* 134.109102 (Feb. 2023). DOI: 10.1016/j.patcog.2022.109102.

[3] ZY Zhang. *Nonnegative Matrix Factorization: Models, Algorithms and Applications*. Vol. 24. Springer, Berlin, 2012. DOI: 10.1007/978-3-642-23241-1_6.

[4] Wen-Sheng Chen, Qianwen Zeng, and Binbin Pan. “A survey of deep nonnegative matrix factorization”. In: *Neurocomputing* 491 (June 2022), pp. 305–320. DOI: 10.1016/j.neucom.2021.08.152.

[5] Daniel D. Lee and H. Sebastian Seung. “Algorithms for Non-negative Matrix Factorization”. In: *Advances in Neural Information Processing Systems* 13 (2000).

[6] D. P. Bertsekas. “On the Goldstein-Levitin-Polyak gradient projection method”. In: *IEEE Transactions on Automatic Control* 21.2 (Apr. 1976), pp. 174–184. DOI: 10.1109/TAC.1976.1101194.

[7] NVIDIA. *CUDA Toolkit*. URL: https://developer.nvidia.com/cuda-toolkit.

[8] Boutsidis and Gallopoulos. “SVD based initialization: A head start for nonnegative matrix factorization”. In: *Pattern Recognition* 41.4 (Apr. 2008), pp. 1350–1362. DOI: 10.1016/j.patcog.2007.09.010.

[9] scikit-learn. *NMF*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html.

[10] Olivier Roy and Martin Vetterli. “The effective rank: A measure of effective dimensionality”. In: *15th European Signal Processing Conference, Poznan, Poland* (2007), pp. 606–610.

A The Adam optimizer

Given a parameter vector θ, gradients $g_t = \nabla_\theta f_t(\theta)$, and hyperparameters α (stepsize), β_1, β_2, and ϵ; parameters are updated as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{12}$$

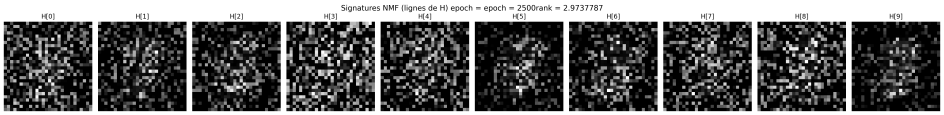
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{13}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{14}$$

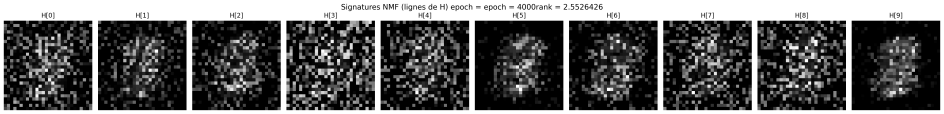
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{15}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \tag{16}$$

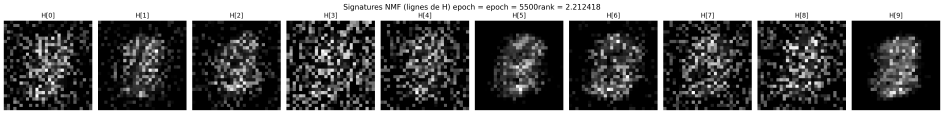
B Archetypal images of H



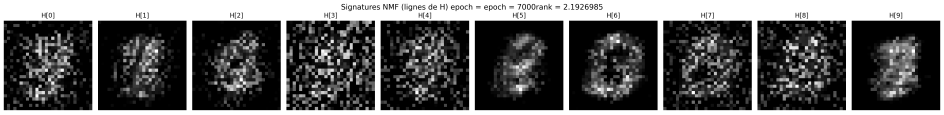
(a) epoch=2500



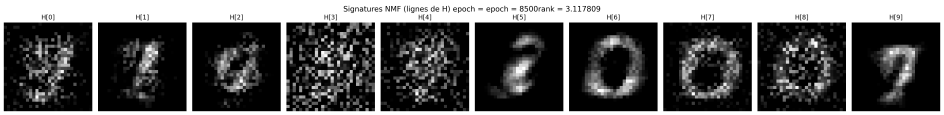
(b) epoch=4000



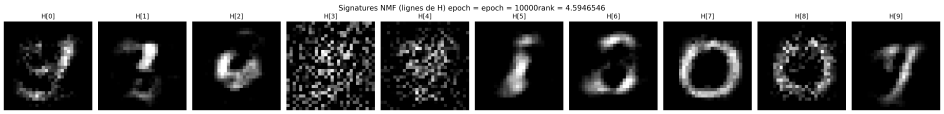
(c) epoch=5500



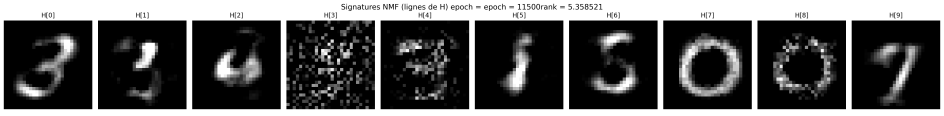
(d) epoch=7000



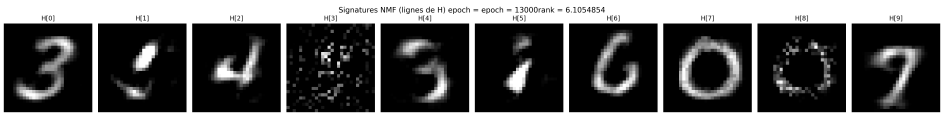
(e) epoch=8500



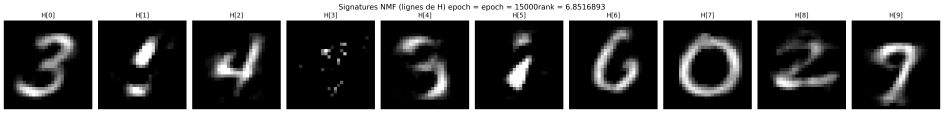
(f) epoch=10000



(g) epoch=11500



(h) epoch=13000

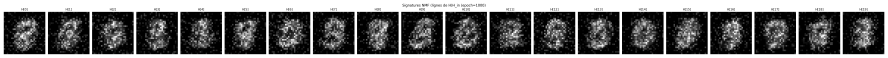


(i) epoch=15000

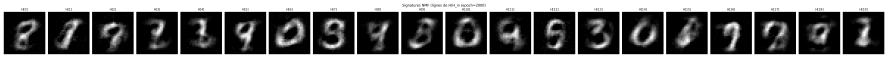
Figure 5: Evolution of the archetypal images of H for DeepNMF with $k = 2$.

C Archetypal images for layer loss, $r_1 = 20$ and $r_2 = 10$

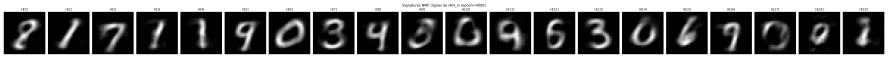
(a) epoch=0



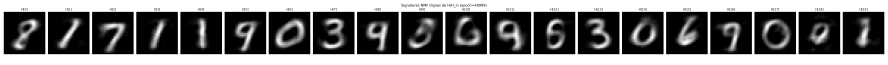
(b) epoch=1000



(c) epoch=2000

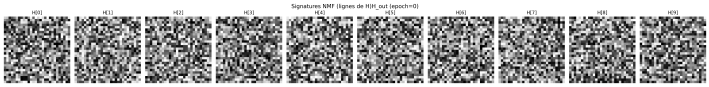


(d) epoch=4000

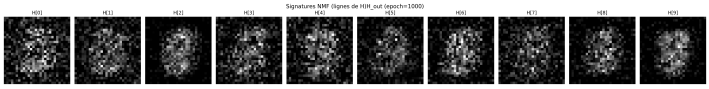


(e) epoch=49999

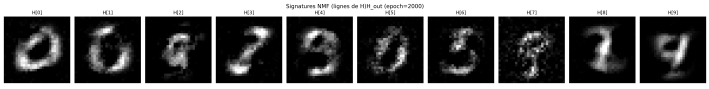
Figure 6: Evolution of the archetypal images of H_{mid} for DeepNMF with Layer loss.



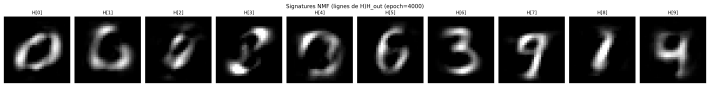
(a) epoch=0



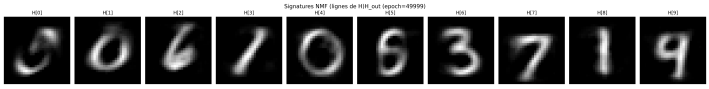
(b) epoch=1000



(c) epoch=2000



(d) epoch=4000



(e) epoch=49999

Figure 7: Evolution of the archetypal images of H for DeepNMF with Layer loss.

D Metrics evolution

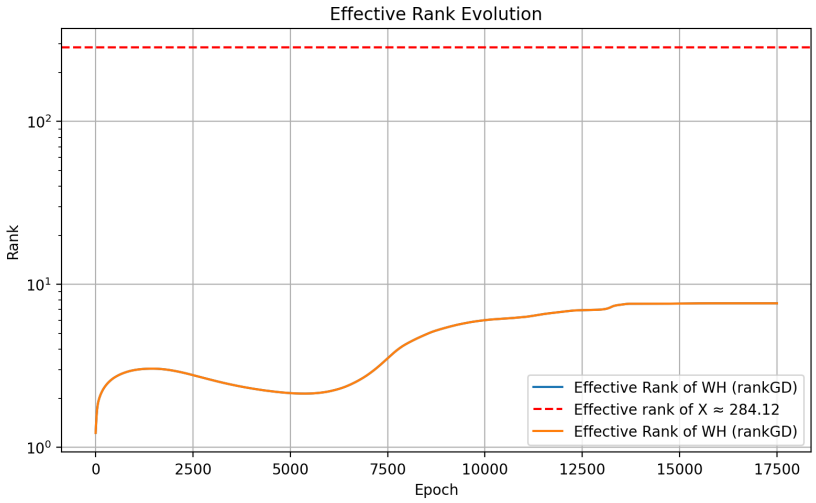


Figure 8: Evolution of the effective rank during training (log-scale) compared to the effective rank of the ground-truth matrix X.
Note: The effective rank is bounded by the dimensions of the matrices, here by 10. The log-scale therefore hides that the variations of the effective rank observed are very significant.

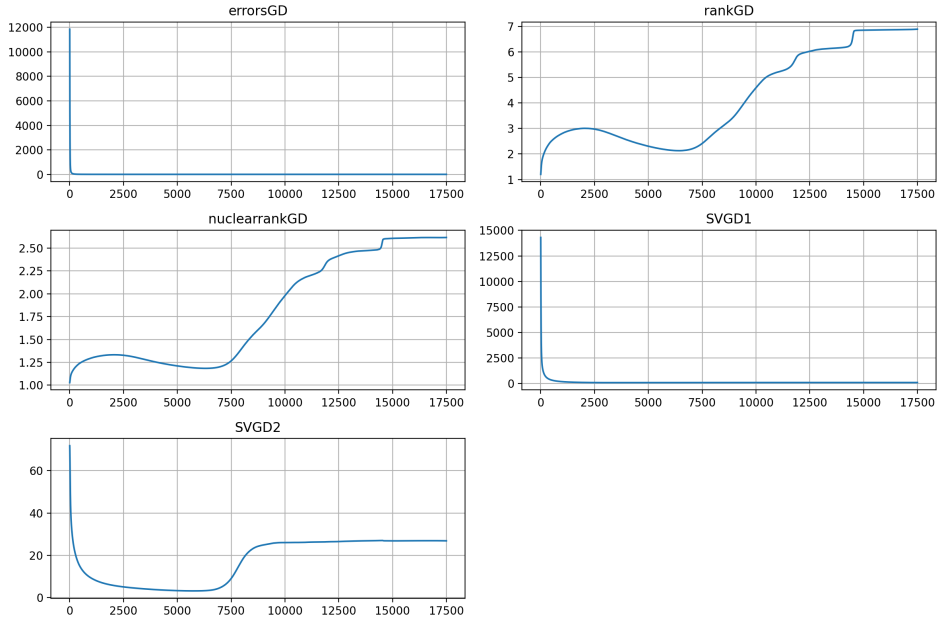


Figure 9: Evolution of the effective rank (rankGD subplot) along with other metrics, including the loss (errorGD), the nuclear rank (nuclearrankGD), and the norm of the two largest singular values (resp. SVGD1 and SVGD2).
Note: Initialization here impacted SVGD1, but its behavior is alike SVGD2 in the long-run.

E Bound for the error

Since the reconstructed matrix is given by a deep factorization of the form:

$$\hat{A} = W_1 W_2 H,$$

its rank is upper bounded by the smallest intermediate dimension, that is,

$$\operatorname{rank}(\hat{A}) \leq \min(r_1, r_2).$$

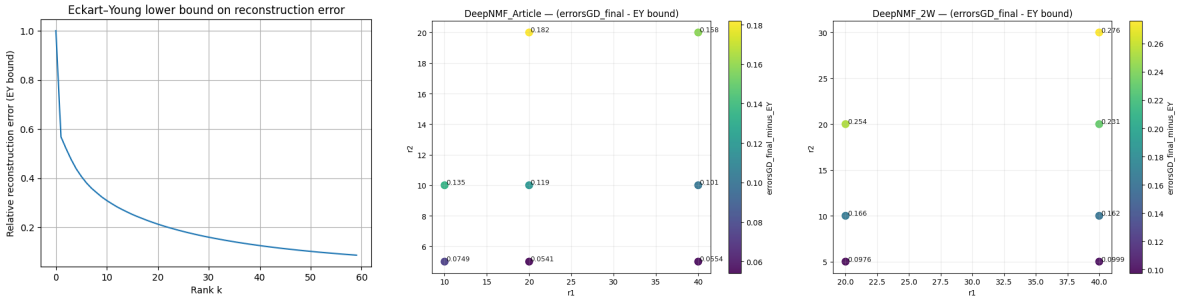
As a consequence, the reconstruction error of our model is lower bounded by the minimum achievable error among all matrices of rank at most $\min(r_1, r_2)$. Formally, for the relative Frobenius reconstruction error, we have:

$$\frac{\|A - \hat{A}\|_F^2}{\|A\|_F^2} \geq \min_{\operatorname{rank}(X) \leq \min(r_1, r_2)} \frac{\|A - X\|_F^2}{\|A\|_F^2}.$$

By the Eckart–Young–Mirsky theorem, this minimum is achieved by the truncated singular value decomposition of A and is given by the normalized tail energy of its singular value spectrum:

$$\min_{\operatorname{rank}(X) \leq k} \frac{\|A - X\|_F^2}{\|A\|_F^2} = \frac{\sum_{i>k} \sigma_i(A)^2}{\sum_i \sigma_i(A)^2}, \qquad k = \min(r_1, r_2),$$

where $\sigma_i(A)$ denotes the singular values of A. This quantity therefore provides a natural lower bound on the reconstruction error achievable by our deep non-negative matrix factorization model.



(a) Eckart–Young lower bound (b) DeepNMF_Article: $\text{errorsGD}_{\text{final}} - \text{EY}$ (c) DeepNMF_2W: $\text{errorsGD}_{\text{final}} - \text{EY}$

Figure 10: Comparison with the Eckart–Young lower bound. Reconstruction error gap $\text{errorsGD}_{\text{final}} - \text{EY}(k)$ with $k = \min(r_1, r_2)$.