AIR-NET: ADAPTIVE AND IMPLICIT REGULARIZATION NEURAL NETWORK FOR MATRIX COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Conventionally, the matrix completion (MC) model aims to recover a matrix from partially observed elements. Accurate recovery necessarily requires a regularization encoding priors of the unknown matrix/signal properly. However, encoding the priors accurately for the complex natural signal is difficult, and even then, the model might not generalize well outside the particular matrix type. This work combines adaptive and implicit low-rank regularization that captures the prior dynamically according to the current recovered matrix. Furthermore, we aim to answer the question: how does adaptive regularization affect implicit regularization? We utilize neural networks to represent Adaptive and Implicit Regularization and named the proposed model *AIR-Net*. Theoretical analyses show that the adaptive part of the AIR-Net enhances implicit regularization. In addition, the adaptive regularizer vanishes at the end, thus can avoid saturation issues. Numerical experiments for various data demonstrate the effectiveness of AIR-Net, especially when the locations of missing elements are not randomly chosen. With complete flexibility to select neural networks for matrix representation, AIR-Net can be extended to solve more general inverse problems.

1 INTRODUCTION

The matrix completion (MC) problem, which aims to recover a matrix $\boldsymbol{X}^* \in \mathbb{R}^{m\times n}$ from its partially observed elements, has arisen in numerous domains, ranging from computer vision (Wen et al., 2012), recommender system (Netflix, 2009), and drug-target interaction (DTI) (Mongia & Majumdar, 2020). This fundamental problem is ill-posed without assumptions on $\boldsymbol{X}^*$ since we have many completions. So it is essential to impose additional information or priors on the unknown matrix/signal.

To describe the prior for natural signal, or restrict the solution in the corresponding space is difficult. Classical methods for MC are mainly based on low-rank, sparsity or piece-wise smoothness assumption (Rudin et al., 1992; Buades et al., 2005; Romano et al., 2014; Dabov et al., 2007). These priors describe simple structural signal well, but may lead to a poor approximation of $\boldsymbol{X}^*$ with complex structures (Radhakrishnan et al., 2021) especially when the observed entries are not sampled uniformly at random. Recently, deep neural networks (DNN) have shown a strong ability in extracting complex structures from large datasets(Li et al., 2018; Mukherjee et al., 2021). However, such a large number of data sets cannot be obtained in many scenarios. Fortunately, DNN also works in solving some inverse problems without any extra training set (Ulyanov et al., 2018). Over-parametric DNN performs well on a single matrix is a mysterious phenomenon. One of the explanations is there exists **implicit regularization** during training (Arora et al., 2019; Xu et al., 2019; Rahaman et al., 2019; Chakrabarty & Maji, 2019). Although DNN with implicit regularization outperforms some classical methods, it is insufficient to describe the space of complex $\boldsymbol{X}^*$. Extra-explicit regularization can improve its performance in signal recovery (Metzler et al., 2018; Boyarski et al., 2019a; Liu et al., 2019; Li et al., 2020). However, such explicit priors are often valid only for specific data or sampling patterns. A more flexible regularization is required to meet practical MC problems.

We introduce flexibility in this paper by firstly representing the explicit regularization using DNN without any extra training set. The explicit regularization we begin with is Dirichlet Energy (DE), which is formulated as $\text{tr}(\boldsymbol{X}^\top \boldsymbol{L}\boldsymbol{X})$, with $\boldsymbol{L}$ a Laplacian matrix describing the similarity between

columns. Note that $\boldsymbol{L}$ in DE is fixed during iteration. Building an exact $\boldsymbol{L}$ based on incomplete observation is very challenging. Therefore, we parameterize $\boldsymbol{L}$ with DNN, and revise $\boldsymbol{L}$ iteratively during training. Furthermore, We combine the learned DE, which is an **adaptive regularizer**, with implicit regularization to form a new regularization method for MC named AIR-Net. The interaction between explicit regularization and implicit regularization in solving MC problems is further studied. The results show that combining the two can obtain a new, more flexible regularization model and enhance the low-rank preference of implicit regularization. In many examples, AIR-NET has a more vital feature representation ability and more comprehensive application range and shows state-of-art performance.

2 ADAPTIVE AND IMPLICIT REGULARIZATION NEURAL NETWORK

Our model is proposed as follow:

$$\min_{\boldsymbol{X},\boldsymbol{W}_i} \mathcal{L}_{all} = \mathcal{L}_{\mathbb{Y}}\left(\mathcal{A}\left(\boldsymbol{X}^*\right), \mathcal{A}(\boldsymbol{X})\right) + \sum_{i=1}^{N} \lambda_i \cdot \mathcal{R}_{\boldsymbol{W}_i}\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\right) \tag{1}$$

where $\mathcal{A}\left(\boldsymbol{X}\right) = \boldsymbol{X} \mid_{\Omega} = \begin{cases} \boldsymbol{X}_{i,j}, & (i,j) \in \Omega \\ 0, & (i,j) \notin \Omega \end{cases}$ and Ω are the observed coordinates set, and the other entries are missing. Different from other regularization models for MC, here $\boldsymbol{X}$ is represented by a neural network which tends to be low-rank implicitly (Section 2.1), and $\mathcal{R}_{\boldsymbol{W}_i}$ is an adaptive regularization with a forward neural network represented Laplacian matrix(Section 2.2). The detailed notations will be introduced in the corresponding sections. A specific case of Equation 1 for matrix completion is given in Section 2.3.

2.1 DMF AS AN IMPLICIT REGULARIZATION

In order to make the model suitable for more matrix types, we need a more general data prior. The low-rank is a very general prior in various matrix types. There are two main ways to encode the low-rank prior into model: (a) Adding an explicit regularization term such as rank and nuclear norm (Candès & Recht, 2009; Lin et al., 2010). (b) Using a low-dimensional latent variable model to represent $\boldsymbol{X}$, including matrix factorization (MF) and its varieties (Koren et al., 2009; Fan & Cheng, 2018). The first case suffers from the saturation issue, which is induced by explicit regularization. The second one faces the problem of estimating a proper latent variable dimension.

Unlike the existing MF model, which constricts the size of the shared dimension of the factorized matrix, DMF can take a large shared dimension and still preserve the low-rank property without explicit regularization. This is the so-called implicit low-rank regularization of DMF:

$$\boldsymbol{X}(t) = \boldsymbol{W}^{[L-1]}(t)\boldsymbol{W}^{[L-2]}(t)\ldots\boldsymbol{W}^{[1]}(t)\boldsymbol{W}^{[0]}(t) \in \mathbb{R}^{m\times n},$$

where L is the depth of MF. $\boldsymbol{W}^{[l]}(t)$ represents the l-th matrix at the step t during training. The results are given under a mild assumption 1 in Section A.2. This property helps avoiding dimension estimation and saturation issues. As for the details of the implicit low-rank we will discuss in Section A.2.

2.2 ADAPTIVE REGULARIZER

Apart from the low-rank prior, self-similarity is also a typical prior. The patch in the image and the rating behavior of users are all examples of self-similarity. For example, there is always a certain degree of self-similarity between the blocks in the image. A classical way to encode the similarity prior to $\boldsymbol{X}$ is Dirichlet Energy (DE) which is formulated as $\mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{L}\boldsymbol{X})$. But DE will face two problems in applications: (a) $\boldsymbol{L}$ is unknown in MC problem, construct $\boldsymbol{L}$ based on incomplete $\boldsymbol{X}$ may induce worse prior. (b) The formulation of DE only encodes the similarity of the columns of $\boldsymbol{X}$. Other similarities such as block similarity cannot be captured. To address both of these issues, we parameterize $\boldsymbol{L}$ with DNN and replace $\boldsymbol{X}$ by a transformed $\mathcal{T}_i\left(\boldsymbol{X}\right)$ to capture the self-similarity flexibly.

The adaptive regularization is defined as

$$\mathcal{R}_{\boldsymbol{W}_i}\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\right) = \mathrm{tr}\left(\mathcal{T}_i\left(\boldsymbol{X}\right)^\top \boldsymbol{L}_i\left(\boldsymbol{W}_i\right)\mathcal{T}_i\left(\boldsymbol{X}\right)\right), i = 1, 2, \ldots, N$$

where $\boldsymbol{L}_i \in \mathbb{R}^{m_i \times m_i}$ is parameterized by $\boldsymbol{W}_i \in \mathbb{R}^{m_i \times m_i}$. To keep the Laplacian properties of $\boldsymbol{L}_i$, special design for the parameterized structure is important. We design a forward neural network which encodes the properties of Laplacian matrix in structure. The details are discussed in A.4. $\mathcal{T}_i : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{m_i \times n_i}$ transforms $\boldsymbol{X}$ into special domain, which makes the AIR-Net possible to capture various relationships embedded in data. The common choice can be $\mathcal{T}_i(\boldsymbol{X}) = \boldsymbol{X}$ which captures the relationship between columns. Regularization captures the relationship between rows when $\mathcal{T}_i(\boldsymbol{X}) = \boldsymbol{X}^\top$. Especially if $\mathcal{T}_i(\boldsymbol{X}) = \left[\textbf{vec}\left(\text{block}(\boldsymbol{X})\right)_1, \textbf{vec}\left(\text{block}(\boldsymbol{X})\right)_2, \ldots, \textbf{vec}\left(\text{block}(\boldsymbol{X})\right)_{n_i}\right]$, where $\textbf{vec}\left(\text{block}(\boldsymbol{X})\right)_j \in \mathbb{R}^{m_i}$ is the vectorization of j-th block in $\boldsymbol{X}$ row by row, then the similarity among blocks can be obtained. A natural problem that arises is what the $\boldsymbol{L}_i$ looks like during training.

Obviously, $\mathcal{R}_{\boldsymbol{W}_i}$ reaches minimum when $\boldsymbol{L}_i = 0$, and this is called a trivial solution. The most exciting thing is that when we minimize Equation 1 with the gradient descent algorithm, $\{\boldsymbol{L}_i(\boldsymbol{W}_i(t))\}$ converges to a non-trivial solution. Another expected phenomenon is that $\mathcal{R}_{\boldsymbol{W}_i}$ vanishes at the end and will not cause the so-called saturation issue. The saturation issue is a bias term that dominates the overall estimation error due to explicit regularization. We illustrate these phenomena both by theoretical analysis (Theorem 2 in Section 3.2) and numerical experiments (Section 4).

2.3 AIR-NET FOR MC

In this subsection, we will focus our model on the MC problem. We select $\mathcal{T}_1(\boldsymbol{X}) = \boldsymbol{X}, \mathcal{T}_2(\boldsymbol{X}) = \boldsymbol{X}^\top, N = 2$ to capture the relationship both in rows and columns of $\boldsymbol{X}$. Overall, the theoretical analyses for general inverse problem is based on Equation 2.

$$\min_{\boldsymbol{X}, \boldsymbol{W_r}, \boldsymbol{W_c}} \mathcal{L}_{all} = \mathcal{L}_{\mathbb{Y}}\left(\mathcal{A}\left(\boldsymbol{X}^*\right), \mathcal{A}\left(\boldsymbol{X}\right)\right) + \lambda_r \cdot \mathcal{R}_{\boldsymbol{W}_r}\left(\boldsymbol{X}\right) + \lambda_c \cdot \mathcal{R}_{\boldsymbol{W}_c}\left(\boldsymbol{X}^\top\right), \tag{2}$$

where $\boldsymbol{X} = \boldsymbol{W}^{[L-1]}\boldsymbol{W}^{[L-2]}\ldots\boldsymbol{W}^{[1]}\boldsymbol{W}^{[0]}$, $\mathcal{R}_{\boldsymbol{W}_r}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{L}_r(\boldsymbol{W}_r)\boldsymbol{X}^\top$, $\mathcal{R}_{\boldsymbol{W}_c}\left(\boldsymbol{X}^\top\right) = \boldsymbol{X}^\top\boldsymbol{L}_c(\boldsymbol{W}_c)\boldsymbol{X}$. Specially, our experiments focus on the MC problem which can reform Equation 2 as follows:

$$\min_{\boldsymbol{W}^{[l]}, \boldsymbol{W_r}, \boldsymbol{W_c}} \mathcal{L}_{all} = \sum_{(i,j)\in\Omega} \left|\boldsymbol{X}_{ij} - \boldsymbol{X}^*_{ij}\right| + \lambda_r \cdot \mathcal{R}_{\boldsymbol{W}_r}\left(\boldsymbol{X}\right) + \lambda_c \cdot \mathcal{R}_{\boldsymbol{W}_c}\left(\boldsymbol{X}^\top\right), \tag{3}$$

with $l = 0, 1, \cdots, L-1$. The parameters in Equation 3 is updated by gradient descent algorithm or its variations. We stop the iteration until $\left|\mathcal{R}_{\boldsymbol{W}_r(T+1)} - \mathcal{R}_{\boldsymbol{W}_r(T)}\right| < \delta$ and $\left|\mathcal{R}_{\boldsymbol{W}_c(T+1)} - \mathcal{R}_{\boldsymbol{W}_c(T)}\right| < \delta$. The recovered matrix is $\hat{\boldsymbol{X}}(T) = \boldsymbol{W}^{[L-1]}(T)\cdots\boldsymbol{W}^{[0]}(T)$.

Some works which combine implicit and explicit regularization also can be regarded as a special case of Equation 1. Both the Total Variation (TV) and DE can be regarded as a fixed $\boldsymbol{L}$. Therefore, the framework of Equation 1 also contains DMF+TV (Li et al., 2020), DMF+DE (Boyarski et al., 2019a). So far, we cannot see any essential difference between Equation 3 and these models. We will illustrate the amazing properties of the model in the next section.

3 THEORETICAL ANALYSIS

In this section, we will analyze the properties based on the dynamics of Equation 3. Theorem 1 shows that our proposed regularization enhances the implicit low-rank regularization of DMF. Theorem 2 shows that the adaptive regularization will converge to a minimum while capturing the inner structure of data flexibly. Although this paper focus on MC problem, the following theoretical analyzes is satisfied for the general inverse problems. As $\mathcal{A}$ and $\mathcal{A}(\boldsymbol{X}^*)$ are fixed during optimization, we simplify $\mathcal{L}_{\mathbb{Y}}\left(\mathcal{A}\left(\boldsymbol{X}^*\right), \mathcal{A}\left(\boldsymbol{X}\right)\right)$ as $\mathcal{L}_{\mathbb{Y}}\left(\boldsymbol{X}\right)$ below. $\boldsymbol{U}_{i,j}$ is the (i,j) th entry of $\boldsymbol{U}$, $\boldsymbol{U}_{:,k}$ and $\boldsymbol{U}_{k,:}$ are the k-th column and the k-th row of $\boldsymbol{U}$ respectively.

3.1 AIR-NET ENHANCES THE IMPLICIT LOW-RANK REGULARIZATION

To simplify the analysis, we keep $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ fixed. Then the $\mathcal{R}_{\boldsymbol{W}_i}, i = r, c$ only varies with $\boldsymbol{X}$. We will demonstrate what the adaptive regularizer brings to the implicit low-rank regularization.

Theorem 1. *Consider the following dynamics with initial data satisfying the balance initialization Assumption 1(see A.2):*

$$\dot{\boldsymbol{W}}^{[l]}(t) = -\frac{\partial}{\partial \boldsymbol{W}^{[l]}}\mathcal{L}_{all}(\boldsymbol{X}(t)), \quad t \geq 0, \quad l = 0, \ldots, L-1,$$

where $\mathcal{L}_{all}(\boldsymbol{X}) = \mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}) + \lambda_r \cdot \mathcal{R}_{\boldsymbol{W}_r}(\boldsymbol{X}) + \lambda_c \cdot \mathcal{R}_{\boldsymbol{W}_c}(\boldsymbol{X})$. *Then for* $k = 1, 2, \ldots$, *we have*

$$\begin{aligned} \dot{\sigma}_k(t) = &- L\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}} \left\langle \nabla_{\boldsymbol{W}} \mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}(t)), \boldsymbol{U}_{:,k}(t)\boldsymbol{V}_{:,k}^{\top}(t) \right\rangle \\ &- 2L\left(\sigma_k^2(t)\right)^{\frac{3}{2}-\frac{1}{L}} \gamma_k(t), \end{aligned} \tag{4}$$

where $\boldsymbol{X}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}^{\top}(t)$ *is the SVD for* $\boldsymbol{X}(t)$, $\boldsymbol{W} = \left[\boldsymbol{W}^{[0]}, \boldsymbol{W}^{[1]}, \ldots, \boldsymbol{W}^{[L-1]}\right]$, $\boldsymbol{X} = \sum_s \sigma_s \boldsymbol{U}_{:,s}\boldsymbol{V}_{:,s}^{\top}, \gamma_k(t) = \boldsymbol{U}_{:,k}^{\top}\boldsymbol{L}_r\boldsymbol{U}_{:,k} + \boldsymbol{V}_{:,k}^{\top}\boldsymbol{L}_c\boldsymbol{V}_{:,k} \geq 0$.

Proof. Directly calculate the gradient of $\mathcal{L}_{all}$ at $\boldsymbol{W}$ and utilize Equation 4 will obtain the result. The details of proof can be found in A.3. □

Compared with the results of vanilla DMF whose order of $\sigma_k(t)$ is $2 - \frac{2}{L}$. This Theorem demonstrates that AIR-Net's $\sigma_k(t)$ has a higher dynamics order which is $3 - \frac{2}{L}$. Notice that the adaptive regularizer keeps $\gamma_k(t) \geq 0$. In this way, a bigger convergence speed gap appears between different singular values σ_r than vanilla DMF. Therefore, the AIR-Net enhances the implicit tendency of DMF toward low-rank.

3.2 The dynamics of adaptive regularizer

Now suppose $\boldsymbol{X}$ is given and fixed. We focus on the converge property of $\mathcal{R}_{\mathcal{T}_i(\boldsymbol{W})}$ based on the evolutionary of the dynamics of $\boldsymbol{L}_i$. $\mathcal{R}_{\mathcal{T}_i(\boldsymbol{W})}$ vanishes at the end and avoids AIR-Net suffering from saturation issues.

Theorem 2. *Consider the gradient flow model, where* $\left\|\mathcal{T}_i(\boldsymbol{X})_{k,:}\right\|_F^2 = 1$ *and* $\mathcal{T}_i(\boldsymbol{X})_{k,l} > 0$. *If we initialize* $\boldsymbol{W}_i(0) = \varepsilon \mathbf{1}_{m_i \times m_i}$, *then* $\boldsymbol{W}(t)$ *will keep symmetric during optimization. We can get the following element-wise convergence relationship*

$$\left|\boldsymbol{L}_{i(k,l)}(t) - \boldsymbol{L}^*_{i(k,l)}\right| \leq \begin{cases} (m_i + 2k_i) \cdot \exp(-D \cdot t), & (k,l) \in \mathbb{C}_1 \\ exp\left(-D \cdot t\right), & (k,l) \in \mathbb{C}_2 \\ (m_i - 1) \cdot (m_i + 2k_i) \exp(-D \cdot t), & k = l \end{cases}$$

where $\mathbb{C}_1 = \left\{(k,l) \mid k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} \neq \mathcal{T}_i(\boldsymbol{X})_{:,l}\right\}$, $\mathbb{C}_2 = \left\{(k,l) \mid k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l}\right\}$,

$$\boldsymbol{L}^*_{i(k,l)} = \begin{cases} 0, & (k,l) \in \mathbb{C}_1 \\ \gamma, & (k,l) \in \mathbb{C}_2 \\ -\sum_{l'=1, l' \neq l}^{m_i} \boldsymbol{L}^*_{i(k,l')}, & k = l \end{cases}$$

$\boldsymbol{L}_{i(k,l)}(t)$ *is* (k,l)*-th the element of* $\boldsymbol{L}_i(t)$, $\mathbf{1}_{m_i \times m_i}$ *is a matrix of all-one.* $\gamma = \frac{2}{\left|\left\{\boldsymbol{C}_{\hat{k},\hat{l}}=0\right\}\right|} = \frac{2}{m_i + 2k_i}$, D *is a constant defined in A.4 which equals to zero if and only if* $\boldsymbol{X} = \boldsymbol{I}_{m_i \times n_i}$.

Proof. We prove this theorem in A.4. □

This Theorem gives the limit point $\boldsymbol{L}^*$ and convergence rate of $\boldsymbol{L}_i(t)$. $(k,l) \in \mathbb{C}_1$ $\boldsymbol{L}_i^*(k,l) = 0$. unless $\mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l}$ or $k = l$, that is to say, in the end, $\boldsymbol{L}_i^*$ will only think that the exact same columns in $\mathcal{T}_i(\boldsymbol{X})$ are related. $\boldsymbol{L}_{i(k,l)}(t)$ converges faster when $(k,l) \in \mathbb{C}_2$ than $(k,l) \in \mathbb{C}_1$. In another word, adaptive regularizer captures the similarity first. This convergence rate gap products a multi-scale similarity which will be discussed in Section 4.1. Additionally, it's not difficult to find $\mathcal{R}_{\boldsymbol{W}_i^*} = 0$, the convergence rate is given as follow:

Corollary 1. *In the setting of Theorem 2, we further have* $0 \leq \mathcal{R}_{\boldsymbol{W}_i}(t) \leq 2\left(m_i + 2k_i\right)\left(m_i - 1\right) m_i \cdot \exp(-Dt), i = 1, 2, \ldots, N$.

Proof. We prove this theorem in appendix A.5. □

According to Corollary 1, $\lim_{t\to+\infty} \mathcal{R}_{\boldsymbol{W}_i}(t) \to 0$. Therefore, the regularization will vanish at the end and not induce the saturation issue.

Remark 1. *Notice that we have no restriction on specific $\mathcal{T}_i$, $\mathcal{A}$ or representation of $\boldsymbol{X}$ in the above proof. Therefore, the conclusion in this subsection is a general result for inverse problem.*

In this subsection, we demonstrate AIR-Net's fantastic theoretical properties. It can both enhance the implicit low-rank and avoid saturation issues. We will verify these properties and the effectiveness of AIR-Net in applications experimentally.

4 Experimental Analysis

Now we demonstrate the adaptive properties of AIR-Net by numerical experiments: (a) $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ capture the structural similarity in data from large scale to small scale (Section 4.1); (b) The comprehensive similarity in all scales contribute to successful MC, therefore the adaptive regularizer is necessary (Section 4.2); (c) Because AIR-Net is **adaptive to data**, it avoids over-fitting and achieves good performance. (Section 4.3).

Data type and sampling pattern Three types of matrices are considered: gray-scale image, user-movie rating matrix, and drug-target interaction (DTI) data. Three standard test gray images of size 240×240(Monti et al., 2017) are included in the image type (Baboon, Barbara, and Cameraman). The user-movie rating matrix is Syn-Netflix which is of 150×200, and the DTI data has Ion channels (IC) and G protein-coupled receptor (GPCR) are shaped 210×204 and 223×95 respectively (Boyarski et al., 2019b; Mongia & Majumdar, 2020). The sampling patterns include random missing, patch missing and textural missing, which are listed in Figure 4. The random missing rate varies in different experiments, and the default is 30%.

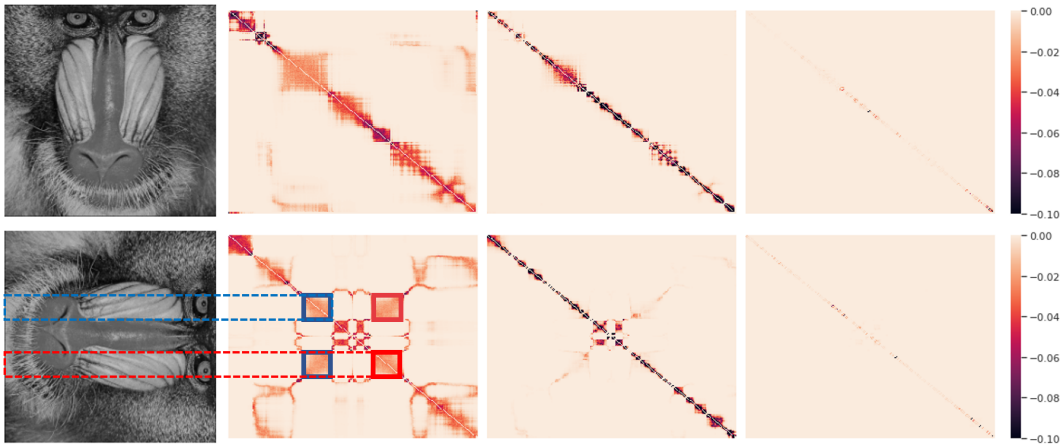
Parameter settings We set $\lambda = \mu = \frac{\boldsymbol{X}^*_{\max} - \boldsymbol{X}^*_{\min}}{m_i \cdot n_i}$ to ensure the fidelity and the regularization are in the same order of magnitude, where $\boldsymbol{X}^*_{\max}$ and $\boldsymbol{X}^*_{\min}$ are maximum and minimum of $\boldsymbol{X}^*$. The δ is a threshold which we set as $\frac{mn}{1000}$ by default. All the parameters in AIR-Net are initialized with Gaussian distribution, which owns zero mean and 10^{-5} as its variance. The Adam is chosen as the optimization algorithm by default (Kingma & Ba, 2015).

4.1 AIR-Net capture relationship adaptive to both spatial and time domain

In this section, we will verify the previously proposed theorems. This section provides a few slices of $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ during training to demonstrate what AIR-Net can learn. The heatmap of $\boldsymbol{L}_r(t)$ and $\boldsymbol{L}_c(t)$ for Baboon at $t = 4000, 7000, 10000$ respectively are shown in Figure 1. The according results for Syn-Netflix are shown in Figure 5 in A.1. The first row shows the heatmap of $\boldsymbol{L}_r(t)$ and the second one shows the heatmap of $\boldsymbol{L}_c(t)$.

As Figure 1 shows, both $\boldsymbol{L}_r(t)$ and $\boldsymbol{L}_c(t)$ first appear many blocks ($t = 4000$). Specially, we sigh two of $\boldsymbol{L}_c(t = 4000)$ out. These blocks indicate that these corresponding blocks columns are highly related. These blocks correspond to columns in which the eyes of Baboon are located, which are indeed highly similar. However, the slight difference between these columns induces the relationship captured by adaptive regularizer focusing on the related columns ($t = 7000$), which is similar to TV(Rudin et al., 1992). The columns of Baboon are not fully the same. The regularization gradually vanishes ($t = 10000$), which matches the results of Theorem 2 (Figure 1). Except the gray-scale images, the results on Syn-Netflix give similar conclusion.

These results illustrate that AIR-Net captures the similarity from large scale to small scale. Meanwhile, a natural question is raised: does there exist a moment that both $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ are captured accurately? If yes, we can train AIR-Net with these fixed $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ to obtain better recovery performance. The experiments below show that the $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ captured by AIR-Net are necessary for MC.



(a) Baboon (b) t=4000 (c) t=7000 (d) t=10000

Figure 1: First row (column) shows the heatmap of $\boldsymbol{L}_r$ ($\boldsymbol{L}_c$) at different t. A darker color indicates a stronger similarity captured by the adaptive regularizer. The (i, j)-th element in the heatmap of $\boldsymbol{L}_r(t)$ has a darker color than the (i, j')-th element indicate that the t-th row is more related to j-th row compared with j'-th row. The area in the middle of the dotted line corresponding to the small block in the figure represents the part of the adaptive positive that is considered similar.

4.2 The necessity of utilizing an adaptive regularizer

In this section, the necessity of adaptive updating $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ is explored. Let AIR-Net have a fixed regularizer, which is an adaptive regularizer learned at a specific step. The Normalized Mean Absolute Error (NMAE) is adopted to measure the distance between the recovered matrix $\hat{\boldsymbol{X}}$ and the actual matrix $\boldsymbol{X}^*$:

$$\text{NMAE} = \frac{1}{(\boldsymbol{X}^*_{\max} - \boldsymbol{X}^*_{\min})\,|\bar{\Omega}|} \sum_{(i,j)\in\bar{\Omega}} \left|\hat{\boldsymbol{X}}_{ij} - \boldsymbol{X}^*_{ij}\right|,$$

where $\bar{\Omega}$ is the complement set of Ω. We utilize the regularization captured by AIR-Net at $t = 4000,\ 7000,\ 9000$ respectively. All of the training hyper-parameters keep the same as AIR-Net. The Baboon under all the three missing patterns are tested.

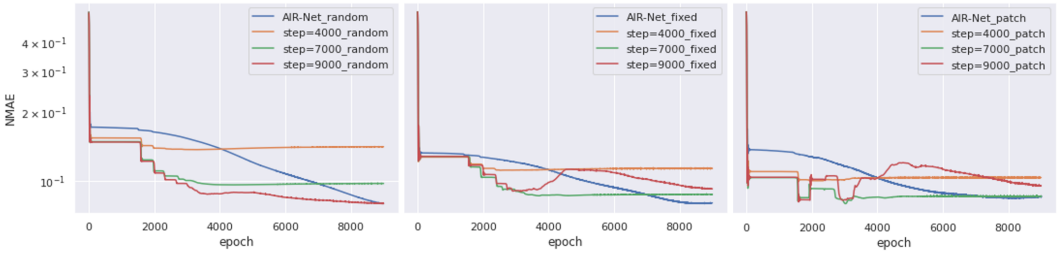
Figure 2 shows how the NMAE changes with the epoch of training. AIR-Net, which updates the regularization during training, achieves the best performance in all missing patterns. The fixed regularization can accelerate the convergence speed of the algorithm. In random missing case, $\mathcal{R}_{\boldsymbol{W}_r}(9000)$ and $\mathcal{R}_{\boldsymbol{W}_c}(9000)$ is the best fixed regularizer among three time steps while other missing cases are $\mathcal{R}_{\boldsymbol{W}_r}(7000)$ and $\mathcal{R}_{\boldsymbol{W}_c}(7000)$. Fixed regularizer based methods will face two problems: (a) How to determine the best step? (b) How to estimate the regularization based on the partially observed matrix before training? These problems are not easy to solve. AIR-Net solves these problems from another perspective by updating the regularization during training. The adaptive property of AIR-Net is essential to the effectiveness of AIR-Net.

4.3 AIR-Net adaptive to both varies data and missing pattern

Now we apply AIR-Net for matrix completion on three data types under different missing patterns.

Peered methods The peered methods include KNN(Goldberger et al., 2004), SVD(Troyanskaya et al., 2001), PNMC(Yang & Xu, 2020), DMF(Arora et al., 2019) and RDMF(Li et al., 2020) in image type. Here RDMF is replaced by DMF+DE(Boyarski et al., 2019a) because it is more suitable in the Syn-Netflix experiment.

Avoid Over-fitting. Figure 3 shows how the NMAE of DMF and AIR-Net changes with the training step. Compared with vanilla DMF, AIR-Net avoids over-fitting and achieves better performance on



(a) Random 30% (b) Texture missing (c) Patch missing

Figure 2: Compare adaptive regularizer with fixed regularizer. The NMAE of recovered Baboon under three types of sampling, including random missing 30% pixels, patch missing, and texture missing, respectively. The blue line indicates the NMAE during training vanilla AIR-Net. Take the $\boldsymbol{L}_r(t)$ and $\boldsymbol{L}_c(t)$ out, t equals 3000, 7000, 9000 respectively. The remind three lines in each figure indicate replacing the $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ with fixed $\boldsymbol{L}_r(t)$ and $\boldsymbol{L}_c(t)$.

all the three data types and missing patterns. The Syn-Netflix and DIT data can be found in Figure 6 at A.1.

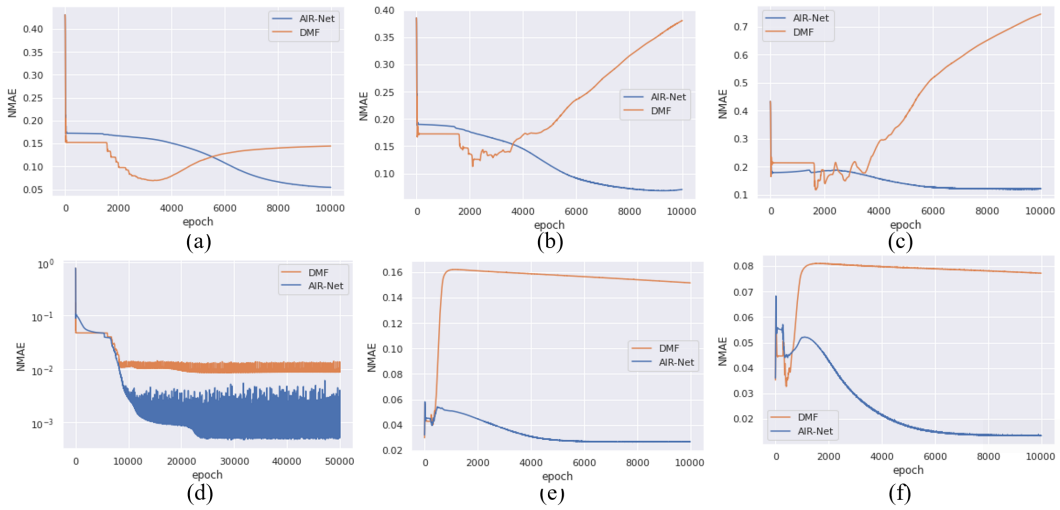


Figure 3: NMAE during training of DMF and AIR-Net. All the figures show the NMAE changes with the training step. The first row shows the results of Cameraman with (a) random missing (The proportion of different percentage figures show that the random missing) (b) textural missing (c) patch missing. The second row shows the remind data type with random missing respectively, including (d) Syn-Netflix, (e)IC and (f) GPCR.

Adaptive to data. Our proposed method achieves the best-recovered performance in most tasks. Table 1 shows the efficacy of AIR-Net on the various data types. More surprising is that our methods perform better than other methods, which are well designed for the particular data type. The recovered results are shown in Figure 4. In this figure, the existing methods perform well on specific missing pattern data. Such as the RDMF achieved good performance on the random missing case but performed not OK on reminding missing patterns. PNMC completed the patch missing well while obtaining worse results on texture missing. Thanks to the proposed model's adaptive properties, our method achieves promising results both visually and by numerical measures.

Table 1: NMAE values of compared algorithms with different missing patterns in different images. The bold font-type indicates the best performance. KNN(Goldberger et al., 2004), SVD(Troyanskaya et al., 2001), PNMC(Yang & Xu, 2020), DMF(Arora et al., 2019), DMF+DE(Boyarski et al., 2019a), RDMF(Li et al., 2020), AIR-Net(proposed). Some elements without value are not suitable for that data type.

Data	Missing	KNN	SVD	PNMC	DMF	RDMF	DMF+DE	Proposed
Barbara	30%	0.083	0.0621	0.0622	0.0613	0.0494	-	**0.0471**
	Patch	0.1563	0.2324	0.2055	0.7664	0.3025	-	**0.1195**
	Texture	0.0712	0.1331	0.1100	0.3885	0.1864	-	**0.0692**
Baboon	30%	0.0831	0.1631	0.0965	0.2134	0.0926	-	**0.0814**
	Patch	0.1195	0.1571	0.1722	0.8133	0.2111	-	**0.1316**
	Texture	0.1237	0.1815	0.1488	0.5835	0.2818	-	**0.1208**
Syn-Netflix	70%	0.0032	0.0376	-	0.0003	-	0.0008	**0.0002**
	75%	0.0046	0.0378	-	0.0004	-	0.0009	**0.0003**
	80%	0.0092	0.0414	-	0.0014	-	0.0012	**0.0007**
IC	20%	0.0169	0.0547	-	0.0773	-	0.0151	**0.0134**
GPCR	20%	0.0409	0.0565	-	0.1513	-	**0.0245**	0.0271

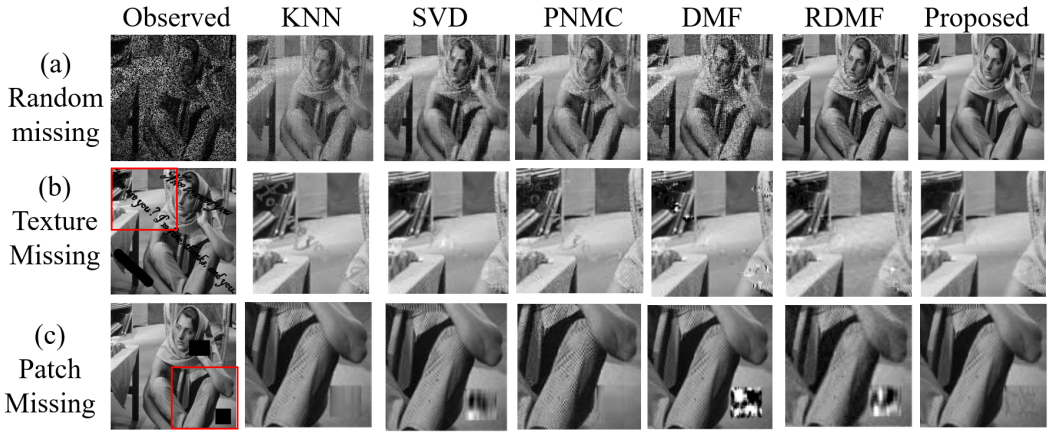


Figure 4: Compared KNN(Goldberger et al., 2004), SVD(Troyanskaya et al., 2001), PNMC(Yang & Xu, 2020), DMF(Arora et al., 2019), RDMF(Li et al., 2020), AIR-Net(proposed) on Babara with three types of data respectively.

5 CONCLUSION

We have proposed AIR-Net which aims to solve the MC problem without knowning the prior in advance. We show that our AIR-Net can adaptively learn the regularization according to different data at different training steps. In addition, we demonstrate that AIR-Net can avoid the saturation issue and over-fitting issue simultaneously. In fact, the AIR-Net is a general framework for solving the inverse problem. In the future work, we will combine other implicit regularization such as F-Principle(Xu et al., 2019) with more flexible $\mathcal{T}_i$ for other inverse problems.

REFERENCES

Sanjeev Arora, Nadav Cohen, W. Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *NeurIPS*, 2019.

A. Boyarski, Sanketh Vedula, and A. Bronstein. Deep matrix factorization with spectral geometric regularization. *arXiv: Learning*, 2019a.

A. Boyarski, Sanketh Vedula, and A. Bronstein. Spectral geometric matrix completion. 2019b.

A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:60–65 vol. 2, 2005.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

Prithvijit Chakrabarty and Subhransu Maji. The spectral bias of the deep image prior. *ArXiv*, abs/2107.01125, 2019.

Kostadin Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16:2080–2095, 2007.

Jicong Fan and Jieyu Cheng. Matrix completion by deep matrix factorization. *Neural networks : the official journal of the International Neural Network Society*, 98:34–41, 2018.

Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42, 2009.

Housen Li, Johannes Schwab, Stephan Antholzer, and M. Haltmeier. Nett: Solving inverse problems with deep neural networks. *ArXiv*, abs/1803.00092, 2018.

Zhemin Li, Zhi-Qin John Xu, Tao Luo, and Hongxia Wang. A regularized deep matrix factorized model of matrix completion for image restoration. *ArXiv*, abs/2007.14581, 2020.

Zhouchen Lin, Minming Chen, and Yuliang Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *ArXiv*, abs/1009.5055, 2010.

Jiaming Liu, Yu Sun, Xiaojian Xu, and U. Kamilov. Image restoration using total variation regularized deep image prior. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7715–7719, 2019.

Christopher A. Metzler, A. Mousavi, Reinhard Heckel, and Richard Baraniuk. Unsupervised learning with stein's unbiased risk estimator. *ArXiv*, abs/1805.10531, 2018.

Aanchal Mongia and A. Majumdar. Drug-target interaction prediction using multi graph regularized nuclear norm minimization. *PLoS ONE*, 15, 2020.

Federico Monti, Michael M. Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*, 2017.

Subhadip Mukherjee, Marcello Carioni, O. Öktem, and C. Schönlieb. End-to-end reconstruction meets data-driven regularization for inverse problems. *ArXiv*, abs/2106.03538, 2021.

Netflix. Netflix prize rules, 2009. `https://www.netflixprize.com/assets/rules.pdf`.

Adityanarayanan Radhakrishnan, G. Stefanakis, Mikhail Belkin, and Caroline Uhler. Simple, fast, and flexible framework for matrix completion with infinite width neural networks. *ArXiv*, abs/2108.00131, 2021.

Nasim Rahaman, A. Baratin, D. Arpit, Felix Dräxler, Min Lin, F. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the spectral bias of neural networks. In *ICML*, 2019.

Yaniv Romano, M. Protter, and Michael Elad. Single image interpolation via adaptive nonlocal sparsity-based modeling. *IEEE Transactions on Image Processing*, 23:3085–3098, 2014.

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17 6:520–5, 2001.

Dmitry Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4:333–361, 2012.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Z. Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *ArXiv*, abs/1901.06523, 2019.

Mingming Yang and Songhua Xu. A novel patch-based nonlinear matrix completion algorithm for image analysis through convolutional neural network. *Neurocomputing*, 389:56–82, 2020.

A Appendix

A.1 Experiments Results

In this section, we place the experiments mentioned before. Figure 5 shows the heatmap of $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ learned by adaptive regularizer. Eventually, adaptive regularizer obtain the $\boldsymbol{L}_r$ and $\boldsymbol{L}_c$ which are highly similar to real $\hat{\boldsymbol{L}}_r$ and $\hat{\boldsymbol{L}}_c$ in first column.

Figure 6 shows the NMAE of Syn-Netflix, IC and GPCR during training, respectively. This experiment result also shows the ability to avoid over-fitting.

A.2 Introduction of DMF

Assumption 1. *Factor matrices are balanced at initialization, i.e.,*

$$\boldsymbol{W}^{[l+1]^\top}(0)\boldsymbol{W}^{[l+1]}(0) = \boldsymbol{W}^{[l]}(0)\boldsymbol{W}^{[l]^\top}(0), \quad l = 0, \ldots, L-2.$$

Under this assumption, Arora et al. studied the gradient flow of the non-regularized risk function $\mathcal{L}_{\mathbb{Y}}$, i.e.,

$$\dot{\boldsymbol{W}}^{[l]}(t) = -\frac{\partial}{\partial \boldsymbol{W}^{[l]}}\mathcal{L}_{\mathbb{Y}}\left(\boldsymbol{X}(t)\right), \quad t \geq 0, \quad l = 0, \ldots, L-1, \tag{5}$$

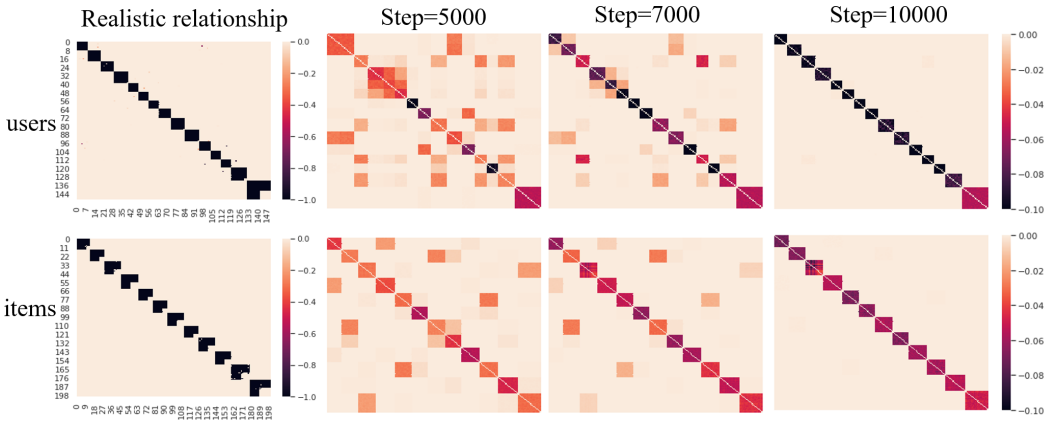


Figure 5: The first column shows the realistic relationship among columns and rows respectively. The remind three columns are the Laplacian matrix learned by AIR at different step.

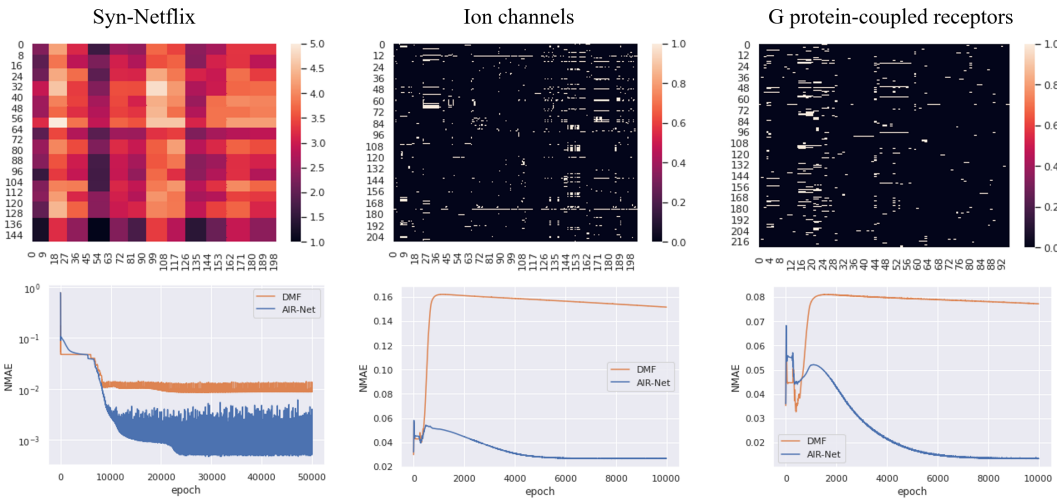


Figure 6: The first row shows the real data of Syn-Netflix, IC and GPCR respectively. The second row shows the corresponding NMAE during training.

where the empirical risk $\mathcal{L}_{\mathbb{Y}}$ can be any analytic function of $\boldsymbol{X}(t)$. According to the analyticity of $\mathcal{L}_{\mathbb{Y}}$, $\boldsymbol{X}(t)$ has the following singular value decomposition where each matrix is an analytic function of t:

$$\boldsymbol{X}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}^{\top}(t),$$

where $\boldsymbol{U}(t) \in \mathbb{R}^{m,\min\{m,n\}}$, $\boldsymbol{S}(t) \in \mathbb{R}^{\min\{m,n\},\min\{m,n\}}$, and $\boldsymbol{V}(t) \in \mathbb{R}^{\min\{m,n\},n}$ are analytic functions of t; and for every t, the matrices $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ have orthonormal columns, while $\boldsymbol{S}(t)$ is diagonal (its diagonal entries may be negative and may appear in any order). The diagonal entries of $\boldsymbol{S}(t)$, which we denote by $\sigma_1(t), \ldots, \sigma_{\min\{m,n\}}(t)$, are signed singular values of $\boldsymbol{X}(t)$. The columns of $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$, denoted by $\boldsymbol{U}_1(t), \ldots, \boldsymbol{U}_{\min\{m,n\}}(t)$ and $\boldsymbol{V}_1(t), \ldots, \boldsymbol{V}_{\min\{m,n\}}(t)$, are the corresponding left and right singular vectors respectively. Based on these notation, Arora derive the following singular values evolutionary dynamics equation.

Proposition 1 ((Arora et al., 2019, Theorem 3)). *Consider the dynamics Equation 5 with initial data satisfying Assumption 1. Then the signed singular values $\sigma_k(t)$ of the product matrix $\boldsymbol{X}(t)$ evolve by:*

$$\begin{aligned} \dot{\sigma}_k(t) = -L\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}} \left\langle \nabla_{\boldsymbol{X}} \mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}(t)), \boldsymbol{U}_{:,k}(t)\boldsymbol{V}_{:,k}^{\top}(t) \right\rangle, \\ k = 1, \ldots, \min\{m,n\}. \end{aligned} \tag{6}$$

If the matrix factorization is non-degenerate, i.e., has depth $L \geq 2$, the singular values need not be signed (we may assume $\sigma_k(t) \geq 0$ for all t).

Arora et al. claimed the terms $\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}}$ enhance the movement of large singular values, and on the other hand, attenuate that of small ones. The enhancement/attenuation becomes more significant as L grows.

A.3 Proof of Theorem 1

We first give the details of the proposed adaptive regularizer with a iterative definition:

$$\begin{cases} \mathcal{R}_{\boldsymbol{W}_i}\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\right) = \operatorname{tr}\left(\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top} \boldsymbol{L}_i \mathcal{T}_i\left(\boldsymbol{X}\right)\right) \\ \boldsymbol{L}_i = \left(\boldsymbol{A}_i^* \cdot \boldsymbol{1}_{m_i \times m_i}\right) \odot \boldsymbol{I}_{m_i} - \boldsymbol{A}_i^* \\ \boldsymbol{A}_i^* = \boldsymbol{A}_i \odot \left(\boldsymbol{1}_{m_i \times m_i} - \boldsymbol{I}_{m_i}\right) \\ \boldsymbol{A}_i = \dfrac{exp(\boldsymbol{W}_i + \boldsymbol{W}_i^{\top})}{\left\|\exp(\boldsymbol{W}_i)\right\|_1} \end{cases},$$

Theorem 1. *Consider the following dynamics with initial parameters satisfying Assumption 1:*

$$\dot{\boldsymbol{W}}^{[l]}(t) = -\frac{\partial}{\partial \boldsymbol{W}^{[l]}} \mathcal{L}_{all}(\boldsymbol{X}(t)), \quad t \geq 0, \quad l = 0, \ldots, L-1,$$

where $\mathcal{L}_{all}\left(\boldsymbol{X}\right) = \mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}) + \lambda_r \cdot \mathcal{R}_{\boldsymbol{W}_r}\left(\boldsymbol{X}\right) + \lambda_c \cdot \mathcal{R}_{\boldsymbol{W}_c}\left(\boldsymbol{X}\right)$. Then we have for any $k = 1, 2, \ldots$

$$\begin{aligned} \dot{\sigma}_k(t) = &- L\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}} \left\langle \nabla_{\boldsymbol{W}} \mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}(t)), \boldsymbol{U}_{:,k}(t)\boldsymbol{V}_{:,k}^{\top}(t) \right\rangle \\ &- 2L\left(\sigma_k^2(t)\right)^{\frac{3}{2}-\frac{1}{L}} \gamma_k(t), \end{aligned}$$

where $\boldsymbol{X}(t) = \boldsymbol{U}(t)\boldsymbol{S}(t)\boldsymbol{V}^{\top}(t)$, $\boldsymbol{X} = \sum_s \sigma_s \boldsymbol{U}_{:,s}\boldsymbol{V}_{:,s}^{\top}$, $\gamma_k(t) = \boldsymbol{U}_{:,k}^{\top}\boldsymbol{L}_r\boldsymbol{U}_{:,k} + \boldsymbol{V}_{:,k}^{\top}\boldsymbol{L}_c\boldsymbol{V}_{:,k} \geq 0$.

Proof. This is proved by direct calculation:

$$\begin{aligned} \nabla_{\boldsymbol{W}}\left(\lambda_r \cdot \mathcal{R}_r + \lambda_c \cdot \mathcal{R}_c\right) &= \frac{\partial \operatorname{tr}\left(\lambda_r \cdot \boldsymbol{X}^{\top}\boldsymbol{L}_r\boldsymbol{X} + \lambda_c \cdot \boldsymbol{X}\boldsymbol{L}_c\boldsymbol{X}^{\top}\right)}{\partial \boldsymbol{X}} \\ &= 2\lambda_r \cdot \boldsymbol{L}_r\boldsymbol{X} + 2\lambda_c \cdot \boldsymbol{X}\boldsymbol{L}_c \\ &= 2\lambda_r \cdot \boldsymbol{L}_r \sum_s \sigma_s \boldsymbol{U}_{:,s}\boldsymbol{V}_{:,s}^{\top} + 2\lambda_c \cdot \sum_s \sigma_s \boldsymbol{U}_{:,s}\boldsymbol{V}_{:,s}^{\top}\boldsymbol{L}_c. \end{aligned}$$

Note that

$$\langle \boldsymbol{V}_{:,s}, \boldsymbol{V}_{s'} \rangle = \langle \boldsymbol{U}_{:,s}, \boldsymbol{U}_{s'} \rangle = \delta_{ss'} = \begin{cases} 1, & s = s', \\ 0. & s \neq s'. \end{cases}$$

Therefore

$$\begin{aligned}\boldsymbol{U}_{:,k}^{\top}\left(\nabla_{\boldsymbol{W}}\left(\lambda_r\cdot\mathcal{R}_r+\lambda_c\cdot\mathcal{R}_c\right)\right)\boldsymbol{V}_{:,k}&=2\sigma_k(\lambda_r\cdot\boldsymbol{U}_{:,k}^{\top}\boldsymbol{L}_r\boldsymbol{U}_{:,k}+\lambda_c\cdot\boldsymbol{V}_{:,k}^{\top}\boldsymbol{L}_c\boldsymbol{V}_{:,k})\\&=2\sigma_k\gamma_k(t),\end{aligned}$$

where the term $\gamma_k(t)=2\sigma_k(\lambda_r\cdot\boldsymbol{U}_{:,k}^{\top}\boldsymbol{L}_r\boldsymbol{U}_{:,k}+\lambda_c\cdot\boldsymbol{V}_{:,k}^{\top}\boldsymbol{L}_c\boldsymbol{V}_{:,k})\geq 0$. Furthermore, according to Equation 6, we have $\boldsymbol{U}_{:,k}^{\top}\nabla_{\boldsymbol{W}}\mathcal{L}_{\mathbb{Y}}\boldsymbol{V}_{:,k}=-L\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}}\left\langle\nabla_{\boldsymbol{W}}\mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}(t)),\boldsymbol{U}_{:,k}(t)\boldsymbol{V}_{:,k}^{\top}(t)\right\rangle$.

Finally, according to $\dot{\boldsymbol{W}}^{[l]}(t)=-\frac{\partial}{\partial\boldsymbol{W}^{[l]}}\mathcal{L}_{all}(\boldsymbol{X}(t))$, we have

$$\dot{\sigma}_k(t)=-L\left(\sigma_k^2(t)\right)^{1-\frac{1}{L}}\left\langle\nabla_{\boldsymbol{W}}\mathcal{L}_{\mathbb{Y}}(\boldsymbol{X}(t)),\boldsymbol{U}_{:,k}(t)\boldsymbol{V}_{:,k}^{\top}(t)\right\rangle-2L\left(\sigma_k^2(t)\right)^{\frac{3}{2}-\frac{1}{L}}\gamma_k(t).$$

□

A.4 Proof of Theorem 2

Proposition 2. *$\nabla_{\boldsymbol{W}_i}\left(\mathcal{R}_{\boldsymbol{W}_i}\left(\boldsymbol{X}\right)\right)=2\boldsymbol{C}\odot\boldsymbol{A}_i-2\mathrm{tr}\left(\boldsymbol{C}\boldsymbol{A}_i'\right)\boldsymbol{A}_i'$, where $\boldsymbol{A}_i'=\frac{\exp(\boldsymbol{W}_i)}{\|\exp(\boldsymbol{W}_i)\|_1}$, $\boldsymbol{A}_i=\boldsymbol{A}_i'+\boldsymbol{A}_i'^{\top}$ and $\boldsymbol{C}=\mathbf{1}_{m_i\times m_i}\cdot\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top}\odot\boldsymbol{I}_{m_i}\right)-\mathcal{T}_i\left(\boldsymbol{X}\right)\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top}$.*

Proof. We denote $\boldsymbol{X}=\mathcal{T}_i\left(\boldsymbol{X}\right)\in\mathbb{R}^{m_i\times n_i}$, then we consider $d\left[\mathrm{tr}\left(\boldsymbol{X}^{\top}\boldsymbol{L}_i\boldsymbol{X}\right)\right]$

$$\begin{aligned}&d\left[\mathrm{tr}\left(\boldsymbol{X}^{\top}\boldsymbol{L}_i\boldsymbol{X}\right)\right]\\=&\mathrm{tr}\left[d\left(\boldsymbol{L}_i\boldsymbol{X}\boldsymbol{X}^{\top}\right)\right]\\=&\mathrm{tr}\left[\left(d\boldsymbol{A}_i\odot\left(\mathbf{1}_{m_i\times m_i}-\mathbf{I}_{m_i}\right)\cdot\mathbf{1}_{m_i\times m_i}\right)\odot\boldsymbol{I}_n\cdot\boldsymbol{X}\boldsymbol{X}^{\top}\right.\\&\left.-d\boldsymbol{A}_i\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\boldsymbol{X}\boldsymbol{X}^{\top}\right]\\=&\mathrm{tr}\left[\left(\boldsymbol{X}\boldsymbol{X}^{\top}\right)^{\top}\left(\boldsymbol{I}_{m_i}\odot\left(d\boldsymbol{A}_i\odot\left(\mathbf{1}_{m_i\times m_i}-\mathbf{I}_{m_i}\right)\cdot\mathbf{1}_{m_i\times m_i}\right)\right)\right.\\&\left.-\left(\boldsymbol{X}\boldsymbol{X}^{\top}\right)^{\top}\left(\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\odot d\boldsymbol{A}_i\right)\right]\\=&\mathrm{tr}\left[\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\boldsymbol{I}_{m_i}\right)^{\top}d\boldsymbol{A}_i\odot\left(\mathbf{1}_{m_i\times m_i}-\mathbf{I}_{m_i}\right)\cdot\mathbf{1}_{m_i\times m_i}\right.\\&\left.-\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\right)^{\top}d\boldsymbol{A}_i\right]\\=&\mathrm{tr}\left[\left(\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\boldsymbol{I}_{m_i}\right)\mathbf{1}_{m_i\times m_i}\right)^{\top}\left(d\boldsymbol{A}_i\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\right)\right.\\&\left.-\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\right)^{\top}d\boldsymbol{A}_i\right]\\=&\mathrm{tr}\left[\left(\left(\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\boldsymbol{I}_{m_i}\right)\cdot\mathbf{1}_{m_i\times m_i}\right)\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)-\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\left(\mathbf{1}_{m_i\times m_i}-\boldsymbol{I}_{m_i}\right)\right)^{\top}\right)d\boldsymbol{A}_i\right]\\=&\mathrm{tr}\left[\left(\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\boldsymbol{I}_{m_i}\right)\mathbf{1}_{m_i\times m_i}-\boldsymbol{X}\boldsymbol{X}^{\top}\right)d\boldsymbol{A}_i\right]\end{aligned}$$

We denote $\boldsymbol{C}=\left(\boldsymbol{X}\boldsymbol{X}^{\top}\odot\boldsymbol{I}_{m_i}\right)\mathbf{1}_{m_i\times m_i}-\boldsymbol{X}\boldsymbol{X}^{\top}, S_{\boldsymbol{W}_i}=\mathbf{1}_{m_i}^{\top}\cdot exp\left(\boldsymbol{W}_i\right)\cdot\mathbf{1}_{m_i}$, then

$$
\begin{aligned}
& d\left[\operatorname{tr}\left(\boldsymbol{X}^{\top}\boldsymbol{L}_i\boldsymbol{X}\right)\right] \\
=&\operatorname{tr}\left(\boldsymbol{C}d\boldsymbol{A}_i\right) \\
=&\frac{1}{S_{\boldsymbol{W}_i}^2}\operatorname{tr}\left[\boldsymbol{C}\left(S_{\boldsymbol{W}_i}\cdot\exp(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top})\odot d(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top})\right)\right. \\
&\left.-\boldsymbol{C}\left(\mathbf{1}_{m_i}^{\top}\left(\exp(\boldsymbol{W}_i)\odot d\boldsymbol{W}_i\right)\mathbf{1}_{m_i}\right)\exp(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top})\right] \\
=&\operatorname{tr}\left[\boldsymbol{C}\cdot\left(\boldsymbol{A}_i\odot d\left(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top}\right)\right)\right] \\
&-\frac{1}{S_{\boldsymbol{W}_i}^2}\operatorname{tr}\left[\mathbf{1}_{m_i\times m_i}\left(\exp(\boldsymbol{W}_i)\odot d\boldsymbol{W}_i\right)\right]\cdot\operatorname{tr}\left[\boldsymbol{C}\cdot exp\left(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top}\right)\right] \\
=&\operatorname{tr}\left[\boldsymbol{C}\cdot\left(\boldsymbol{A}_i\odot d\left(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top}\right)\right)\right] \\
&-\frac{1}{S_{\boldsymbol{W}_i}^2}\operatorname{tr}\left[\left(\mathbf{1}_{m_i\times m_i}\odot\exp(\boldsymbol{W}_i)\right)d\boldsymbol{W}_i\right]\cdot\operatorname{tr}\left[\boldsymbol{C}\cdot exp\left(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top}\right)\right] \\
=&\operatorname{tr}\left[\boldsymbol{C}\cdot\left(\boldsymbol{A}_i\odot d\left(\boldsymbol{W}_i+\boldsymbol{W}_i^{\top}\right)\right)\right]-\operatorname{tr}\left[\operatorname{tr}\left(\boldsymbol{C}\cdot\boldsymbol{A}_i\right)\cdot\boldsymbol{A}_i'd\boldsymbol{W}_i\right] \\
=&\operatorname{tr}\left[\left(\left(\boldsymbol{C}^{\top}\odot\boldsymbol{A}_i\right)^{\top}+\boldsymbol{C}^{\top}\odot\boldsymbol{A}_i-\operatorname{tr}\left(\boldsymbol{C}\cdot\boldsymbol{A}_i\right)\boldsymbol{A}_i'\right)d\boldsymbol{W}_i\right]
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\nabla_{\boldsymbol{W}_i}\operatorname{tr}\left(\boldsymbol{X}^{\top}\boldsymbol{L}_i\boldsymbol{X}\right)&=\left(\boldsymbol{C}^{\top}\odot\boldsymbol{A}_i\right)^{\top}+\boldsymbol{C}^{\top}\odot\boldsymbol{A}_i-\operatorname{tr}\left(\boldsymbol{C}\cdot\boldsymbol{A}_i\right)\boldsymbol{A}_i' \\
&=2\boldsymbol{C}\odot\boldsymbol{A}_i-2\operatorname{tr}\left(\boldsymbol{C}\boldsymbol{A}_i'\right)\boldsymbol{A}_i'
\end{aligned}
$$

Notice that $\boldsymbol{X}=\mathcal{T}_i\left(\boldsymbol{X}\right)\in\mathbb{R}^{m_i\times n_i}$, the proposition is proved. □

Theorem 2. *Consider the gradient flow model, assume* $\left\|\mathcal{T}_i\left(\boldsymbol{X}\right)_{k,:}\right\|_F^2=1$ *and* $\mathcal{T}_i\left(\boldsymbol{X}\right)_{k,l}>0$*, if we initialize* $\boldsymbol{W}_i\left(0\right)=\varepsilon\mathbf{1}_{m_i\times m_i}$*, then* $\boldsymbol{W}\left(t\right)$ *will keep symmetric during optimization. We can get the element-wise convergence relationship*

$$
\left|\boldsymbol{L}_{i(k,l)}(t)-\boldsymbol{L}_{i(k,l)}^{*}\right|\leq\begin{cases}(m_i+2k_i)\cdot\exp(-D\cdot t), & (k,l)\in\mathbb{C}_1\\ exp\left(-D\cdot t\right), & (k,l)\in\mathbb{C}_2\\ (m_i-1)\cdot(m_i+2k_i)\exp(-D\cdot t), & k=l\end{cases}
$$

where $\mathbb{C}_1=\left\{(k,l)\mid k\neq l,\mathcal{T}_i\left(\boldsymbol{X}\right)_{:,k}\neq\mathcal{T}_i\left(\boldsymbol{X}\right)_{:,l}\right\}$, $\mathbb{C}_2=\left\{(k,l)\mid k\neq l,\mathcal{T}_i\left(\boldsymbol{X}\right)_{:,k}=\mathcal{T}_i\left(\boldsymbol{X}\right)_{:,l}\right\}$,

$$
\boldsymbol{L}_{i(k,l)}^{*}=\begin{cases}0, & (k,l)\in\mathbb{C}_1\\ \gamma, & (k,l)\in\mathbb{C}_2\\ -\sum_{l'=1,l'\neq l}^{m_i}\boldsymbol{L}_{i(k,l')}^{*}, & k=l\end{cases}
$$

$\boldsymbol{L}_{i(k,l)}(t)$ *is the element of* $\boldsymbol{L}_i(t)$ *at the* k*-th row and* l*-th column,* $\mathbf{1}_{m_i\times m_i}$ *is all one elements matrix.* $\gamma=\frac{2}{\left|\left\{\boldsymbol{C}_{k,l}=0\right\}\right|}=\frac{2}{m_i+2k_i}$*,* D *is a constant defined in A.4 which equals to zero if and only if* $\boldsymbol{X}=\boldsymbol{1}_{m_i\times n_i}$.

Proof. We rewritten the gradient in proposition 2 with element wise formulation:

$$
\dot{\boldsymbol{W}}_{\mathcal{T}_{i(k,l)}}\left(t\right)=(2Ca(t)-4\boldsymbol{C}_{k,l})\cdot\boldsymbol{A}_{i(k,l)}'(t),
$$

where $Ca(t)=\operatorname{tr}\left(\boldsymbol{C}\boldsymbol{A}_i'\right)$ and the sub index denote the element in matrix.

With the assumption that $\left\|\mathcal{T}_i\left(\boldsymbol{X}\right)_{k,:}\right\|_F^2=1$ and $\mathcal{T}_i\left(\boldsymbol{X}\right)_{k,l}>0$, we have $0\leq\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top}\right)_{k,l}\leq 1$. Therefore $\boldsymbol{C}_{k,l}=\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top}\right)_{k,k}-\left(\mathcal{T}_i\left(\boldsymbol{X}\right)\mathcal{T}_i\left(\boldsymbol{X}\right)^{\top}\right)_{k,l}\geq 0$ and specially $\boldsymbol{C}_{k,k}=0$, as $\boldsymbol{A}_{i(k,l)}'=\frac{\exp(\boldsymbol{W}_{ik,l})}{\|\exp(\boldsymbol{W}_i)\|_1}>0$, we have

$$
Ca(t)=\operatorname{tr}\left(\boldsymbol{C}\boldsymbol{A}_i'\right)=\sum_{k=1,l=1}^{m_i}C_{k,l}\boldsymbol{A}_{i(k,l)}'(t)>0
$$

Denote $\boldsymbol{C}_{\hat{k},\hat{l}} \in \min_{k,l} \boldsymbol{C}_{k,l}$, we have $\boldsymbol{C}_{\hat{k},\hat{l}} \leq \boldsymbol{C}_{k,l}$ and then consider

$$\begin{aligned}
&\dot{\boldsymbol{W}}_{\mathcal{T}_{i(\hat{k},\hat{l})}}(t) - \dot{\boldsymbol{W}}_{\mathcal{T}_{i(k,l)}}(t) \\
&= 2Ca(t)\left(\boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) - \boldsymbol{A}'_{i(k,l)}(t)\right) - 4\left(\boldsymbol{C}_{\hat{k},\hat{l}}\boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) - \boldsymbol{C}_{k,l}\boldsymbol{A}'_{i(k,l)}(t)\right)
\end{aligned}$$

As we initialize $\boldsymbol{W}_i(0) = \varepsilon \mathbf{1}_{m_i \times m_i}$, therefore $\boldsymbol{A}'_{i(k,l)}(0) = \frac{1}{m_i^2}, \forall k,l$. Therefore $\dot{\boldsymbol{W}}_{\mathcal{T}_{i(\hat{k},\hat{l})}}(0) - \dot{\boldsymbol{W}}_{\mathcal{T}_{i(k,l)}}(0) = -4\left(\boldsymbol{C}_{\hat{k},\hat{l}} - \boldsymbol{C}_{k,l}\right)\boldsymbol{A}'_{i(k,l)}(0) = -\frac{4}{m_i^2}\left(\boldsymbol{C}_{\hat{k},\hat{l}} - \boldsymbol{C}_{k,l}\right) \geq 0$. Then we have $\boldsymbol{W}_{\mathcal{T}_{i(\hat{k},\hat{l})}}(t) \geq \boldsymbol{W}_{\mathcal{T}_{i(k,l)}}(t)$ and $\boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) \geq \boldsymbol{A}'_{i(k,l)}(t)$, the equal is token if and only if $t = 0$ or $\boldsymbol{C}_{\hat{k},\hat{l}} = \boldsymbol{C}_{k,l}$. Furthermore, $\dot{\boldsymbol{W}}_{\mathcal{T}_{i(\hat{k},\hat{l})}}(t) - \dot{\boldsymbol{W}}_{\mathcal{T}_{i(k,l)}}(t) \geq -4\left(\boldsymbol{C}_{\hat{k},\hat{l}} - \boldsymbol{C}_{k,l}\right)\boldsymbol{A}'_{i(k,l)}(0)$, then $\boldsymbol{W}_{\mathcal{T}_{i(\hat{k},\hat{l})}}(t) - \boldsymbol{W}_{\mathcal{T}_{i(k,l)}}(t) \geq \boldsymbol{D}_{\hat{k},\hat{l},k,l} \cdot t$, where $\boldsymbol{D}_{\hat{k},\hat{l},k,l} = -4\left(\boldsymbol{C}_{\hat{k},\hat{l}} - \boldsymbol{C}_{k,l}\right)\boldsymbol{A}'_{i(k,l)}(0) \geq 0$. Next, we consider

$$\begin{aligned}
\boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) &= \frac{\exp(\boldsymbol{W}_{i(\hat{k},\hat{l})})}{\left\|\exp(\boldsymbol{W}_{\mathcal{T}_{i(k,l)}})\right\|_1} \\
&= \frac{\exp(\boldsymbol{W}_{i(\hat{k},\hat{l})})}{\sum_{k,l}\exp(\boldsymbol{W}_{\mathcal{T}_{i(k,l)}})} = \frac{1}{\sum_{k,l}\exp(\boldsymbol{W}_{\mathcal{T}_{i(k,l)}} - \boldsymbol{W}_{\mathcal{T}_{i(\hat{k},\hat{l})}})} \\
&\geq \frac{1}{\sum_{k,l} exp\left(-\boldsymbol{D}_{\hat{k},\hat{l},k,l} \cdot t\right)}
\end{aligned}$$

As $\boldsymbol{C}_{k,k} = 0$ and $\boldsymbol{C}_{k,l} \geq 0$, therefore $\boldsymbol{C}_{k,k} \in \min_{k,l} \boldsymbol{C}_{k,l}$=0. It is not difficult to show that $\boldsymbol{C}_{\hat{k},\hat{l}} = 0$ if and only if $\mathcal{T}_i(\boldsymbol{X})_{:,\hat{k}} = \mathcal{T}_i(\boldsymbol{X})_{:,\hat{l}}$. If $\left|\left\{\boldsymbol{C}_{\hat{k},\hat{l}} = 0\right\}\right| = m_i + 2k_i$, then when $\boldsymbol{C}_{\hat{k},\hat{l}} = 0$, $\boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) \geq \frac{1}{m_i + 2k_i + \boldsymbol{E}_{\hat{k},\hat{l}}(t)}$, where $\boldsymbol{E}_{\hat{k},\hat{l}}(+\infty) = 0$. Notice that $\sum_{k,l} \boldsymbol{A}'_{i(k,l)}(t) = 1$, we have

$$\frac{1}{m_i + 2k_i + \boldsymbol{E}_{\hat{k},\hat{l}}(t)} \leq \boldsymbol{A}'_{i(\hat{k},\hat{l})}(t) \leq \frac{1}{m_i + 2k_i}$$

Therefore, $\boldsymbol{A}'_{i(\hat{k},\hat{l})}(+\infty) = \begin{cases} 0 & ,\mathcal{T}_i(\boldsymbol{X})_{:,k} \neq \mathcal{T}_i(\boldsymbol{X})_{:,l} \\ \frac{1}{m_i+2k_i} & ,\mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l} \end{cases}$. Furthermore, $\boldsymbol{A}_{i(\hat{k},\hat{l})} = 2\boldsymbol{A}'_{i(\hat{k},\hat{l})} = \frac{2}{m_i+2k_i} = \gamma$, $\boldsymbol{A}_{i(k,l)}(+\infty) = \begin{cases} 0 & ,\mathcal{T}_i(\boldsymbol{X})_{:,k} \neq \mathcal{T}_i(\boldsymbol{X})_{:,l} \\ \gamma & ,\mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l} \end{cases}$. According to the definition of $\boldsymbol{L}_i$,

$$\boldsymbol{L}^*_{i(k,l)} = \boldsymbol{L}_{i(k,l)}(t)(+\infty) = \begin{cases} 0 & k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} \neq \mathcal{T}_i(\boldsymbol{X})_{:,l} \\ \gamma & k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l} \\ -\sum_{l'=1,l'\neq k}^{m_i} \boldsymbol{L}^*_{i(k,l')} & k = l \end{cases}$$

Until now, we have prove that the adaptive regularization part of AIR-Net will convergence at the end. That is the upper bound of $\left|\boldsymbol{L}^*_{i(k,l)} - \boldsymbol{L}_{i(k,l)}(t)\right|$. Next we will focus on the convergence rate of AIR-Net. We discuss the rate under the three cases in the aforementioned formulation separately.

We simplify the notation furthermore before continue. $\mathbb{C}_1 = \left\{(k,l) \mid k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} \neq \mathcal{T}_i(\boldsymbol{X})_{:,l}\right\}$, $\mathbb{C}_2 = \left\{(k,l) \mid k \neq l, \mathcal{T}_i(\boldsymbol{X})_{:,k} = \mathcal{T}_i(\boldsymbol{X})_{:,l}\right\}$. Then the formulation is simplified as

$$\boldsymbol{L}^*_{i(k,l)} = \begin{cases} 0 & (k,l) \in \mathbb{C}_1 \\ \gamma & (k,l) \in \mathbb{C}_2 \\ -\sum_{l'=1,l'\neq k}^{m_i} \boldsymbol{L}^*_{i(k,l')} & k = l \end{cases}$$

If $(k,l) \in \mathbb{C}_2$ or $k = l$, we denote $D = \min D_{k,l}$, according to the definition of $\boldsymbol{E}_{k,l}(t)$, we have $\boldsymbol{E}_{k,l}(t) \leq \exp(-D \cdot t)$

$$\begin{aligned}\left|\boldsymbol{A}^*_{i(k,l)} - \boldsymbol{A}_{i(k,l)}(t)\right| &= 2\left[\frac{1}{m_i + 2k_i} - \frac{1}{m_i + 2k_i + \boldsymbol{E}_{k,l}(t)}\right] \\ &\leq 2\frac{\boldsymbol{E}_{k,l}(t)}{(m_i + 2k_i)^2} \\ &\leq 2 \cdot \frac{\frac{m_i(m_i-1)}{2} \cdot \exp(-D \cdot t)}{(m_i + 2k_i)^2} \\ &\leq exp\left(-D \cdot t\right)\end{aligned}$$

Specially, when $(k,l) \in \mathbb{C}_2$ we have

$$\left|\boldsymbol{L}^*_{i(k,l)} - \boldsymbol{L}_{i(k,l)}(t)\right| = \left|\boldsymbol{A}^*_{i(k,l)} - \boldsymbol{A}_{i(k,l)}(t)\right| \leq exp\left(-D \cdot t\right)$$

If $(k,l) \in \mathbb{C}_1$,

$$\begin{aligned}\left|\boldsymbol{L}^*_{i(k,l)} - \boldsymbol{L}_{i(k,l)}(t)\right| &= \left|\boldsymbol{A}^*_{i(k,l)} - \boldsymbol{A}_{i(k,l)}(t)\right| = \left|\boldsymbol{A}_{i(k,l)}(t)\right| \\ &\leq \left|\sum_{(k',l') \in \mathbb{C}_1 \cup sC_3} \boldsymbol{A}_{i(k',l')}(t)\right| \\ &= \left|2 - \sum_{(k',l') \in \mathbb{C}_2 \cup \mathbb{C}_3} \boldsymbol{A}_{i(k',l')}(t)\right| \\ &= \left|\gamma \cdot (m_i + 2k_i) - \sum_{(k',l') \in \mathbb{C}_2 \cup \mathbb{C}_3} \boldsymbol{A}_{i(k',l')}(t)\right| \\ &= \left|\sum_{(k',l') \in \mathbb{C}_2 \cup \mathbb{C}_3} \left(\gamma - \boldsymbol{A}_{i(k',l')}(t)\right)\right| \\ &\leq \sum_{(k',l') \in \mathbb{C}_2 \cup \mathbb{C}_3} \left|\gamma - \boldsymbol{A}_{i(k',l')}(t)\right| \\ &= \sum_{(k',l') \in \mathbb{C}_2 \cup \mathbb{C}_3} \left|\boldsymbol{A}^*_{i(k',l')} - \boldsymbol{A}_{i(k',l')}(t)\right| \leq (m_i + 2k_i) \cdot \exp(-D \cdot t)\end{aligned}$$

If $k = l$,

$$\left|\boldsymbol{L}^*_{i(k,l)} - \boldsymbol{L}_{i(k,l)}(t)\right| = \left|\sum_{l'=1, l' \neq k}^{m_i} \left(\boldsymbol{L}^*_{i(k,l')} - \boldsymbol{L}_{i(k,l')}(t)\right)\right| \leq (m_i - 1) \cdot (m_i + 2k_i) \cdot \exp(-D \cdot t)$$

□

A.5 Proof of Corollary 1

Corollary 1. *In the setting of Theorem 2, we further have* $0 \leq \mathcal{R}_{\boldsymbol{W}_i}(t) \leq 2\left(m_i + 2k_i\right)\left(m_i - 1\right) m_i \cdot \exp(-Dt)$.

Proof.

$$\begin{aligned}\mathcal{R}_{\boldsymbol{W}_i}(t) &= \sum_{k,l} \boldsymbol{L}_{i(k,l)}(t)(t) \left\|\mathcal{T}_i(\boldsymbol{X})_{:,k} - \mathcal{T}_i(\boldsymbol{X})_{:,l}\right\|_F^2 \\ &\leq 2 \cdot \sum_{k \neq l} \boldsymbol{L}_{i(k,l)}(t)(t) \leq m_i\left(m_i - 1\right)\left(m_i + 2k_i\right) \cdot \exp(-Dt)\end{aligned}$$

□