Interpretable Feature Extraction by Supervised Dictionary Learning for Identification of Cancer-Associated Gene Clusters

Anonymous Authors[1]

Abstract

Supervised dictionary learning (SDL) is a popular machine learning method that tackles the tasks of feature extraction and classification tasks simultaneously, which are not necessarily inherently aligned. Training an SDL model involves solving a non-convex and possibly constrained optimization with at least three blocks of parameters. In this paper, we provide a novel framework that 'lifts' SDL as a low-rank matrix estimation problem in a combined factor space and propose an efficient algorithm that provably converges exponentially fast to a global minimizer of the objective with arbitrary initialization. Our framework applies to a wide range of SDL-type problems for multi-class classification with the inclusion of possible auxiliary covariates. We demonstrate that our algorithm successfully identifies discriminative gene groups that include well-known cancer-associated genes.

1. Introduction

In classical classification models, such as logistic regression, the conditional class-generating probability distribution is modeled as a simple function of the observed features with unknown parameters to be trained. However, the raw observed features may be high-dimensional, and most of them might be uninformative and hard to interpret (e.g., pixel values of an image). Therefore, it would be desirable to extract more informative and interpretable low-dimensional features prior to the classification task.

A classical unsupervised feature extraction framework is called *dictionary learning* (DL), a machine-learning technique that learns latent structures of complex datasets and is regularly applied in the analysis of text and images (Elad & Aharon, 2006; Mairal et al., 2007; Peyré, 2009). Extensive research has been conducted to adapt dictionary learning models to perform classification tasks by supervising the dictionary learning process using additional class labels. Note that dictionary learning and classification are not necessarily aligned objectives, so some degree of trade-off is necessary when seeking to achieve both goals simultaneously. *Supervised dictionary learning* (SDL) provides systematic approaches for such multi-objective tasks (Mairal et al., 2008; Austin et al., 2018; Leuschner et al., 2019; Ritchie et al., 2020). SDL has been widely applied in diverse research domains, demonstrating its versatility and effectiveness. For instance, it has been successfully applied in speech and emotion recognition (Gangeh et al., 2014), music genre classification (Zhao et al., 2015a), concurrent brain network inference (Zhao et al., 2015a), structure-aware clustering (Yankelevsky & Elad, 2017), and object recognition (Li et al., 2019). See the survey (Gangeh et al., 2015) on SDL.

Various SDL-type models have been proposed in the past two decades. We divide them into two categories depending on whether the extracted low-dimensional feature or the feature extraction mechanism itself is supervised. We refer to them as "feature-based" and "filter-based" SDL, respectively. Feature-based SDL models include the classical ones by Mairal et al. (see, e.g., (Mairal et al., 2008; 2011)) as well as the more recent model of Convolutional Matrix Factorization by Kim et al. (2016) for a contextual text recommendation system. Filter-based SDL models have been studied more recently in the supervised matrix factorization literature, most notably from supervised nonnegative matrix factorization (Austin et al., 2018; Leuschner et al., 2019) and supervised PCA (Ritchie et al., 2020). In spite of the vast literature on SDL, due to the high non-convexity of the associated optimization problem, algorithms for SDL mostly lack rigorous convergence analysis and there has not been any algorithm that provably converges to a global minimizer of the objective at an exponential rate.

In this paper, we formulate a general class of SDL-type models encompassing both feature-based and filter-based approaches for multi-class classification. These models are designed to effectively handle high-dimensional features and incorporate valuable information from low-dimensional auxiliary covariates. To find the solutions of SDL-type models, we provide a novel framework that 'lifts' SDL as

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.

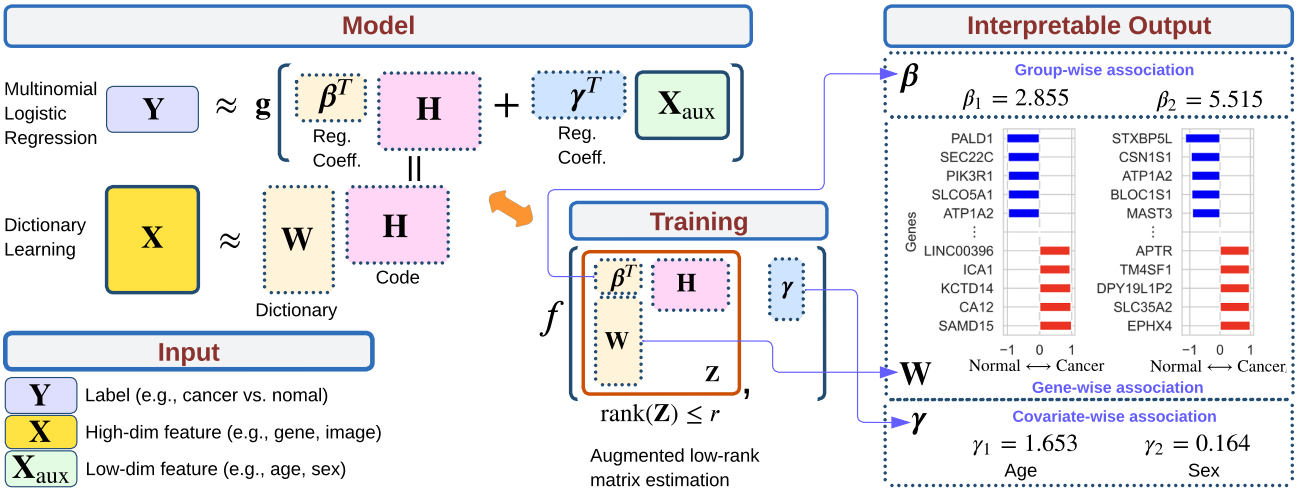


Figure 1. Overall scheme of the proposed method for SDL-$\mathbf{H}$. ***Model***: The model is designed for multi-class classification by combining low-dimensional informative features (such as age and sex) and high-dimensional features (such as genes) that may not all be informative or easily interpretable. 'Discriminative feature extraction' is performed so that we can identify meaningful, low-dimensional structures from the original high-dimensional features for classification tasks, such as cancer-associated gene groups containing genes that are highly correlated with each other. These extracted features, along with the low-dimensional features themselves, are utilized in the classification task. Our approach performs classification and feature extraction tasks simultaneously, ensuring effective performance in learning the classification model. ***Training***: We propose a novel framework that transforms the problem into a low-rank matrix estimation problem and an exponentially fast algorithm for finding the global optimum, providing reliable and efficient results. ***Output***: The resulting model takes the form of (multinomial) logistic regression, which offers interpretability in its outputs. Specifically, the regression coefficients reveal insights into the importance of the low-dimensional features, discriminative gene groups, and individual genes within its groups on a covariate-wise, group-wise, and gene-wise level, respectively.

a low-rank matrix estimation problem in combined factor space. Additionally, we introduce an efficient algorithm that converges exponentially fast to a global minimizer of the objective, regardless of the initial conditions. Our theoretical findings are validated through extensive numerical experiments. Applying our method to microarray datasets for cancer classification, we show that not only it is competitive against benchmark methods, but also it is able to identify groups of genes including well-known cancer-associated genes.

2. Methods

2.1. Model setup

Suppose we are given with n labeled signals $(y_i, \mathbf{x}_i, \mathbf{x}_i')$ for $i = 1, \ldots, n$, where $y_i \in \{0, 1, \ldots, \kappa\}$ is the label, $\mathbf{x}_i \in \mathbb{R}^p$ is a high-dimensional feature of i, and $\mathbf{x}_i' \in \mathbb{R}^q$ is a low-dimensional auxiliary feature of i ($p \gg q$). For a vivid context, think of $\mathbf{x}_i$ as an X-ray image of a patient i and $\mathbf{x}_i'$ denoting some biological measurements, such as gender, smoking status, and body mass index. When making predictions of y_i, we use a suitable $r\,(\ll p)$ dimensional compression of the high-dimensional feature $\mathbf{x}_i$ as well as the low-dimensional feature $\mathbf{x}_i'$ as-is. We assume such compression is done by some *latent basis* or *dictionary* $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_r] \in \mathbb{R}^{p \times r}$ that is *reconstructive* in the sense that the observed signals $\mathbf{x}_i$ can be reconstructed as (or approximated by) the linear transform of the 'atoms' $\mathbf{w}_1, \ldots, \mathbf{w}_r \in \mathbb{R}^p$ for some suitable 'code' $\mathbf{h}_i \in \mathbb{R}^r$. More concisely, $\mathbf{X}_{\text{data}} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \approx \mathbf{W}\mathbf{H}$, where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{r \times n}$. In practice, we can choose r to be the approximate rank of data matrix $\mathbf{X}_{\text{data}}$ (e.g., by finding the elbow of the scree plot).

Now, we state our probabilistic modeling assumption. Fix parameters $\mathbf{W} \in \mathbb{R}^{p \times r}$, $\mathbf{h}_i \in \mathbb{R}^r$, $\boldsymbol{\beta} \in \mathbb{R}^{r \times \kappa}$, and $\boldsymbol{\gamma} \in \mathbb{R}^{q \times \kappa}$. We assume y_i is a realization of a random variable whose conditional distribution is specified as $[\mathbb{P}(y_i = 0 \,|\, \mathbf{x}_i, \mathbf{x}_i'), \ldots, \mathbb{P}(y_i = \kappa \,|\, \mathbf{x}_i, \mathbf{x}_i')] = \mathbf{g}(\mathbf{a}_i) := C[1, \exp(\mathbf{a}_{i,1}), \ldots, \exp(\mathbf{a}_{i,\kappa})]$, where C is the normalization constant and $\mathbf{a}_i = (\mathbf{a}_{i,1}, \ldots, \mathbf{a}_{i,\kappa}) \in \mathbb{R}^\kappa$ is the *activation* for y_i defined in two ways, depending on whether we use a 'feature-based' or 'filter-based' SDL model:

$$\mathbf{a}_i = \begin{cases} \boldsymbol{\beta}^T \mathbf{h}_i + \boldsymbol{\gamma}^T \mathbf{x}_i' & \text{feature-based (SDL-}\mathbf{H}\text{)}, \\ \boldsymbol{\beta}^T \mathbf{W}^T \mathbf{x}_i + \boldsymbol{\gamma}^T \mathbf{x}_i' & \text{filter-based (SDL-}\mathbf{W}\text{)}. \end{cases} \quad (1)$$

One may regard $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ as the 'multinomial regression coefficients' with input feature $(\mathbf{h}_i, \mathbf{x}_i')$ or $(\mathbf{W}^T\mathbf{x}_i, \mathbf{x}_i')$. We regard the code $\mathbf{h}_i$ (coming from $\mathbf{x}_i \approx \mathbf{W}\mathbf{h}_i$) or the 'filtered signal' $\mathbf{W}^T\mathbf{x}_i$ as the r-dimensional compression of $\mathbf{x}_i$.

In order to estimate the model parameters $(\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\gamma})$

from observed training data $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, n$, we consider the following multi-objective optimization:

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{i=1}^{n} \ell(y_i, \mathbf{a}_i) + \xi \|\mathbf{X}_{\text{data}} - \mathbf{W}\mathbf{H}\|_F^2, \qquad (2)$$

where $\mathbf{X}_{\text{data}} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$, $\mathbf{a}_i$ is as in (1), and $\ell(\cdot)$ is the classification loss measured by the negative log-likelihood: $\ell(y, \mathbf{a}) = \log \sum_{c=1}^{\kappa} \exp(a_c) - \sum_{c=1}^{\kappa} \mathbf{1}_{\{y_i=c\}} a_c$. In (2), the *tuning parameter* ξ controls the trade-off between the two objectives of classification and dictionary learning. The above is a nonconvex problem involving four blocks of parameters that could have additional constraints (e.g., bounded norm). In Figure 2, we will demonstrate that the best reconstructive dictionary $\mathbf{W}$ could be significantly different from the supervised dictionary learned by SDL and may not be very effective for the classification tasks.

2.2. Sketch of algorithm

Our key idea to solve (2) is to transform it into a variant of the low-rank matrix estimation problem (3) and then use a *Low-rank Projected Gradient Descent* (LPGD) algorithm (4) ($\tau > 0$ a fixed stepsize):

$$\min_{\mathbf{Z}=[\boldsymbol{\theta},\boldsymbol{\gamma}]\in\boldsymbol{\Theta},\, \text{rank}(\boldsymbol{\theta})\le r} F(\mathbf{Z}) \qquad (3)$$

$$\mathbf{Z}_t \leftarrow \Pi_r\big(\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}_{t-1} - \tau\nabla F(\mathbf{Z}_{t-1})\right)\big). \qquad (4)$$

In (3), one seeks to minimize an objective F w.r.t. a paired matrix parameter $\mathbf{Z} = [\boldsymbol{\theta}, \boldsymbol{\gamma}]$ within a convex constraint set $\boldsymbol{\Theta}$ and an additional rank constraint $\text{rank}(\boldsymbol{\theta}) \le r$. In (4), Π_r denotes applying rank-r projection on the first factor $\boldsymbol{\theta}$ while keeping $\boldsymbol{\gamma}$ the same. Below we give a sketch of the steps for applying the above scheme to SDL (2) and we refer to Alg. 1 in the appendix for details. For a more detailed explanation of the algorithm, see Sec. B in the Appendix.

Step 1: Convert SDL problem into a low-rank matrix estimation problem In this step, instead of a four-block, nonconvex optimization problem (2) that is computationally challenging to solve exactly, we consider reformulating it into a problem with a convex objective function with two blocks by suitably stacking up the matrices. For SDL-$\mathbf{H}$, we bind 'rows' of $\boldsymbol{\beta}^T\mathbf{H}$ and $\mathbf{W}\mathbf{H}$ so that we have one matrix $\boldsymbol{\theta} \in \mathbb{R}^{(\kappa+p)\times n}$ in (3) instead of three factors $\boldsymbol{\beta}, \mathbf{W}$, and $\mathbf{H}$ (See Fig. 1 "Training"). Similarly, for SDL-$\mathbf{W}$, we make $\boldsymbol{\theta} \in \mathbb{R}^{p\times(\kappa+n)}$ by binding 'columns' of $\mathbf{W}\boldsymbol{\beta}$ and $\mathbf{W}\mathbf{H}$. $\boldsymbol{\gamma}^T\mathbf{x}_i'$ remains the same in $\mathbf{a}_i$ in (2).

Step 2: Apply LPGD algorithm To solve (3) with properly defined $\boldsymbol{\theta}$ through step 1, we propose to use the LGPD algorithm (4). We iterate gradient descent followed by projecting onto the convex constraint set $\boldsymbol{\Theta}$ of the combined factor $[\boldsymbol{\theta}, \boldsymbol{\gamma}]$ and then perform a rank-r projection of the first factor $\boldsymbol{\theta}$ via the truncated singular value decomposition (SVD) until convergence.

Step 3: Decompose the lifted solution Since we stack matrices to reformulate the problem in step 1, it is necessary to recover the original three factors $\mathbf{W}$, $\mathbf{H}$, and $\boldsymbol{\beta}$. Once we have a solution $[\boldsymbol{\theta}^\star, \boldsymbol{\gamma}^\star]$ from step 2, we can implement rank-r SVD of $\boldsymbol{\theta}^\star$ to obtain a solution to (2). Let $\boldsymbol{\theta}^\star = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ denote the SVD of $\boldsymbol{\theta}$. For SDL-$\mathbf{H}$, as $\boldsymbol{\theta}^\star$ constitutes the row-binded matrix of $(\boldsymbol{\beta}^\star)^T\mathbf{H}^\star$ and $\mathbf{W}^\star\mathbf{H}^\star$, we can assign $\mathbf{H}^\star = \boldsymbol{\Sigma}^{1/2}\mathbf{V}^T$ while the row binding of $[(\boldsymbol{\beta}^\star)^T, \mathbf{W}^\star] = \mathbf{U}\boldsymbol{\Sigma}^{1/2}$. Similarly, in the case of SDL-$\mathbf{W}$, we assign $\mathbf{W}^\star = \mathbf{U}$ and column binding of $[\boldsymbol{\beta}_\star,\ \mathbf{H}_\star] = \boldsymbol{\Sigma}\mathbf{V}^T$.

2.3. Exponentially convergence to the global minimum

Our main result, Theorem 2.1, establishes that the algorithm introduced in Sec. 2.2 can obtain optimal parameters, up to rotation, that globally minimize the objective function at an exponential rate. With technical assumptions of bounded activation in (1) and bounded eigenvalues of the covariance matrix of the features, we have the following Theorem.

Theorem 2.1. *(Exponential convergence) Let* $\mathbf{Z}_t := [\boldsymbol{\theta}_t, \boldsymbol{\gamma}_t]$ *denote the iterates of* (4) *for SDL and* μ *and* L *be a strongly convex parameter and smoothness parameter of* F*, respectively (see* (16) *in the appendix). Fix* $\tau \in (\frac{1}{2\mu}, \frac{3}{2L})$*, and let* $\rho := 2(1-\tau\mu) \in (0,1)$*. Suppose* $L/\mu < 3$ *and let* $\mathbf{Z}^* = [\boldsymbol{\theta}^*, \boldsymbol{\gamma}^*]$ *be any stationary point of* F *over* $\boldsymbol{\Theta}$ *s.t.* $\text{rank}(\boldsymbol{\theta}^*) \le r$*. Then* $\mathbf{Z}^*$ *is the unique global minimizer of* F *among all* $\mathbf{Z} = [\boldsymbol{\theta}, \boldsymbol{\gamma}]$ *with* $\text{rank}(\boldsymbol{\theta}) \le r$*. Moreover,* $\|\mathbf{Z}_t - \mathbf{Z}^*\|_F \le \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^*\|_F$ *for* $t \ge 1$.

While the LPGD algorithm is in general more expensive per iteration than the nonconvex method by using truncated SVD, the iteration complexity is only $O(\log \epsilon^{-1})$ thanks to the exponential convergence to the global optimum. Hence for ϵ small enough, our algorithm achieves an ϵ-accurate global optimum for SDL with a total computational cost comparable to a nonconvex SDL algorithm to achieve at best an ϵ-stationary point with $O(\epsilon^{-1})$. See Appendix F for the proof of Theorem 2.1 and Sec. I for numerical validations.

3. Application: Microarray Analysis for Cancer Classification

We apply the proposed methods to two datasets from the Curated Microarray Database (CuMiDa) (Feltes et al., 2019). CuMiDa provides well-preprocessed microarray data for various cancer types for various machine-learning approaches. One consists of 54,676 gene expressions from 51 subjects with binary labels indicating pancreatic cancer; The other we use has 35,982 gene expressions from 289 subjects with binary labels indicating breast cancer. The primary purpose of the analysis is to classify cancer patients solely based on their gene expression. We compare the accuracies of the proposed methods – SDL-$\mathbf{W}$ and SDL-$\mathbf{H}$ with a binary logistic classifier trained using our algorithm – against the following benchmark algorithms: SDL-$\mathbf{W}$ and

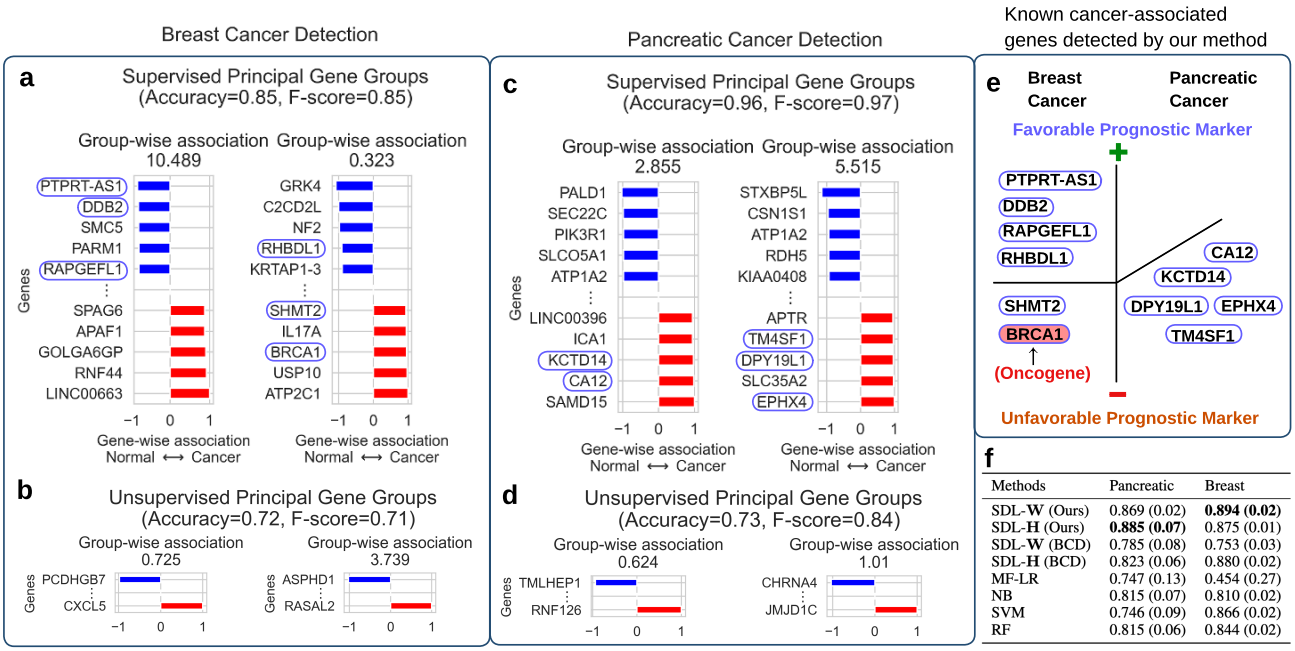


Figure 2. (**a-b**) Two selected supervised/unsupervised principal gene groups (low-dimensional compression of genes) learned by rank-16 SDL-**W**/SVD and their associated logistic regression coefficients for breast cancer detection. (**c-d**) Similar to **a-b** learned by rank-2 SDL-**W**/SVD for pancreatic cancer detection. **(e)** Blue-circled genes within each gene group of extreme coefficients coincide with known prognostic markers (for pancreatic cancer) and oncogene (for breast cancer). (**f**) Average classification accuracies and their standard deviations (in parenthesis) for various methods on two cancer microarray datasets over five-fold cross-validation. The highest-performing instances are marked in bold.

SDL-**H** trained using BCD (Grippo & Sciandrone, 2000; Xu & Yin, 2013); Naive Bayes (NB); Support Vector Machine (SVM); Random Forest (RF); Logistic Regression with Matrix Factorization by truncated SVD (MF-LR). For the last benchmark method MF-LR, we use rank-r SVD to factorize $\mathbf{X}_{\text{train}} \approx \mathbf{U}\Sigma\mathbf{V}^T$ and take $\mathbf{W} = \mathbf{U}$ and $\mathbf{H} = \Sigma\mathbf{V}^T$.

We normalize gene expression for stable matrix factorization and interpretability of regression coefficients. We split each data into 50% of the training set and 50% of the test set and repeat the comparison procedure 5 times. A scree plot is used to determine the rank r. Other parameters are chosen through 5-fold cross-validation ($\xi \in \{0.1, 1, 10\}$ and $\lambda \in \{0.1, 1, 10\}$), and the algorithms are repeated in 1,000 iterations or until convergence. As can be seen in the table in Figure 2**f**, the proposed methods show the best performance for both types of cancers.

An important advantage of SDL methods is that they provide interpretable results in the form of supervised dictionaries with associated regression coefficients. In the context of microarray analysis for cancer research, each column of supervised dictionary $\mathbf{W}$ corresponds to a weighted group of genes (which we call a 'principal gene group'), and its corresponding β represents the strength of its association with cancer. SDL learns supervised gene groups (Fig. 2**a, c**) with significantly higher classification accuracy than the unsupervised gene groups (Fig. 2**b, d**). Both gene groups (consisting of p genes) in Fig. 2**a, c** have positive regression coefficients, so they are positively associated with the log odds of the predictive probability of breast/pancreatic cancer. Remarkably, total ten genes (in Fig. 2 **e**) in these groups of extreme coefficients are known to be prognostic markers of pancreatic/breast cancer or well-known oncogene for breast cancer (see Human Protein Atlas (Sjöstedt et al., 2020)). The high classification accuracy, along with findings of oncogene and prognostic markers, suggests a strong association between the identified supervised principal gene groups and cancer. These findings not only demonstrate the effectiveness of the classification model but also provide valuable insights into potential biological discovery.

4. Conclusion

We propose an exponentially convergent algorithm for nonconvex SDL problems using novel lifting techniques. In cancer classification using microarray data analysis, our algorithm successfully identifies discriminative gene groups for pancreatic/breast cancer and shows potential for identifying important gene groups as protein complexes or pathways in biomedical research. Our analysis framework can be extended to more complex classification models, such as combining a feed-forward deep neural network with a dictionary learning objective.

References

Real / fake job posting prediction. https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction. Accessed: 2023-05-16.

Agarwal, A., Negahban, S., and Wainwright, M. J. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. *Advances in Neural Information Processing Systems*, 23, 2010.

Amani, S. and Thrampoulidis, C. Ucb-based algorithms for multinomial logistic regression bandits. *Advances in Neural Information Processing Systems*, 34, 2021.

Austin, W., Anderson, D., and Ghosh, J. Fully supervised non-negative matrix factorization for feature extraction. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5772–5775. IEEE, 2018.

Beck, A. *First-order methods in optimization*. SIAM, 2017.

Böhning, D. Multinomial logistic regression algorithm. *Annals of the institute of Statistical Mathematics*, 44(1): 197–200, 1992.

Chu, M. T., Funderlic, R. E., and Plemmons, R. J. Structured low rank approximation. *Linear algebra and its applications*, 366:157–172, 2003.

Elad, M. and Aharon, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Feltes, B. C., Chandelier, E. B., Grisci, B. I., and Dorn, M. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. *Journal of Computational Biology*, 26(4):376–386, 2019. doi: 10.1089/cmb.2018.0238. URL https://doi.org/10.1089/cmb.2018.0238. PMID: 30789283.

Gangeh, M. J., Fewzee, P., Ghodsi, A., Kamel, M. S., and Karray, F. Multiview supervised dictionary learning in speech emotion recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(6):1056–1068, 2014.

Gangeh, M. J., Farahat, A. K., Ghodsi, A., and Kamel, M. S. Supervised dictionary learning and sparse representation-a review. *arXiv preprint arXiv:1502.05928*, 2015.

Grippo, L. and Sciandrone, M. On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations research letters*, 26(3):127–136, 2000.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Jain, P., Meka, R., and Dhillon, I. Guaranteed rank minimization via singular value projection. *Advances in Neural Information Processing Systems*, 23, 2010.

Jain, P., Netrapalli, P., and Sanghavi, S. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, 2013.

Kim, D., Park, C., Oh, J., Lee, S., and Yu, H. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 233–240, 2016.

Lecué, G. and Mendelson, S. Sparse recovery under weak moment assumptions. *Journal of the European Mathematical Society*, 19(3):881–904, 2017.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Leuschner, J., Schmidt, M., Fernsel, P., Lachmund, D., Boskamp, T., and Maass, P. Supervised non-negative matrix factorization methods for maldi imaging applications. *Bioinformatics*, 35(11):1940–1947, 2019.

Li, Z., Zhang, Z., Qin, J., Zhang, Z., and Shao, L. Discriminative fisher embedding dictionary learning algorithm for object recognition. *IEEE transactions on neural networks and learning systems*, 31(3):786–800, 2019.

Mairal, J., Elad, M., and Sapiro, G. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2007.

Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. Supervised dictionary learning. *Advances in Neural Information Processing Systems*, 21:1033–1040, 2008.

Mairal, J., Bach, F., and Ponce, J. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.

Meka, R., Jain, P., and Dhillon, I. S. Guaranteed rank minimization via singular value projection. *arXiv preprint arXiv:0909.5457*, 2009.

Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5-6):183–197, 1991.

Negahban, S. and Wainwright, M. J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161, 2013.

Park, D., Kyrillidis, A., Bhojanapalli, S., Caramanis, C., and Sanghavi, S. Provable non-convex projected gradient descent for a class of constrained matrix optimization problems. *stat*, 1050:4, 2016.

Park, D., Kyrillidis, A., Carmanis, C., and Sanghavi, S. Non-square matrix sensing without spurious local minima via the burer-monteiro approach. In *Artificial Intelligence and Statistics*, pp. 65–74. PMLR, 2017.

Park, D., Kyrillidis, A., Caramanis, C., and Sanghavi, S. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, 2018.

Peyré, G. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009.

Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Ritchie, A., Balzano, L., Kessler, D., Sripada, C. S., and Scott, C. Supervised pca: A multiobjective approach. *arXiv preprint arXiv:2011.05309*, 2020.

Sjöstedt, E., Zhong, W., Fagerberg, L., Karlsson, M., Mitsios, N., Adori, C., Oksvold, P., Edfors, F., Limiszewska, A., Hikmet, F., et al. An atlas of the protein-coding genes in the human, pig, and mouse brain. *Science*, 367(6482): eaay5947, 2020.

Tian, G.-X. and Huang, T.-Z. Inequalities for the minimum eigenvalue of m-matrices. *The Electronic Journal of Linear Algebra*, 20:291–302, 2010.

Tong, T., Ma, C., and Chi, Y. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

Tu, S., Boczar, R., Simchowitz, M., Soltanolkotabi, M., and Recht, B. Low-rank solutions of linear matrix equations via procrustes flow. In *International Conference on Machine Learning*, pp. 964–973. PMLR, 2016.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wang, L., Zhang, X., and Gu, Q. A unified computational and statistical framework for nonconvex low-rank matrix estimation. In *Artificial Intelligence and Statistics*, pp. 981–990. PMLR, 2017.

Wright, S. J. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Xu, Y. and Yin, W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3):1758–1789, 2013.

Yankelevsky, Y. and Elad, M. Structure-aware classification using supervised dictionary learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4421–4425. IEEE, 2017.

Yaskov, P. Controlling the least eigenvalue of a random gram matrix. *Linear Algebra and its Applications*, 504: 108–123, 2016.

Zhang, Q. and Li, B. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 2691–2698. IEEE, 2010.

Zhao, S., Han, J., Lv, J., Jiang, X., Hu, X., Zhao, Y., Ge, B., Guo, L., and Liu, T. Supervised dictionary learning for inferring concurrent brain networks. *IEEE transactions on medical imaging*, 34(10):2036–2045, 2015a.

Zhao, T., Wang, Z., and Liu, H. A nonconvex optimization framework for low rank matrix estimation. *Advances in Neural Information Processing Systems*, 28:559, 2015b.

Zheng, Q. and Lafferty, J. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. *arXiv preprint arXiv:1506.06081*, 2015.

A. Recap of the model: SDL-H and SDL-W

Suppose we are given with n labeled signals $(y_i, \mathbf{x}_i, \mathbf{x}'_i)$ for $i = 1, \dots, n$, where $y_i \in \{0, 1, \dots, \kappa\}$ is the label, $\mathbf{x}_i \in \mathbb{R}^p$ is a high-dimensional feature of i, and $\mathbf{x}'_i \in \mathbb{R}^q$ is a low-dimensional auxiliary feature of i ($p \gg q$). When making predictions of y_i, we use a suitable r ($\ll p$) dimensional compression of the high-dimensional feature $\mathbf{x}_i$ as well as the low-dimensional feature $\mathbf{x}'_i$ as-is. We assume such compression is done by some *latent basis* or *dictionary* $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{p \times r}$ that is *reconstructive* in the sense that the observed signals $\mathbf{x}_i$ can be reconstructed as (or approximated by) the linear transform of the 'atoms' $\mathbf{w}_1, \dots, \mathbf{w}_r \in \mathbb{R}^p$ for some suitable 'code' $\mathbf{h}_i \in \mathbb{R}^r$. More concisely, $\mathbf{X}_{\text{data}} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \approx \mathbf{WH}$, where $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_n] \in \mathbb{R}^{r \times n}$. In practice, we can choose r to be the approximate rank of data matrix $\mathbf{X}_{\text{data}}$.

Fix parameters $\mathbf{W} \in \mathbb{R}^{p \times r}$, $\mathbf{h}_i \in \mathbb{R}^r$, $\boldsymbol{\beta} \in \mathbb{R}^{r \times \kappa}$, and $\boldsymbol{\gamma} \in \mathbb{R}^{q \times \kappa}$. Let $h : \mathbb{R} \to [0, \infty)$ be a *score function* (e.g., $h(\cdot) = \exp(\cdot)$ for multinomial logistic regression)[1]. We assume the class label y_i is a realization of a random variable whose conditional distribution is specified as

$$[\mathbb{P}(y_i = 0 \,|\, \mathbf{x}_i, \mathbf{x}'_i), \dots, \mathbb{P}(y_i = \kappa \,|\, \mathbf{x}_i, \mathbf{x}'_i)] = \mathbf{g}(\mathbf{a}_i) := C[1, h(\mathbf{a}_{i,1}), \dots, h(\mathbf{a}_{i,\kappa})],$$

where C is the normalization constant and $\mathbf{a}_i = (\mathbf{a}_{i,1}, \dots, \mathbf{a}_{i,\kappa}) \in \mathbb{R}^\kappa$ is the *activation* for y_i defined in two ways, depending on whether we use a 'feature-based' or 'filter-based' SDL model:

$$\mathbf{a}_i = \begin{cases} \boldsymbol{\beta}^T \mathbf{h}_i + \boldsymbol{\gamma}^T \mathbf{x}'_i & \text{feature-based (SDL-}\mathbf{H}\text{)}, \\ \boldsymbol{\beta}^T \mathbf{W}^T \mathbf{x}_i + \boldsymbol{\gamma}^T \mathbf{x}'_i & \text{filter-based (SDL-}\mathbf{W}\text{)}. \end{cases}$$

One may regard $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ as the 'multinomial regression coefficients' with input feature $(\mathbf{h}_i, \mathbf{x}'_i)$ or $(\mathbf{W}^T \mathbf{x}_i, \mathbf{x}'_i)$. In (1), we may regard the code $\mathbf{h}_i$ (coming from $\mathbf{x}_i \approx \mathbf{W}\mathbf{h}_i$) or the 'filtered signal' $\mathbf{W}^T \mathbf{x}_i$ as the r-dimensional compression of $\mathbf{x}_i$. Note that these two coincide if we have perfect factorization $\mathbf{x}_i = \mathbf{W}\mathbf{h}_i$ and the dictionary $\mathbf{W}$ is orthonormal, i.e., $\mathbf{W}^T \mathbf{W} = \mathbf{I}_r$, but we do not necessarily make such an assumption.

In order to estimate the model parameters $(\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ from observed training data $(\mathbf{x}_i, y_i)$ for $i = 1, \dots, n$, we consider the following multi-objective optimization:

$$\min_{\mathbf{W}, \mathbf{H}, \boldsymbol{\beta}, \boldsymbol{\gamma}} \sum_{i=1}^{n} \ell(y_i, \mathbf{a}_i) + \xi \|\mathbf{X}_{\text{data}} - \mathbf{WH}\|_F^2,$$

where $\mathbf{X}_{\text{data}} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$, $\mathbf{a}_i$ is as in (1), and $\ell(\cdot)$ is the classification loss measured by the negative log-likelihood:

$$\ell(y, \mathbf{a}) = \log \sum_{c=1}^{\kappa} h(a_c) - \sum_{c=1}^{\kappa} \mathbf{1}_{\{y_i = c\}} \log h(a_c). \tag{5}$$

In (2), the *tuning parameter* ξ controls the trade-off between the two objectives of classification and dictionary learning. The above is a nonconvex problem involving four blocks of parameters that could have additional constraints (e.g., bounded norm).

There are some notable differences between SDL-$\mathbf{H}$ and SDL-$\mathbf{W}$ when predicting the unknown label of a test point. If we are given a test point $(\mathbf{x}_{\text{test}}, \mathbf{x}'_{\text{test}})$, the predictive probabilities for its unknown label y_{test} is given by (5) with activation $\mathbf{a}$ computed as in (1). This only involves straightforward matrix multiplications for SDL-$\mathbf{W}$, which can also be viewed as a forward propagation in a multilayer perceptron (Murtagh, 1991) with $\mathbf{W}$ acting as the first layer weight matrix (hence named 'filter'). However, for SDL-$\mathbf{H}$, one needs to solve additional optimization problems for testing. Namely, for every single test signal $\mathbf{x}_{\text{test}}$, its correct code representation $\mathbf{h}_{\text{test}}$ needs to be learned by solving the following 'supervised sparse coding' problem (see (Mairal et al., 2008)):

$$\min_{y \in \{0, 1, \dots, \kappa\}} \min_{\mathbf{h}} \ell(y, \boldsymbol{\beta}^T \mathbf{h}) + \xi \|\mathbf{x}_{\text{test}} - \mathbf{W}\mathbf{h}\|_F^2. \tag{6}$$

A more efficient heuristic testing method for SDL-$\mathbf{H}$ is by approximately computing $\mathbf{h}_{\text{test}}$ by only minimizing the second term in (6).

[1] Notice that in (5), we have used a general score function h instead of the exponential function as we did in the main text. We will analyze this more general SDL model in this appendix. See Section D for background on multinomial logistic regression with general score function.

A.1. Notations

Throughout this paper, we denote by $\mathbb{R}^p$ the ambient space for data equipped with standard inner project $\langle \cdot, \cdot \rangle$ that induces the Euclidean norm $\|\cdot\|$. We denote by $\{0, 1, \ldots, \kappa\}$ the space of class labels with $\kappa + 1$ classes. For a convex subset $\boldsymbol{\Theta}$ in a Euclidean space, we denote $\Pi_{\boldsymbol{\Theta}}$ the projection operator onto $\boldsymbol{\Theta}$. For an integer $r \geq 1$, we denote by Π_r the rank-r projection operator for matrices. For a matrix $\mathbf{A} = (a_{ij})_{ij} \in \mathbb{R}^{m \times n}$, we denote its Frobenius, operator (2-), and supremum norm by $\|\mathbf{A}\|_F^2 := \sum_{i,j} a_{ij}^2, \|\mathbf{A}\|_2 := \sup_{\mathbf{x} \in \mathbb{R}^n,\, \|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|, \|\mathbf{A}\|_\infty := \max_{i,j} |a_{ij}|$, respectively. For each $1 \leq i \leq m$ and $1 \leq j \leq n$, we denote $\mathbf{A}[i, :]$ and $\mathbf{A}[:, j]$ for the ith row and the jth column of $\mathbf{A}$, respectively. For each integer $n \geq 1$, $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. For square symmetric matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, we denote $\mathbf{A} \preceq \mathbf{B}$ if $\mathbf{v}^T \mathbf{A} \mathbf{v} \leq \mathbf{v}^T \mathbf{B} \mathbf{v}$ for all unit vectors $\mathbf{v} \in \mathbb{R}^n$. For two matrices $\mathbf{A}$ and $\mathbf{B}$, we denote $[\mathbf{A}, \mathbf{B}]$ and $[\mathbf{A} \parallel \mathbf{B}]$ the matrices obtained by concatenating (stacking) them by horizontally and vertically, respectively, assuming matching dimensions.

B. Statement of the algorithm and key idea

In the main text, we mentioned that our key idea to solve (2) is to transform it into a variant of the low-rank matrix estimation problem (3)

$$\min_{\mathbf{Z}=[\boldsymbol{\theta},\boldsymbol{\gamma}] \in \boldsymbol{\Theta},\, \mathrm{rank}(\boldsymbol{\theta}) \leq r} F(\mathbf{Z}),$$

where one seeks to minimize an objective f w.r.t. a paired matrix parameter $\mathbf{Z} = [\boldsymbol{\theta}, \boldsymbol{\gamma}]$ within a convex constraint set $\boldsymbol{\Theta}$ and an additional rank constraint $\mathrm{rank}(\boldsymbol{\theta}) \leq r$. Then, we use a *Low-rank Projected Gradient Descent* (LPGD) algorithm (4) ($\tau > 0$ a fixed stepsize) to solve the transformed problem (3)

$$\mathbf{Z}_t \leftarrow \Pi_r\big(\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}_{t-1} - \tau \nabla F(\mathbf{Z}_{t-1})\right)\big).$$

B.1. Illustration of the key idea: Double-lifting

To illustrate the transformation of the SDL problem (2) into a low-rank matrix estimation (3), first assume we have no augmented variable $\boldsymbol{\gamma}$. Then consider a much simpler version of SDL-$\mathbf{H}$ where the response variable y is scalar and continuous and Namely, we replace the multi-class classification problem (2) with **linear regression**. We seek to solve matrix factorization and linear regression problems simultaneously for data matrix $\mathbf{X}_{\text{data}} \in \mathbb{R}^{p \times n}$ and response variable $\mathbf{Y} \in \mathbb{R}^{1 \times n}$: $\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} \|\mathbf{Y} - \boldsymbol{\beta}^T \mathbf{H}\|_F^2 + \xi \|\mathbf{X}_{\text{data}} - \mathbf{W}\mathbf{H}\|_F^2$. This is a three-block optimization problem involving three factors $\mathbf{W} \in \mathbb{R}^{p \times r}, \mathbf{H} \in \mathbb{R}^{r \times n}$ and $\boldsymbol{\beta} \in \mathbb{R}^{r \times 1}$, which is nonconvex and computationally challenging to solve exactly. Instead, consider reformulating the above nonconvex problem (7) into a problem with a convex objective function by suitably stacking up the matrices using the following matrix factorization:

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} f\left(\begin{bmatrix}\boldsymbol{\beta}^T \\ \mathbf{W}\end{bmatrix}\mathbf{H}\right) := \left\|\begin{bmatrix}\mathbf{Y} \\ \sqrt{\xi}\mathbf{X}_{\text{data}}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\beta}^T \\ \sqrt{\xi}\mathbf{W}\end{bmatrix}\mathbf{H}\right\|_F^2. \tag{7}$$

Indeed, we now seek to find *two* decoupled matrices (instead of three), one for $\boldsymbol{\beta}^T$ and $\mathbf{W}$ stacked vertically, and the other for $\mathbf{H}$. The idea of matrix stacking was used in (Zhang & Li, 2010) for discriminative K-SVD. Proceeding one step further, another important observation we make is that it is also equivalent to finding a *single* matrix $\boldsymbol{\theta} := \left[\boldsymbol{\beta}^T\mathbf{H} \parallel \mathbf{W}\mathbf{H}\right] \in \mathbb{R}^{(1+p)\times n}$ of rank at most r that minimizes the function f in (7): (See Fig. 1 Training).

For SDL-W, consider the following analogous linear regression model:

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} f\left(\mathbf{W}[\boldsymbol{\beta}, \mathbf{H}]\right) := \|\mathbf{Y} - \boldsymbol{\beta}^T \mathbf{W}^T \mathbf{X}_{\text{data}}\|_F^2 + \xi \|\mathbf{X}_{\text{data}} - \mathbf{W}\mathbf{H}\|_F^2, \tag{8}$$

where the right-hand side above is obtained by replacing $\mathbf{H}$ with $\mathbf{W}^T\mathbf{X}_{\text{data}}$ in (7). Note that the objective function depends only on the product of the two matrices $\mathbf{W}$ and $[\boldsymbol{\beta}, \mathbf{H}]$. Then, we may further lift it as the low-rank matrix estimation problem by seeking a single matrix $\boldsymbol{\theta} := [\mathbf{W}\boldsymbol{\beta},\ \mathbf{W}\mathbf{H}] \in \mathbb{R}^{p \times (1+n)}$ of rank at most r that solves (3) with f being the function in (8).

B.2. Statement of the algorithm

Motivated by the observation we made in Section B.1, we rewrite SDL-$\mathbf{H}$ in (2) as

$$\min_{\substack{[\boldsymbol{\theta},\boldsymbol{\gamma}]\in\boldsymbol{\Theta}\\ \operatorname{rank}(\boldsymbol{\theta})\le r}} F(\boldsymbol{\theta},\boldsymbol{\gamma}) := \sum_{i=1}^{n} \ell(y_i, \mathbf{A}[:,i] + \boldsymbol{\gamma}^T\mathbf{x}'_i) + \xi\|\mathbf{X}_{\text{data}} - \mathbf{B}\|_F^2 + \lambda\left(\|\mathbf{A}\|_F^2 + \|\boldsymbol{\gamma}\|_F^2\right), \tag{9}$$

where $\mathbf{A} = \boldsymbol{\beta}^T\mathbf{H}$, $\mathbf{B} = \mathbf{W}\mathbf{H}$, $\boldsymbol{\theta} = [\mathbf{A} \,\|\, \mathbf{B}] \in \mathbb{R}^{(\kappa+p)\times n}$, and $\boldsymbol{\Theta}$ is a convex subset of $\mathbb{R}^{(\kappa+p)\times n} \times \mathbb{R}^{q\times\kappa}$. The last quadratic term above is the L_2-regularization term for $\mathbf{A}$ and $\boldsymbol{\gamma}$ with coefficient $\lambda \ge 0$, which plays a crucial role in well-conditioning (9). As for SDL-$\mathbf{W}$, we can rewrite (2) with additional L_2-regularizer for $\mathbf{A} = \mathbf{W}\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ as

$$\min_{\substack{[\boldsymbol{\theta},\boldsymbol{\gamma}]\in\boldsymbol{\Theta}\\ \operatorname{rank}(\boldsymbol{\theta})\le r}} F(\boldsymbol{\theta},\boldsymbol{\gamma}) = \sum_{i=1}^{n} \ell(y_i, \mathbf{A}^T\mathbf{x}_i + \boldsymbol{\gamma}^T\mathbf{x}'_i) + \xi\|\mathbf{X}_{\text{data}} - \mathbf{B}\|_F^2 + \lambda\left(\|\mathbf{A}\|_F^2 + \|\boldsymbol{\gamma}\|_F^2\right), \tag{10}$$

where $\boldsymbol{\theta} = [\mathbf{A}, \mathbf{B}] = \mathbf{W}[\boldsymbol{\beta}, \mathbf{H}] \in \mathbb{R}^{p\times(\kappa+n)}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{p\times(\kappa+n)} \times \mathbb{R}^{q\times\kappa}$ is a convex set.

For solving (9), we propose to use the LGPD algorithm (4): *We iterate gradient descent followed by projecting onto the convex constraint set $\boldsymbol{\Theta}$ of the combined factor $[\boldsymbol{\theta},\boldsymbol{\gamma}]$ and then perform rank-r projection of the first factor $\boldsymbol{\theta} = [\mathbf{A} \,\|\, \mathbf{B}]$ via truncated SVD until convergence.* Once we have a solution $[\boldsymbol{\theta}^\star,\boldsymbol{\gamma}^\star]$ to (9), we can use SVD of $\boldsymbol{\theta}^\star$ to obtain a solution to (2). Let $\boldsymbol{\theta}^\star = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ denote the SVD of $\boldsymbol{\theta}$. Since $\operatorname{rank}(\boldsymbol{\theta}^\star) \le r$, $\boldsymbol{\Sigma}$ is an $r \times r$ diagonal matrix of singular values of $\boldsymbol{\theta}$. Then $\mathbf{U} \in \mathbb{R}^{(\kappa+p)\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$ are semi-orthonormal matrices, that is, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_r$. Then since $\boldsymbol{\theta}^\star = [(\boldsymbol{\beta}^\star)^T \,\|\, \mathbf{W}^\star]\mathbf{H}^\star$, we can take $\mathbf{H}^\star = \boldsymbol{\Sigma}^{1/2}\mathbf{V}^T$ and $[(\boldsymbol{\beta}^\star)^T \,\|\, \mathbf{W}^\star] = \mathbf{U}\boldsymbol{\Sigma}^{1/2}$. Algorithm 1 for SDL-$\mathbf{W}$ follows similar reasoning as before with the reformulation above.

We summarize this approach of solving (2) for SDL-$\mathbf{H}$ in Algorithm 1. Here, SVD_r denotes rank-r truncated SVD and the projection operators $\Pi_{\boldsymbol{\Theta}}$ and Π_r are defined in Subsection A.1.

Algorithm 1 Lifted PGD for SDL

Input: $\mathbf{X}_{\text{data}} \in \mathbb{R}^{p\times n}$; $\mathbf{X}'_{\text{aux}} \in \mathbb{R}^{q\times n}$ (Auxiliary covariates); $\mathbf{Y}_{\text{label}} \in \{0, 1, \ldots, \kappa\}^n$
Parameters: $\tau > 0$ (stepsize); $N \in \mathbb{N}$ (iterations); $r \ge 1$ (rank); $\lambda \ge 0$ (L_2-reg. param.)
Constraints: Convex $\boldsymbol{\Theta} \subseteq \mathbb{R}^{(\kappa+p)\times n} \times \mathbb{R}^{q\times\kappa}$ for SDL-$\mathbf{H}$, $\boldsymbol{\Theta} \subseteq \mathbb{R}^{p\times(\kappa+n)} \times \mathbb{R}^{q\times\kappa}$ for SDL-$\mathbf{W}$;
Initialize $\mathbf{W}_0 \in \mathbb{R}^{p\times r}$, $\mathbf{H}_0 \in \mathbb{R}^{r\times n}$, $\boldsymbol{\beta}_0 \in \mathbb{R}^{r\times\kappa}$, $\boldsymbol{\gamma}_0 \in \mathbb{R}^{q\times\kappa}$
$\begin{cases} \boldsymbol{\theta}_0 \leftarrow [\boldsymbol{\beta}_0^T\mathbf{H}_0 \,\|\, \mathbf{W}_0\mathbf{H}_0] \in \mathbb{R}^{(\kappa+p)\times n} & (\triangleright \text{ for SDL-}\mathbf{H}) \\ \boldsymbol{\theta}_0 \leftarrow [\mathbf{W}_0\boldsymbol{\beta}_0, \mathbf{W}_0\mathbf{H}_0] \in \mathbb{R}^{p\times(\kappa+n)} & (\triangleright \text{ for SDL-}\mathbf{W}) \end{cases}$
for $k = 1$ **to** N **do**
  $\boldsymbol{\theta}_k \leftarrow \Pi_r\left(\Pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}_{k-1} - \tau\nabla_{\boldsymbol{\theta}}F(\boldsymbol{\theta}_{k-1},\boldsymbol{\gamma}_{k-1})\right)\right)$ (▷ See Appendix **??** for computation)
  $\boldsymbol{\gamma}_k \leftarrow \boldsymbol{\gamma}_{k-1} - \tau\nabla_{\boldsymbol{\gamma}}F(\boldsymbol{\theta}_{k-1},\boldsymbol{\gamma}_{k-1})$
end for
$\boldsymbol{\theta}_N = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ (▷ rank-r SVD)
$\begin{cases} [\boldsymbol{\beta}_N^T \,\|\, \mathbf{W}_N] \leftarrow \mathbf{U}\boldsymbol{\Sigma}^{1/2},\ \mathbf{H}_N \leftarrow (\boldsymbol{\Sigma})^{1/2}\mathbf{V}^T & (\triangleright \text{ SDL-}\mathbf{H}) \\ \mathbf{W}_N \leftarrow \mathbf{U},\ [\boldsymbol{\beta}_N,\ \mathbf{H}_N] \leftarrow \boldsymbol{\Sigma}\mathbf{V}^T & (\triangleright \text{ SDL-}\mathbf{W}) \end{cases}$
Output: $(\mathbf{W}_N, \mathbf{H}_N, \boldsymbol{\beta}_N, \boldsymbol{\gamma}_N)$

A straightforward computation shows (recall that $\boldsymbol{\theta} = [\mathbf{A} \,\|\, \mathbf{B}]$ for SDL-$\mathbf{H}$ and $\boldsymbol{\theta} = [\mathbf{A}, \mathbf{B}]$ for SDL-$\mathbf{W}$)

$$\nabla_{\text{vec}(\mathbf{A})}F - 2\lambda\,\text{vec}(\mathbf{A}) = \begin{cases} \sum_{s=1}^{n} \nabla_{\mathbf{a}}\ell(y_s, \mathbf{a}_s) \otimes \mathbf{x}_s & \text{for SDL-}\mathbf{H} \\ \sum_{s=1}^{n} \nabla_{\mathbf{a}}\ell(y_s, \mathbf{a}_s) \otimes \mathbf{I}_n[:,s] & \text{for SDL-}\mathbf{W}, \end{cases} \tag{11}$$

$$\nabla_{\mathbf{B}}F = 2\xi(\mathbf{B} - \mathbf{X}_{\text{data}}), \qquad \nabla_{\text{vec}(\boldsymbol{\gamma})}F = \left(\sum_{s=1}^{n} \nabla_{\mathbf{a}}\ell(y_s, \mathbf{a}_s) \otimes \mathbf{x}'_s\right) + 2\lambda\,\text{vec}(\boldsymbol{\gamma}), \tag{12}$$

where $\otimes$ denotes the Kronecker product. Here, we have $\nabla_{\mathbf{a}}\ell(y, \mathbf{a}) = (\dot{h}_1, \ldots, \dot{h}_\kappa)$, where

$$\dot{h}_j := \frac{h'(a_j)}{1 + \sum_{c=1}^{\kappa} h(a_c)} - \mathbf{1}_{\{y=j\}}\frac{h'(a_j)}{h(a_j)}. \tag{13}$$

By using randomized truncated SVD for the efficient low-rank projection in Algorithm 1, the per-iteration complexity is $O(pn\min(n,p))$, while that for the nonconvex algorithm is $O((pr+q)n)$. While the LPGD algorithm is in general more expensive per iteration than the nonconvex method, the iteration complexity is only $O(\log\epsilon^{-1})$ thanks to the exponential convergence to the global optimum (will be discussed in Theorem 2.1). To our best knowledge, the nonconvex algorithm for SDL does not have any guarantee to converge to a global optimum, and the iteration complexity of the nonconvex SDL method to reach an ϵ-stationary point is at best $O(\epsilon^{-1})$ using standard analysis. Hence for ϵ small enough, Algorithm 1 achieves an ϵ-accurate global optimum for SDL with a total computational cost comparable to a nonconvex SDL algorithm to achieve an ϵ-stationary point.

C. Theoretical guarantees

For theoretical analysis of Algorithm 1, we introduce the following technical assumptions (C.1-C.3).

Assumption C.1. (Bounded activation) The activation $\mathbf{a}\in\mathbb{R}^\kappa$ defined in (1) assumes bounded norm, i.e., $\|\mathbf{a}\|\le M$ for some constant $M\in(0,\infty)$.

Assumption C.2. (Bounded eigenvalues of covariance matrix) Denote $\mathbf{\Phi}=[\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_n]\in\mathbb{R}^{(p+q)\times n}$, where $\boldsymbol{\phi}_i=[\mathbf{x}_i\parallel\mathbf{x}_i']\in\mathbb{R}^{p+q}$ (so $\mathbf{\Phi}=[\mathbf{X}_{\text{data}}\parallel\mathbf{X}_{\text{aux}}]$), where $\mathbf{X}_{\text{aux}}=[\mathbf{x}_1',\dots,\mathbf{x}_n']$. Then, there exist constants $\delta^-,\delta^+>0$ such that for all $n\ge1$,

$$\delta^-\le\lambda_{\min}(n^{-1}\mathbf{\Phi}\mathbf{\Phi}^T)\le\lambda_{\max}(n^{-1}\mathbf{\Phi}\mathbf{\Phi}^T)\le\delta^+. \tag{14}$$

Assumption C.3. (Bounded stiffness and eigenvalues of observed information) The score function $h:\mathbb{R}\to[0,\infty)$ is twice continuously differentiable. Further, let observed information $\ddot{\mathbf{H}}(y,\mathbf{a}):=\nabla_{\mathbf{a}}\nabla_{\mathbf{a}^T}\ell(y,\mathbf{a})$ for y and $\mathbf{a}$. Then, for the constant $M>0$ in Assumption C.1, there are constants $\gamma_{\max},\alpha^-,\alpha^+>0$ s.t. $\gamma_{\max}:=\sup_{\|\mathbf{a}\|<M}\max_{1\le s\le n}\|\nabla_{\mathbf{a}}\ell(y_s,\mathbf{a}_s)\|_\infty$ and

$$\alpha^-:=\inf_{\|\mathbf{a}\|<M}\min_{1\le s\le n}\lambda_{\min}(\ddot{\mathbf{H}}(y_s,\mathbf{a})),\quad \alpha^+:=\sup_{\|\mathbf{a}\|<M}\max_{1\le s\le n}\lambda_{\max}(\ddot{\mathbf{H}}(y_s,\mathbf{a})). \tag{15}$$

Assumption C.1 limits the norm of the activation $\mathbf{a}$ as an input for the classification model in (2) is bounded. This is standard in the literature (see, e.g., (Negahban & Wainwright, 2011; Yaskov, 2016; Lecué & Mendelson, 2017)) in order to uniformly bound the eigenvalues of the Hessian of the (multinomial) logistic regression model. Assumption C.2 introduces uniform bounds on the eigenvalues of the covariance matrix. Assumption C.3 introduces uniform bounds on the eigenvalues of the $\kappa\times\kappa$ observed information as well as the first derivative of the predictive probability distribution (see (Böhning, 1992) and Sec. F for more details). In fact, Assumption C.3 is easily satisfied under Assumption C.1 and the multinomial logistic regression model $h(\cdot)=\exp(\cdot)$, as discussed in the following remark.

Remark D.3. (Multinomial Logistic Classifier) In the special case of a multinomial logistic model with the score function $h(\cdot)=\exp(\cdot)$, we have $h=h'=h''$ so the second term in (29) in the Appendix vanishes, so $\dot{h}_j(y,\mathbf{a})=g_j(\mathbf{a})-\mathbf{1}(y=j)$ and $\ddot{H}(y,\mathbf{a})_{i,j}=g_i(\mathbf{a})\left(\mathbf{1}(i=j)-g_j(\mathbf{a})\right)$. Under Assumption C.1, according to Lemma D.1, we can take $\gamma_{\max}=1+\frac{e^M}{1+e^M+(\kappa-1)e^{-M}}\le2$, $\alpha^-=\frac{e^{-M}}{1+e^{-M}+(\kappa-1)e^M}$, and $\alpha^+=\frac{e^M\left(1+2(\kappa-1)e^M\right)}{(1+e^M+(\kappa-1)e^{-M})^2}$. For binary classification, $\alpha^+\le1/4$.

Now define the following quantities: $\lambda^+:=\lambda_{\max}(n^{-1}\mathbf{X}_{\text{aux}}\mathbf{X}_{\text{aux}}^T)$,

$$\mu:=\begin{cases}\min(2\xi,\,2\lambda+n\delta^-\alpha^-)\\ \min(2\xi,\,2\lambda+\alpha^-)\end{cases},\quad L:=\begin{cases}\max(2\xi,\,2\lambda+n\delta^+\alpha^+) & \text{for SDL-}\mathbf{W}\\ \max(2\xi,2\lambda+\alpha^+\lambda^+n,\,2\lambda+\alpha^+) & \text{for SDL-}\mathbf{H}\end{cases}. \tag{16}$$

C.1. Computational convergence guarantee

Theorem 2.1 in the main text is a special case of the following more general result, specifically when the model is 'correctly specified', allowing the rank-r SDL model to effectively account for the observed data. This implies the existence of a 'low-rank stationary point' of F, as also demonstrated in (Wang et al., 2017). In this section, we prove the following more general result.

Theorem C.4. *(Exponential convergence for SDL) Let $\mathbf{Z}_t:=[\boldsymbol{\theta}_t,\boldsymbol{\gamma}_t]$ denote the iterates of Algorithm 1. Assume C.1, C.2, and C.3 hold. Let μ and L be as in (16), fix stepsize $\tau\in(\frac{1}{2\mu},\frac{3}{2L})$, and let $\rho:=2(1-\tau\mu)\in(0,1)$. Suppose $L/\mu<3$.*

(i) *(Correctly specified case; Theorem 2.1 in the main text) Suppose there exists a stationary point $\mathbf{Z}^*=[\boldsymbol{\theta}^*,\boldsymbol{\gamma}^*]$ of F over the convex constraint set $\mathbf{\Theta}$ s.t. $\text{rank}(\boldsymbol{\theta}^*)\le r$. Then $\mathbf{Z}^*$ is the unique global minimizer of F among all $\mathbf{Z}=[\boldsymbol{\theta},\boldsymbol{\gamma}]$ with*

$\text{rank}(\boldsymbol{\theta}) \le r$. *Moreover,*

$$\|\mathbf{Z}_t - \mathbf{Z}^*\|_F \le \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^*\|_F \qquad \text{for } t \ge 1. \tag{17}$$

(ii) *(Possibly misspecified case) Let* $\mathbf{Z}^\star = [\boldsymbol{\theta}^\star, \boldsymbol{\gamma}^\star]$ *be arbitrary in* $\boldsymbol{\Theta}$ *s.t.* $\text{rank}(\boldsymbol{\theta}^\star) \le r$. *Denote the gradient mapping at* $\mathbf{Z}^\star$ *as* $[\Delta\boldsymbol{\theta}^\star, \Delta\boldsymbol{\Gamma}^\star] := \frac{1}{\tau}\left(\boldsymbol{\theta}^\star - \Pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta}^\star - \tau\nabla F(\mathbf{Z}^\star)\right)$. *Then for* $t \ge 1$,

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \le \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{\tau}{1-\rho}\left(\sqrt{3r}\|\Delta\boldsymbol{\theta}^\star\|_2 + \|\Delta\boldsymbol{\gamma}^\star\|_F\right). \tag{18}$$

Note that we may view the ratio L/μ that appears in Theorem 2.1 as the condition number of the SDL problem in (2), whereas the ratio L^*/μ^* for $\mu^* := \delta^-\alpha^-$ and $L^* := \delta^+\alpha^+$ as the condition number for the multinomial classification problem. These two condition numbers are closely related. First, note that for any given μ^*, L^* and sample size n, we can always make $L/\mu < 3$ by choosing sufficiently large ξ and λ so that Theorem 2.1 holds. However, using large L_2-regularization parameter λ may perturb the original objective in (2) too much that the converged solution may not be close to the optimal solution. Hence, we may want to take λ as small as possible. Setting $\lambda = 0$ leads to

$$\frac{L}{\mu} < 3,\ \lambda = 0 \ \Leftrightarrow\ \begin{cases} 0 < \frac{L^*}{\mu^*} < 3,\ \frac{L^*}{6} < \frac{\xi}{n} < \frac{3\mu^*}{2} & \text{for SDL-}\mathbf{W} \\ \frac{\max(2\xi, \alpha^+\lambda^+ n)}{\min(2\xi, \alpha^- n)} < 3 & \text{for SDL-}\mathbf{H}. \end{cases} \tag{19}$$

First, for SDL-$\mathbf{W}$, if the multinomial classification problem is well-conditioned ($L^*/\mu^* < 3$) and the ratio ξ/n is in the above interval, then SDL-$\mathbf{W}$ enjoys exponential convergence in Theorem 2.1. However, the condition for SDL-$\mathbf{H}$ in (19) is violated for large n, so L_2-regularization is necessary for guaranteeing exponential convergence of SDL-$\mathbf{H}$. Second, suppose no auxiliary covariate is used (e.g., $\mathbf{X}_{\text{aux}} = O$) so that $\lambda^+ = 0$. Then the condition $L/\mu < 3$ in Theorem 2.1 reduces to $\frac{\alpha^+}{4} < \lambda < \frac{\xi}{3}$, which holds for $\lambda, \xi = O(1)$. This contrast is closely related to the statistical robustness of SDL-$\mathbf{H}$ over SDL-$\mathbf{W}$, see Theorem C.5 and the following remark.

The proof of Theorem 2.1 involves two steps: (1) We establish a general exponential convergence result for the general LPGD algorithm (4) in Theorem E.2 in the Appendix. (2) We compute the Hessian eigenvalues of the SDL objectives (9)-(10) and apply the result to obtain Theorem 2.1. The proof contains two challenges: first, the low-rank projection in (4) is not non-expansive in general. To overcome this, we show that the iterates closely approximate certain 'auxiliary iterates' which exhibit exponential convergence towards the global optimum. Secondly, the second-order analysis is highly non-trivial since the SDL problem (2) has a total of four unknown matrix factors that are intertwined through the joint multi-class classification and DL tasks. See Appendix F for the details.

C.2. Statistical estimation guarantee

In this section, we formulate a generative model for SDL (2) and state statistical parameter estimation guarantee. Fix dimensions $p \gg q$, and let $n \ge 1$ be possibly growing sample size, and fix unknown true parameters $\mathbf{B}^\star \in \mathbb{R}^{p\times n}$, $\mathbf{C}^\star \in \mathbb{R}^{q\times n}$, $\boldsymbol{\gamma}^\star \in \mathbb{R}^{q\times\kappa}$. In addition, fix $\mathbf{A}^\star \in \mathbb{R}^{\kappa\times n}$ for SDL-$\mathbf{H}$ and $\mathbf{A}^\star \in \mathbb{R}^{p\times\kappa}$ for SDL-$\mathbf{W}$. Now suppose that class label, data, and auxiliary covariates are drawn i.i.d. according to the following joint distribution:

$$\begin{cases} \mathbf{x}_i = \mathbf{B}^\star[:,i] + \boldsymbol{\varepsilon}_i, \quad \mathbf{x}_i' = \mathbf{C}^\star[:,i] + \boldsymbol{\varepsilon}_i', \\ y_i \,|\, \mathbf{x}_i, \mathbf{x}_i' \sim \text{Multinomial}\big(1, \mathbf{g}\left(\mathbf{a}_i\right)\big), \\ \mathbf{a}_i = \begin{cases} \mathbf{A}^\star[:,i] + (\boldsymbol{\gamma}^\star)^T\mathbf{x}_i' & \textit{SDL-}\mathbf{H}, \\ (\mathbf{A}^\star)^T\mathbf{x}_i + (\boldsymbol{\gamma}^\star)^T\mathbf{x}_i' & \textit{SDL-}\mathbf{W}, \end{cases} \end{cases}, \quad \begin{cases} \text{rank}([\mathbf{A}^\star \parallel \mathbf{B}^\star]) \le r & \textit{for SDL-}\mathbf{H}, \\ \text{rank}([\mathbf{A}^\star, \mathbf{B}^\star]) \le r & \textit{for SDL-}\mathbf{W}. \end{cases} \tag{20}$$

where each $\boldsymbol{\varepsilon}_i$ (resp., $\boldsymbol{\varepsilon}_i'$) are $p \times 1$ (resp., $q \times 1$) vector of i.i.d. mean zero Gaussian entries with variance σ^2 (resp., $(\sigma')^2$). We call the above the *generative SDL model*. In what follows, we will assume that the noise levels σ and σ' are known and focus on estimating the four-parameter matrices.

The (L_2-regularized) normalized negative log-likelihood of observing triples $(y_i, \mathbf{x}_i, \mathbf{x}_i')$ for $i = 1, \dots, n$ is given as $\mathcal{L}_n := F(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) + \frac{1}{2(\sigma')^2}\|\mathbf{X}_{\text{aux}} - \mathbf{C}\|_F^2 + c$, where c is a constant and F is as in (9) or (10) depending on the activation type with tuning parameter $\xi = \frac{1}{2\sigma^2}$. The L_2 regularizer in F can be understood as Gaussian prior for the parameters and interpreting the right-hand side above as the negative logarithm of the posterior distribution function (up to a constant) in a

Bayesian framework. Note that the problem of estimating $\mathbf{A}$ and $\mathbf{B}$ are coupled due to the low-rank model assumption in (20), while the problem of estimating $\mathbf{C}$ is standard and separable, so it is not of our interest. The joint estimation problem for $[\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}]$ is equivalent to the corresponding SDL problem (2) with tuning parameter $\xi = (2\sigma^2)^{-1}$. This and Theorem 2.1 motivate us to estimate the true parameters $\mathbf{A}^\star$, $\mathbf{B}^\star$, and $\boldsymbol{\gamma}^\star$ by the output of Algorithm 1 with $\xi = (2\sigma^2)^{-1}$ for $O(\log n)$ iterations.

Now we give the second main result. Roughly speaking, it states that the estimated parameter $\mathbf{Z}_t$ is within the true parameter $\mathbf{Z}^\star = [\mathbf{A}^\star, \mathbf{B}^\star, \boldsymbol{\gamma}^\star]$ within $O(1/\sqrt{n})$ with high probability, provided that the noise variance σ^2 is small enough and the SDL objective (9)-(10) is well-conditioned.

Theorem C.5. *(Statistical estimation for SDL) Assume the model (20) with fixed p. Suppose Assumptions C.1, C.2, and C.3 hold. Let μ, L be as in (16), $\rho := 2(1-\tau\mu)$ and $c = O(1)$ if $\mathbf{Z}^\star - \tau\nabla_{\mathbf{Z}}\mathcal{L}_n(\mathbf{Z}^\star) \in \boldsymbol{\Theta}$ and $c = O(\sqrt{\min(p,n)})$ otherwise. Let $\mathbf{Z}_t$ denote the iterates of Algorithm 1 with the tuning parameter $\xi = (2\sigma^2)^{-1}$, L_2-regularization parameter $\lambda > 0$, and stepsize $\tau \in (\frac{1}{2\mu}, \frac{3}{2L})$. The following holds with probability at least $1 - \frac{1}{n}$: For all $t \geq 1$ and $n \geq 1$, $\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F - \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F \leq c\frac{(\sqrt{n}+\lambda)}{\mu}$, provided $L/\mu < 3$. In particular, the upper bound $c\frac{(\sqrt{n}+\lambda)}{\mu}$ is $O(1/\sqrt{n})$ if $\sigma^{-2} = O(1/n)$.*

We remark that Theorem C.5 implies that *SDL-$\mathbf{H}$ is statistically more robust than SDL-$\mathbf{W}$* in the absence of auxiliary covariates. Namely, in order to have an arbitrary accurate estimate with high probability, one needs to have $1/\mu = o(\sqrt{n})$. Combining with the expression in (16) and the well-balancing condition $L/\mu < 3$, one needs to require small noise variance $\sigma^2 = O(1/n)$ for SDL-$\mathbf{W}$. However, for SDL-$\mathbf{H}$, this is guaranteed whenever $\sigma^2 = o(1/\sqrt{n})$, in case there is no auxiliary covariate (i.e., $\lambda^+ = 0$) and moderate regularization $\lambda = o(1/\mu)$.

C.3. Work related to our theoretical contribution

The SDL training problem (2) is a nonconvex and possibly constrained optimization problem, generally with non-unique minimizers. Since it is difficult to solve exactly, approximate procedures such as *Block Coordinate Descent* (BCD) (see, e.g., (Wright, 2015)) are often used. Such methods utilize the fact that the objective function in (2) is convex in each of the four (matrix) variables. Such an algorithm proceeds by iteratively optimizing for only one block while fixing the others (see (Mairal et al., 2008; Austin et al., 2018; Leuschner et al., 2019; Ritchie et al., 2020)). However, convergence analysis or statistical estimation bounds of such algorithms are quite limited. Appealing to general convergence results for BCD methods (e.g., (Grippo & Sciandrone, 2000; Xu & Yin, 2013)), one can at most guarantee asymptotic convergence to the stationary points or polynomial convergence to Nash equilibria or of the objective (2), modulo carefully verifying the assumptions of these general results. We also remark that (Mairal et al., 2011) provided a rigorous justification of the differentiability of a feature-based SDL model.

The main finding of our work is that the non-convexity of the SDL problem (2) is 'benign', in the sense that *there exists an algorithm globally convergent to a global optimum at an exponential rate.* We use a 'double-lifting' technique that converts the non-convex SDL problem (2) into a low-rank factored estimation with a convex objective. This is reminiscent of the tight relation between a low-rank matrix estimation and a nonconvex factored estimation problem, which has been actively employed in a body of works in statistics and optimization (Agarwal et al., 2010; Ravikumar et al., 2011; Negahban & Wainwright, 2011; Zheng & Lafferty, 2015; Tu et al., 2016; Wang et al., 2017; Park et al., 2017; 2018; Tong et al., 2021). Our exponentially convergent SDL algorithms are versions of low-rank projected gradient descent in the algorithm (43) that operates in the double-lifted space.

D. Generalized multinomial logistic Regression

In this section, we provide some background on a generalized multinomial logistic regression and record some useful computations. (See (Böhning, 1992) for backgrounds on multinomial logistic regression.) Without loss of generality, we can assume that the κ classes are the integers in $\{1, 2, \ldots, \kappa\}$. Say we have training examples $(\boldsymbol{\phi}(\mathbf{x}_1), y_1), \ldots, (\boldsymbol{\phi}(\mathbf{x}_N), y_N)$, where

- $\mathbf{x}_1, \ldots, \mathbf{x}_N$: Input data (e.g., collection of all medical records of each patient)
- $\boldsymbol{\phi}_i := \boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}_N := \boldsymbol{\phi}(\mathbf{x}_N) \in \mathbb{R}^p$: Features (e.g., some useful information for each patient)
- $y_1, \ldots, y_n \in \{0, 1, \ldots, \kappa\}$: κ class labels (e.g., digits from 0 to 9).

The basic idea of multinomial logistic regression is to model the output y as a discrete random variable Y with probability mass function $\mathbf{p} = [p_0, p_1, \ldots, p_\kappa]$ that depends on the observed feature $\boldsymbol{\phi}(\mathbf{x})$, link function $h : \mathbb{R} \to \mathbb{R}$, and a parameter $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_\kappa] \in \mathbb{R}^{p\times\kappa}$ through the following relation:

$$p_0 = \frac{1}{1 + \sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_c\rangle)}, \qquad p_j = \frac{h(\langle \boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_i\rangle)}{1 + \sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_c\rangle)}, \qquad \text{for } j = 1, \ldots, \kappa. \tag{21}$$

That is, given the feature vector $\boldsymbol{\phi}(\mathbf{x})$, the probability p_i of $\mathbf{x}$ having label i is proportional to h evaluated at the 'linear activation' $\langle \boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_i\rangle$. Note that using $h(x) = \exp(x)$, the above multiclass classification model reduces to the classical multinomial logistic regression. In this case, the corresponding predictive probability distribution $\mathbf{p}$ is called the *softmax distribution* with activation $\mathbf{a} = [a_1, \ldots, a_\kappa]$ with $a_i = \langle \boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_i\rangle$ for $i = 1, \ldots, \kappa$. Notice that this model has parameter vectors $\mathbf{w}_1, \ldots, \mathbf{w}_\kappa \in \mathbb{R}^p$, one for each of the κ nonzero class labels.

Next, we derive the maximum log likelihood formulation for finding optimal parameter $\mathbf{W}$ for the given training set $(\boldsymbol{\phi}_i, y_i)_{i=1,\ldots,N}$. For each $1 \le i \le N$ and $1 \le j \le \kappa$, denote $p_{ij} := h(\langle \boldsymbol{\phi}_i, \mathbf{w}_j\rangle) / \sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}_i, \mathbf{w}_c\rangle)$, the predictive probability of the y_i given $\boldsymbol{\phi}_i$ being j. We introduce the following matrix notations

$$\mathbf{Y} := \begin{bmatrix} \mathbf{1}(y_1 = 1) & \cdots & \mathbf{1}(y_1 = \kappa) \\ \vdots & & \vdots \\ \mathbf{1}(y_N = 1) & \cdots & \mathbf{1}(y_N = \kappa) \end{bmatrix}, \quad \mathbf{P} := \begin{bmatrix} p_{11} & \cdots & p_{1\kappa} \\ \vdots & & \vdots \\ p_{N1} & \cdots & p_{N\kappa} \end{bmatrix} \tag{22}$$

$$\mathbf{\Phi} := \underset{\in \mathbb{R}^{p\times N}}{\begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{\phi}(\mathbf{x}_1) & \cdots & \boldsymbol{\phi}(\mathbf{x}_N) \\ \downarrow & & \downarrow \end{bmatrix}}, \quad \mathbf{W} := \underset{\in \mathbb{R}^{p\times\kappa}}{\begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{w}_1 & \cdots & \mathbf{w}_\kappa \\ \downarrow & & \downarrow \end{bmatrix}}. \tag{23}$$

Note that the sth row of $\mathbf{Y}$ is a one-hot encoding of the lable y_s and the corresponding row of $\mathbf{Q}$ is its predictive probability distribution. Then the joint likelihood function of observing labels $(y_1, \ldots, y_N)$ given input data $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ under the above probabilistic model is

$$L(y_1, \ldots, y_N\,;\, \mathbf{W}) = \mathbb{P}(Y_1 = y_1, \ldots, Y_N = y_N\,;\, \mathbf{W}) = \prod_{s=1}^{N} \prod_{j=1}^{\kappa} (p_{sj})^{\mathbf{1}(y_s = j)}. \tag{24}$$

We can derive the negative log likelihood function $\ell(\mathbf{\Phi}, \mathbf{W}) := -\sum_{s=1}^{N} \sum_{j=1}^{\kappa} \mathbf{1}(y_s = j) \log p_{sj}$ in a matrix form as follows:

$$\ell(\mathbf{\Phi}, \mathbf{W}) = \sum_{s=1}^{N} \log \left(\sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}(\mathbf{x}_s), \mathbf{w}_c\rangle) \right) - \sum_{s=1}^{N} \sum_{j=1}^{\kappa} \mathbf{1}(y_s = j) \log h\left(\langle \boldsymbol{\phi}(\mathbf{x}_s), \mathbf{w}_j\rangle\right) \tag{25}$$

$$= \left(\sum_{s=1}^{N} \log \left(\sum_{q=1}^{\kappa} h(\langle \boldsymbol{\phi}(\mathbf{x}_s), \mathbf{w}_q\rangle) \right) \right) - \operatorname{tr}\left(\mathbf{Y}^T h(\mathbf{\Phi}^T \mathbf{W}) \right). \tag{26}$$

Then the maximum likelihood estimate $\hat{\mathbf{W}}$ is defined as the minimizer of the above loss function in $\mathbf{W}$ while fixing the feature matrix $\mathbf{\Phi}$.

Both the maps $\mathbf{W} \mapsto \ell(\mathbf{\Phi}, \mathbf{W})$ and $\mathbf{\Phi} \mapsto \ell(\mathbf{\Phi}, \mathbf{W})$ are convex and we can compute their gradients as well as the Hessian explicitly as follows. For each $y \in \{0, 1, \ldots \kappa\}$, $\boldsymbol{\phi} \in \mathbb{R}^p$, and $\mathbf{W} \in \mathbb{R}^{p\times\kappa}$, define vector and matrix functions

$$\dot{\mathbf{h}}(y, \boldsymbol{\phi}, \mathbf{W}) := (\dot{h}_1, \ldots, \dot{h}_\kappa)^T \in \mathbb{R}^{\kappa\times 1}, \; \dot{h}_j := \frac{h'(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)}{1 + \sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}, \mathbf{w}_c\rangle)} - \mathbf{1}(y = j) \frac{h'(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)}{h(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)} \tag{27}$$

$$\ddot{\mathbf{H}}(y, \boldsymbol{\phi}, \mathbf{W}) := \left(\ddot{\mathbf{H}}_{ij} \right)_{i,j} \in \mathbb{R}^{\kappa\times\kappa}, \tag{28}$$

$$\ddot{\mathbf{H}}_{ij} = \frac{h''(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)\mathbf{1}(i=j)}{1+\sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}, \mathbf{w}_c\rangle)} - \frac{h'(\langle \boldsymbol{\phi}, \mathbf{w}_i\rangle) h'(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)}{\left(1+\sum_{c=1}^{\kappa} h(\langle \boldsymbol{\phi}, \mathbf{w}_c\rangle)\right)^2} - \mathbf{1}(y = i = j) \left(\frac{h''(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)}{h(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)} - \frac{\left(h'(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle)\right)^2}{(h(\langle \boldsymbol{\phi}, \mathbf{w}_j\rangle))^2} \right). \tag{29}$$

For each $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_\kappa] \in \mathbb{R}^{p\times\kappa}$, let $\mathbf{W}^{\text{vec}} := [\mathbf{w}_1^T, \ldots, \mathbf{w}_\kappa^T]^T \in \mathbb{R}^{p\kappa}$ denote its vectorization. Then a straightforward computation shows

$$\nabla_{\text{vec}(\mathbf{W})}\ell(\mathbf{\Phi}, \mathbf{W}) = \sum_{s=1}^{N} \dot{\mathbf{h}}(y_s, \boldsymbol{\phi}_i, \mathbf{W}) \otimes \boldsymbol{\phi}_s, \tag{30}$$

$$\mathbf{H} := \nabla_{\text{vec}(\mathbf{W})}\nabla_{\text{vec}(\mathbf{W})^T}\ell(\mathbf{\Phi}, \mathbf{W}) = \sum_{s=1}^{N} \ddot{\mathbf{H}}(y_s, \boldsymbol{\phi}_s, \mathbf{W}) \otimes \boldsymbol{\phi}_s\boldsymbol{\phi}_s^T, \tag{31}$$

where $\otimes$ above denotes the Kronecker product. Recall that the eigenvalues of $\mathbf{A} \times \mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are two square matrices, are given by $\lambda_i\mu_j$, where λ_i and μ_j run over all eigenvalues of $\mathbf{A}$ and $\mathbf{B}$, respectively. Hence we can deduce

$$\lambda_{\min}\left(\mathbf{\Phi}\mathbf{\Phi}^T\right) \min_{1\le s\le N,\,\mathbf{W}} \lambda_{\min}\left(\ddot{\mathbf{H}}(y_s, \boldsymbol{\phi}_s, \mathbf{W})\right) \le \lambda_{\min}(\mathbf{H}) \tag{32}$$

$$\le \lambda_{\max}(\mathbf{H}) \le \lambda_{\max}\left(\mathbf{\Phi}\mathbf{\Phi}^T\right) \max_{1\le s\le N,\,\mathbf{W}} \lambda_{\min}\left(\ddot{\mathbf{H}}(y_s, \boldsymbol{\phi}_s, \mathbf{W})\right). \tag{33}$$

There are some particular cases worth noting. First, suppose binary classification case, $\kappa = 1$. Then the Hessian $\mathbf{H}$ above reduces to

$$\mathbf{H} = \sum_{s=1}^{N} \ddot{\mathbf{H}}_{11}(y_s, \boldsymbol{\phi}_s, \mathbf{W})\boldsymbol{\phi}_s\boldsymbol{\phi}_s^T. \tag{34}$$

Second, let $h(x) = \exp(x)$ and consider the multinomial logistic regression case. Then $h = h' = h''$ so the above yields the following concise matrix expression

$$\nabla_{\mathbf{W}}\,\ell(\mathbf{\Phi}, \mathbf{W}) = \mathbf{\Phi}(\mathbf{P} - \mathbf{Y}) \in \mathbb{R}^{p\times\kappa}, \qquad \nabla_{\mathbf{\Phi}}\,\ell(\mathbf{\Phi}, \mathbf{W}) = \mathbf{W}(\mathbf{P} - \mathbf{Y})^T \in \mathbb{R}^{p\times N}, \tag{35}$$

$$\mathbf{H} = \sum_{s=1}^{N} \begin{bmatrix} p_{s1}(1-p_{s1}) & -p_{s1}p_{s2} & \ldots & -p_{s1}p_{s\kappa} \\ -p_{s2}p_{s1} & p_{s2}(1-p_{s2}) & \ldots & -p_{s2}p_{s\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ -p_{s\kappa}p_{s1} & -p_{s\kappa}p_{s2} & \ldots & p_{s\kappa}(1-p_{s\kappa}) \end{bmatrix} \otimes \boldsymbol{\phi}_s\boldsymbol{\phi}_s^T. \tag{36}$$

It follows that eigenvalues of $\mathbf{H}$ are bounded above by $1/4$. The lower bound on the eigenvalues depend on the range of linear activation $\langle\boldsymbol{\phi}_i, \mathbf{w}_j\rangle$ may take. For instance, if we restrict the norms of the input feature vector ϕ_i and parameter $\mathbf{w}_j$, then we can find a suitable positive uniform lower bound on the eigenvalues of $\mathbf{H}$.

Lemma D.1. *Supose* $h(\cdot) = \exp(\cdot)$. *Then*

$$\lambda_{\min}\left(\ddot{\mathbf{H}}(\boldsymbol{\phi}_s, \mathbf{W})\right) \ge \min_{1\le i\le\kappa} \frac{\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_i\rangle)}{1 + \sum_{c=1}^{\kappa}\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_c\rangle)}, \tag{37}$$

$$\lambda_{\max}\left(\ddot{\mathbf{H}}(\boldsymbol{\phi}_s, \mathbf{W})\right) \le \max_{1\le i\le\kappa} \frac{\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_i\rangle)}{\left(1 + \sum_{c=1}^{\kappa}\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_c\rangle)\right)^2} \left(1 + 2\sum_{c=2}^{\kappa}\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_c\rangle)\right). \tag{38}$$

Proof. For the lower bound on the minimum eigenvalue, we note that

$$\lambda_{\min}\left(\ddot{\mathbf{H}}(\boldsymbol{\phi}_s, \mathbf{W})\right) \ge \min_{1\le i\le\kappa} \sum_{j=1}^{\kappa} \ddot{H}_{ij} = \min_{1\le i\le\kappa} p_{si}p_{s0} = \min_{1\le i\le\kappa} \frac{\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_i\rangle)}{1 + \sum_{c=1}^{\kappa}\exp(\langle\boldsymbol{\phi}_s, \mathbf{w}_c\rangle)} \tag{39}$$

where the first inequality was shown in (Amani & Thrampoulidis, 2021) using the fact that $\ddot{\mathbf{H}}(\boldsymbol{\phi}_s, \mathbf{W})$ is a diagonally dominant M-matrix (see (Tian & Huang, 2010)). The following equalities can be verified easily.

For the upper bound on the maximum eigenvalue, we use the Gershgorin circle theorem (see, e.g., (Horn & Johnson, 2012)) to bound

$$\lambda_{\max}\left(\ddot{\mathbf{H}}(\boldsymbol{\phi}_s, \mathbf{W})\right) \le \max_{1\le i\le\kappa} \left(p_{si}(1-p_{si}) + \sum_{c=2}^{\kappa} p_{si}p_{sc}\right) \le \max_{1\le i\le\kappa} p_{si}\left(2 - p_{s0} - 2p_{si}\right). \tag{40}$$

Then simplifying the last expression gives the assertion. □

E. Exponential convergence of Low-rank PGD

In Section B, we sketched our key idea of solving the SDL problem (2), which was to 'lift' the nonconvex problem *two steps* to a low-rank matrix estimation problem. In this section, we make this approach precise by considering abstract forms of optimization problems that specializes to the SDL problem (2).

Fix a function $f : \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_3 \times d_4} \to \mathbb{R}$, which takes the input of a $d_1 \times d_2$ matrix and an augmented variable in $\mathbb{R}^{d_3 \times d_4}$. Consider the following *constrained and augmented low-rank estimation* (CALE) problem

$$\min_{\mathbf{Z}=[\mathbf{X},\mathbf{\Gamma}] \in \subseteq \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_3 \times d_4}} f(\mathbf{Z}), \qquad \text{subject to } \mathbf{Z} \in \mathbf{\Theta} \text{ and } \operatorname{rank}(\mathbf{X}) \leq r, \tag{41}$$

where $\mathbf{\Theta}$ is a convex subset of $\mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_3 \times d_4}$. Here, we seek to find a global minimizer $\mathbf{Z}^\star = [\mathbf{X}^\star, \mathbf{\Gamma}^\star]$ of the objective function f over the convex set $\mathbf{\Theta}$, consisting of a low-rank matrix component $\mathbf{X}^\star \in \mathbb{R}^{d_1 \times d_2}$ and an auxiliary variable $\mathbf{\Gamma}^\star \in \mathbb{R}^{d_3 \times d_4}$. In a statistical inference setting, the loss function $f = f_n$ may be based on n noisy observations according to a probabilistic model, and the true parameter $\mathbf{Z}^*$ to be estimated may approximately minimize f over the constraint set $\mathbf{\Theta}$, with some statistical error $\varepsilon(n)$ depending on the sample size n. In this case, a global minimizer $\mathbf{Z}^\star \in \arg\min_{\mathbf{\Theta}} f$ serves as an estimate of the true parameter $\mathbf{Z}^*$. The matrix completion and low-rank matrix estimation problem (Meka et al., 2009; Recht et al., 2010) can be considered as special cases of (41) without constraint $\mathbf{\Theta}$ and the auxiliary variable $\mathbf{\Gamma}$. This problem setting has been one of the most important research topics in the machine learning and statistics literature for the past few decades. More importantly for our purpose, we have seen in (9) and (10) in the main manuscript that both the feature- and filter-based SDL problems can be cast as the form of (41) after some lifting and change of variables.

One can reformulate (41) as the following nonconvex problem, where one parameterizes the low-rank matrix variable $\mathbf{X}$ with product $\mathbf{U}\mathbf{V}^T$ of two matrices, which we call the *constrained and augmented factored estimation* (CAFE) problem:

$$\min_{\mathbf{U} \in \mathbb{R}^{d_1 \times r}, \mathbf{V} \in \mathbb{R}^{d_2 \times r}, \mathbf{\Gamma} \in \mathbb{R}^{d_3 \times d_4}} f(\mathbf{U}\mathbf{V}^T, \mathbf{\Gamma}), \qquad \text{subject to } [\mathbf{U}\mathbf{V}^T, \mathbf{\Gamma}] \in \mathbf{\Theta}. \tag{42}$$

Note that a solution to (42) gives a solution to (41). Conversely, for (41) without constraint on the first matrix component, singular value decomposition of the first matrix component easily shows that a solution to (41) is also a solution to (42). Recently, there has been a surge of progress in global guarantees of solving the factored problem (42) using various nonconvex optimization methods (Jain et al., 2010; 2013; Zhao et al., 2015b; Zheng & Lafferty, 2015; Tu et al., 2016; Park et al., 2017; Wang et al., 2017; Park et al., 2016; 2018). Most of the work considers (42) without the auxiliary variable and constraints, some with a particular type of constraints (e.g., matrix norm bound), but not general convex constraints.

It is common that the nonconvex factored problem (42) is introduced as a more efficient formulation of the convex problem (41). Interestingly, in the present work, we will reformulate the four-factor nonconvex problem of SDL in (2) as a three-factor nonconvex CAFE problem in (42) and then realize it as a single-factor convex CALE problem in (41). We illustrated this connection briefly in Section B.1.

In order to solve the CALE problem (41), consider the following *Low-rank Projected Gradient Descent* (LPGD) algorithm:

$$\mathbf{Z}_t \leftarrow \Pi_r \left(\Pi_{\mathbf{\Theta}} \left(\mathbf{Z}_{t-1} - \tau \nabla f(\mathbf{Z}_{t-1}) \right) \right), \tag{43}$$

where τ is a stepsize parameter, $\Pi_{\mathbf{\Theta}}$ denotes projection onto the convex constraint set $\mathbf{\Theta} \subseteq \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_3 \times d_4}$, and Π_r denotes the projection of the first matrix component onto matrices of rank at most r in $\mathbb{R}^{d_1 \times d_2}$. More precisely, let $\mathbf{Z} = [\mathbf{X}, \mathbf{\Gamma}]$. Then $\Pi_r(\mathbf{Z}) := [\Pi_r(\mathbf{X}), \mathbf{\Gamma}]$. It is well-known that the rank-r projection above can be explicitly computed by the singular value decomposition (SVD). Namely, $\Pi_r(\mathbf{X}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma}$ is the $r \times r$ diagonal matrix of the top r singular values of $\mathbf{X}$ and $\mathbf{U} \in \mathbb{R}^{d_1 \times r}$, $\mathbf{V} \in \mathbb{R}^{d_2 \times r}$ are semi-orthonormal matrices (i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_r$). Note that algorithm (43) resembles the standard projected gradient descent (PGD) in the optimization literature, as a gradient descent step is followed first by projecting onto the convex constraint set $\mathbf{\Theta}$ and then by the rank-r projection. It is also worth noting the similarity of (43) to the 'lift-and-project' algorithm in (Chu et al., 2003) for structured low-rank approximation problem, which proceeds by alternatively applying the projections $\Pi_{\mathbf{\Theta}}$ and Π_r to a given matrix until convergence.

In Theorem E.2, we show that the iterate $\mathbf{Z}_t$ of algorithm (43) converges exponentially to a low-rank approximation of the global minimizer of the objective f over $\mathbf{\Theta}$, given that the objective f satisfies the following restricted strong convexity (RSC) and restricted smoothness (RSM) properties in Definition E.1. These properties were first used in (Agarwal et al., 2010; Ravikumar et al., 2011; Negahban & Wainwright, 2011) for a class of matrix estimation problems and have found a number of applications in optimization and machine learning literature (Wang et al., 2017; Park et al., 2018; Tong et al., 2021).

Definition E.1. (Restricted Strong Convexity and Smoothness) A function $f : \mathbb{R}^{d_1\times d_2} \times \mathbb{R}^{d_3\times d_4} \to \mathbb{R}$ is *r-restricted strongly convex and smooth* with parameters $\mu, L > 0$ if for all $\mathbf{Z}, \mathbf{Z}' \in \mathbb{R}^{d_1\times d_2} \times \mathbb{R}^{d_3\times d_4}$ whose matrix coordinates are of rank $\leq r$,

$$\frac{\mu}{2}\|\mathrm{vec}(\mathbf{Z}) - \mathrm{vec}(\mathbf{Z}')\|_2^2 \overset{\text{(RSC)}}{\leq} f(\mathbf{Z}') - f(\mathbf{Z}) - \langle \nabla f(\mathbf{Z}), \mathbf{Z}' - \mathbf{Z}\rangle \overset{\text{(RSM)}}{\leq} \frac{L}{2}\|\mathrm{vec}(\mathbf{Z}) - \mathrm{vec}(\mathbf{Z}')\|_2^2. \tag{44}$$

Recall that the CALE problem (41) is a constrained optimization problem, where the global minimizer of the objective function f over the constraint set $\mathbf{\Theta}$ need not be a critical point of f, but only a stationary point when it is at the boundary of $\mathbf{\Theta}$. In order to measure the rate of convergence of an algorithm to a stationary point, we use gradient mapping (Nesterov, 2013; Beck, 2017) as a measure of the degree at which a point $\mathbf{Z}^\star$ in $\mathbf{\Theta}$ fails to be a stationary point, which is particularly well-suited for projected gradient descent type algorithms. Namely, for the CALE problem in (41), we define a map $G : \mathbf{\Theta} \times (0, \infty) \to \mathbb{R}$ by

$$G(\mathbf{Z}, \tau) := \frac{1}{\tau}(\mathbf{Z} - \Pi_{\mathbf{\Theta}}(\mathbf{Z} - \tau\nabla f(\mathbf{Z}))). \tag{45}$$

We call G the *gradient mapping* associated with problem (41). In order to motivate the definition, fix $\mathbf{Z} \in \mathbf{\Theta}$ and decompose it as

$$\mathbf{Z} = \Pi_{\mathbf{\Theta}}(\mathbf{Z} - \tau\nabla f(\mathbf{Z})) + (\mathbf{Z} - \Pi_{\mathbf{\Theta}}(\mathbf{Z} - \tau\nabla f(\mathbf{Z}))) \tag{46}$$

$$= \Pi_{\mathbf{\Theta}}(\mathbf{Z} - \tau\nabla f(\mathbf{Z})) + \tau G(\mathbf{Z}, \tau). \tag{47}$$

Namely, the first term above is a one-step update of a projected gradient descent at $\mathbf{Z}$ over $\mathbf{\Theta}$ with stepsize τ, and the second term above is the error term. If $\mathbf{Z}$ is a stationary point of f over $\mathbf{\Theta}$, then $-\nabla f(\mathbf{Z})$ lies in the normal cone of $\mathbf{\Theta}$ at $\mathbf{Z}$, so $\mathbf{Z}$ is invariant under the projected gradient descent and the error term above is zero. If $\mathbf{Z}$ is only approximately stationary, then the error above is nonzero. In fact, $G(\mathbf{Z}, \tau) = 0$ if and only if $\mathbf{Z}$ is a stationary point of f over $\mathbf{\Theta}$ (see Theorem 10.7 in (Beck, 2017)). Therefore, we may use the size of $G(\mathbf{Z}, \tau)$ (measured using an appropriate norm) as a measure of first-order optimality of $\mathbf{Z}$ for the problem (41). In the special cases when $\mathbf{\Theta}$ is the whole space or when $\mathbf{Z}$ is in the interior of $\mathbf{\Theta}$, if τ is sufficiently small (so that $\mathbf{Z} - \tau\nabla f(\mathbf{Z}) \in \mathbf{\Theta}$), then $\|G(\mathbf{Z}, \tau)\|_F = \|\nabla f(\mathbf{Z})\|_F$, which is the standard measure of first-order optimality of $\mathbf{Z}$ for minimizing the objective f. In general, it holds that $\|G(\mathbf{Z}, \tau)\|_F \leq \|\nabla f(\mathbf{Z})\|_F$ (see Lemma H.1).

Now we state our result concerning exponential convergence of the LPGD algorithm (43) for CALE (41).

Theorem E.2. *(Exponential convergence of LPGD) Let $f : \mathbb{R}^{d_1\times d_2} \times \mathbb{R}^{d_3\times d_4} \to \mathbb{R}$ be r-restricted strongly convex and smooth with parameters μ and L, respectively, with $L/\mu < 3$. Let $(\mathbf{Z}_t)_{t\geq 0}$ be the iterates generated by algorithm (43). Suppose $\mathbf{\Theta} \subseteq \mathbb{R}^{d_1\times d_2} \times \mathbb{R}^{d_3\times d_4}$ is a convex subset and fix a stepsize $\tau \in (\frac{1}{2\mu}, \frac{3}{2L})$. Then $\rho := 2\max(|1-\tau\mu|, |1-\tau L|) \in (0, 1)$ and the followings hold:*

(i) *(Correctly specified case) Suppose $\mathbf{Z}^\star = [\mathbf{X}^\star, \mathbf{\Gamma}^\star]$ is a stationary point of f over $\mathbf{\Theta}$ such that $\mathrm{rank}(\mathbf{X}^\star) \leq r$. Then $\mathbf{Z}^\star$ is the unique global minimizer of (41), $\lim_{t\to\infty} \mathbf{Z}_t = \mathbf{Z}^\star$, and for $t \geq 1$,*

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \leq \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F. \tag{48}$$

(ii) *(Possibly misspecified case) Let $\mathbf{Z}^\star = [\mathbf{X}^\star, \mathbf{\Gamma}^\star]$ be an arbitrary point in the interior of $\mathbf{\Theta}$ with $\mathrm{rank}(\mathbf{X}^\star) \leq r$. Then for $t \geq 1$,*

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \leq \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{\tau}{1-\rho}\left(\sqrt{r}\|\nabla_{\mathbf{X}} f(\mathbf{Z}^\star)\|_2 + \|\nabla_{\mathbf{\Gamma}} f(\mathbf{Z}^\star)\|_F\right). \tag{49}$$

In general, if $\mathbf{Z}^\star$ is an arbitrary point of $\mathbf{\Theta}$, then denoting the gradient mapping $[\Delta\mathbf{X}^\star, \Delta\mathbf{\Gamma}^\star] := \frac{1}{\tau}(\mathbf{Z}^\star - \Pi_{\mathbf{\Theta}}(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star)))$ at $\mathbf{Z}^\star$, then for $t \geq 1$,

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \leq \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{\tau}{1-\rho}\left(\sqrt{r}\|\Delta\mathbf{X}^\star\|_2 + \|\Delta\mathbf{\Gamma}^\star\|_F\right). \tag{50}$$

Theorem E.2 **(i)** assers that the LPGD algorithm (43) converges at a linear rate to the unique global minimizer $\mathbf{Z}^\star$, provided that there exists a stationry point $\mathbf{Z}^\star$ of f over the convex constraint set $\mathbf{\Theta}$ with the first matrix factor $\mathbf{X}$ having rank at most

r. In a statistical estimation setting where one seeks to estimate a 'ground-truth' parameter $\mathbf{Z}^\star$ with low-rank matrix factor from noisy observations. In this case the objective f represents the empirical error. Hence in this case, is reasonable to assume that the gradient $\nabla f(\mathbf{Z}^\star)$ is small or at least $\mathbf{Z}^\star$ is near-stationary. In fact, Wang et al. (Wang et al., 2017) makes such an assumption.

In contrast, Theorem E.2 does not require such an assumption of near-optimality of the parameter $\mathbf{Z}^\star$ to be estimated. In practical situations, the rank of the ground-truth parameter is often unknown, and one attempts to explain observed data by using a low-rank model, in which case the assumed rank r could be much lower than the true rank. For such generic situations, Theorem E.2 **(ii)** shows that the LPGD algorithm (43) converges linearly to a low-rank parameter that comes closest to being first-order optimal for f within the convex constraint $\mathbf{\Theta}$. This general result will also be used in the proof of Theorems 2.1 and C.5, the computational and the statistical estimation guarantee of SDL.

In order to establish Theorem E.2, which shows exponential convergence of the low-rank projected gradient descent (algorithm (43)) for the CALE problem 41. The proof is similar to the standard argument that shows exponential convergence projected gradient descent with fixed step size for constrained strongly convex problems (see, e.g., Theorem 10.29 in (Beck, 2017)). However, when we minimize a strongly convex objective with a rank-constrained matrix parameter, the constraint set of low-rank matrices is not convex, so one cannot use non-expansiveness of convex projection operator. Indeed, the rank-r projection Π_r by truncated SVD is not guaranteed to be non-expansive.

In order to circumvent the above issue, we use the idea of comparing the iterates $\mathbf{Z}_t$ from (43) with an auxiliary iterates $\hat{\mathbf{Z}}_t$, which is obtained by using a suitable linear projection in place of the rank-r projection. This will allow us to show that the rank-r projection is essentially 2-Lipschitz. So if the contraction constant in standard analysis of projected gradient descent for strongly convex objectives is small enough ($< 1/2$), then overall one still retains exponential convergence. (See Lemma E.3.) We emphasize that our analysis sketched above applies to the original LPGD algorithm (43): *We do NOT analyze an easier algorithm that replaces the low-rank projection with a linear projection.*

Lemma E.3. *(Linear projection factoring through rank-r projection) Fix $\mathbf{Y} \in \mathbb{R}^{d_1 \times d_2}$, $R \geq r \in \mathbb{N}$, and denote $\mathbf{X} = \Pi_r(\mathbf{Y})$ and $\hat{\mathbf{X}} = \Pi_{\mathcal{A}}(\mathbf{Y})$, where $\mathcal{A} \subseteq \mathbb{R}^{d_1 \times d_2}$ is a linear subspace. Let $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ denote the SVD of $\mathbf{X}$. Suppose there exists $\overline{\mathbf{U}} \in \mathbb{R}^{d_1 \times R}$ and $\overline{\mathbf{V}} \in \mathbb{R}^{d_2 \times R}$ such that*

$$\mathcal{A} = \left\{ \mathbf{A} \in \mathbb{R}^{d_1 \times d_2} \,\middle|\, \mathrm{col}(\mathbf{A}^T) \subseteq \mathrm{col}(\overline{\mathbf{V}}),\ \mathrm{col}(\mathbf{A}) \subseteq \mathrm{col}(\overline{\mathbf{U}}) \right\}, \tag{51}$$

$$\mathrm{col}(\mathbf{U}) \subseteq \mathrm{col}(\overline{\mathbf{U}}), \quad \mathrm{col}(\mathbf{V}) \subseteq \mathrm{col}(\overline{\mathbf{V}}). \tag{52}$$

Then $\mathbf{X} = \Pi_r(\hat{\mathbf{X}})$.

Proof. Write $\mathbf{Y} - \mathbf{X} = \dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T$ for its SVD. Let $d := \mathrm{rank}(\mathbf{Y})$ and let $\sigma_1 \geq \cdots \geq \sigma_d > 0$ denote the nonzero singular values of $\mathbf{Y}$. Since $\mathbf{X} = \Pi_r(\mathbf{Y}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T + \dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T$, we must have that $\mathbf{\Sigma}$ consists of the top r singular values of $\mathbf{Y}$ and the rest of $d - r$ singular values are contained in $\dot{\mathbf{\Sigma}}$. Furthermore, $\mathrm{col}(\mathbf{U}) \perp \mathrm{col}(\dot{\mathbf{U}})$.

Now, since $\mathbf{X} \in \mathcal{A}$ and $\Pi_{\mathcal{A}}$ is linear, we get

$$\hat{\mathbf{X}} = \Pi_{\mathcal{A}}(\mathbf{X} + (\mathbf{Y} - \mathbf{X})) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T + \Pi_{\mathcal{A}}(\dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T). \tag{53}$$

Let $\mathbf{Z} := \Pi_{\mathcal{A}}(\dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T)$ and write its SVD as $\mathbf{Z} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^T$. Then note that $(\mathbf{U}^T\overline{\mathbf{U}}\,\overline{\mathbf{U}}^T)^T = \overline{\mathbf{U}}\,\overline{\mathbf{U}}^T\mathbf{U} = \mathbf{U}$ since $\overline{\mathbf{U}}\,\overline{\mathbf{U}}^T : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}$ is the orthogonal projection onto $\mathrm{col}(\overline{\mathbf{U}}) \supseteq \mathrm{col}(\mathbf{U})$. Hence $\mathbf{U}^T\overline{\mathbf{U}}\,\overline{\mathbf{U}}^T = \mathbf{U}^T$, so we get

$$\mathbf{U}^T\mathbf{Z} = \left(\mathbf{U}^T\overline{\mathbf{U}}\,\overline{\mathbf{U}}^T\right)\dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T\mathbf{V}^T\overline{\mathbf{V}} = \left(\mathbf{U}^T\dot{\mathbf{U}}\right)\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T\mathbf{V}^T\overline{\mathbf{V}} = O. \tag{54}$$

It follows that $\mathbf{U}^T\widetilde{\mathbf{U}} = O$, since $\mathbf{U}^T\widetilde{\mathbf{U}} = \mathbf{U}^T\mathbf{Z}\widetilde{\mathbf{V}}(\widetilde{\mathbf{\Sigma}})^{-1} = O$. Therefore, rewriting (53) gives the SVD of $\hat{\mathbf{X}}$ as

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{U} & \widetilde{\mathbf{U}} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & O \\ O & \widetilde{\mathbf{\Sigma}} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \widetilde{\mathbf{V}} \end{bmatrix}. \tag{55}$$

Furthermore, $\|\Pi_{\mathcal{A}}(\dot{\mathbf{U}}\dot{\mathbf{\Sigma}}\dot{\mathbf{V}}^T)\|_2 \leq \|\dot{\mathbf{\Sigma}}\|_2 = \sigma_{r+1}^t$, so Σ consists of the top r singular values of $\hat{\mathbf{X}}$. It follows that $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the best rank-r approximation of $\hat{\mathbf{X}}$, as desired. □

***Proof of Theorem** E.2.* We first derive **(i)** assuming **(ii)**. Suppose $\mathbf{Z}^\star = [\mathbf{X}^\star, \mathbf{\Gamma}^\star]$ is a stationary point of f over $\mathbf{\Theta}$ such that $\text{rank}(\mathbf{X}^\star) \leq r$. Let $\mathbf{Z} = [\mathbf{X}, \mathbf{\Gamma}]$ be arbitrary in $\mathbf{\Theta}$ with $\text{rank}(\mathbf{X}) \leq r$. By stationarity of $\mathbf{Z}^\star$ we have $\langle \nabla f(\mathbf{Z}^\star),\ \mathbf{Z} - \mathbf{Z}^\star \rangle \geq 0$, so by RSC (44),

$$\frac{\mu}{2}\|\text{vec}(\mathbf{Z}) - \text{vec}(\mathbf{Z}^\star)\|^2 \leq f(\mathbf{Z}) - f(\mathbf{Z}^\star). \tag{56}$$

Hence $f(\mathbf{Z}^\star) \geq f(\mathbf{Z})$. Thus $\mathbf{Z}^\star$ is the unique global minimizer of (41). Also, since $\mathbf{Z}^\star$ is a stationary point of f over $\mathbf{\Theta}$, the gradient mapping $\frac{1}{\tau}(\mathbf{Z}^\star - \Pi_{\mathbf{\Theta}}(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star)))$ is zero. Thus the rest of **(i)** follows from **(ii)**.

Next, we prove **(i)**. Let $\mathbf{Z}^\star = [\mathbf{X}^\star, \boldsymbol{\gamma}^\star] \in \mathbf{\Theta}$ be arbitrary with $\text{rank}(\mathbf{X}^\star) \leq r$. Fix an iteration counter $t \geq 1$. Our proof consists of several steps.

Step 1: Constructing an approximating linear subspace $\mathcal{A}$

Let $\mathbf{X}^\star = \mathbf{U}^\star\mathbf{\Sigma}^\star(\mathbf{V}^\star)^T$ denote the SVD of $\mathbf{X}^\star$. For each iteration t, denote $\mathbf{Z}_t = [\mathbf{X}_t, \boldsymbol{\gamma}_t]$ and let $\mathbf{X}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^T$ denote the SVD of $\mathbf{X}_t$. Since $\mathbf{X}_t$ and $\mathbf{X}^\star$ have rank at most r, all of both $\mathbf{U}^\star$, $\mathbf{U}_t$, $\mathbf{V}^\star$, and $\mathbf{V}_t$ have at most r columns. Define a matrix $\mathbf{U}_{3r}$ so that its columns form a basis for the subspace spanned by the columns of $[\mathbf{U}^\star, \mathbf{U}_{t-1}, \mathbf{U}_t]$. Then $\mathbf{U}_{3r}$ has at most $3r$ columns. Similarly, let $\mathbf{U}_{3r}$ be a matrix so that its columns form a basis for the subspace spanned by the columns of $[\mathbf{V}^\star, \mathbf{V}_{t-1}, \mathbf{V}_t]$. Then $\mathbf{V}_{3r}$ has at most $3r$ columns. Now, define the subspace

$$\mathcal{A} := \left\{\Delta \in \mathbb{R}^{d_1 \times d_2} \,|\, \text{span}(\Delta^T) \subseteq \text{span}(\mathbf{V}_{3r}),\ \text{span}(\Delta) \subseteq \text{span}(\mathbf{U}_{3r})\right\}. \tag{57}$$

Note that $\mathcal{A}$ is a convex subset of $\mathbb{R}^{d_1 \times d_2}$. Also note that, by definition, $\mathbf{X}^\star, \mathbf{X}_t, \mathbf{X}_{t-1} \in \mathcal{A}$. Let $\Pi_{\mathcal{A}}$ denote the projection operator onto $\mathcal{A}$. More precisely, for each $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2}$, we have

$$\Pi_{\mathcal{A}}(\mathbf{X}) = \mathbf{U}_{3r}\mathbf{U}_{3r}^T\mathbf{X}\mathbf{V}_{3r}\mathbf{V}_{3r}^T. \tag{58}$$

Step 2: Constructing auxiliary iterates $\hat{\mathbf{Z}}_t$

Let $\mathcal{A}$ denote the linear subspace of $\mathbb{R}^{d_1 \times d_2}$ in (57). Denote the projection operator

$$\Pi' := \Pi_{\mathcal{A} \times \mathbb{R}^{d_3 \times d_4}}. \tag{59}$$

Define the following auxiliary iterates

$$\hat{\mathbf{Z}}_t = [\hat{\mathbf{X}}_t, \mathbf{\Gamma}_t] := \Pi'\left(\Pi_{\mathbf{\Theta}}\left(\mathbf{Z}_{t-1} - \tau\nabla f(\mathbf{Z}_t)\right)\right). \tag{60}$$

By Lemma E.3 and the choice of $\mathcal{A}$, we have

$$\mathbf{X}_t = \Pi_r(\hat{\mathbf{X}}_t) \in \underset{\mathbf{X}, \text{rank}(\mathbf{X}) \leq r}{\arg\min} \|\hat{\mathbf{X}}_t - \mathbf{X}\|_F \quad \text{and} \quad \mathbf{Z}_t, \mathbf{Z}_{t-1}, \mathbf{Z}^\star \in \mathcal{A} \times \mathbb{R}^{d_3 \times d_4}. \tag{61}$$

It follows that

$$\begin{aligned}
\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F &\leq \|\mathbf{Z}_t - \hat{\mathbf{Z}}_t\|_F + \|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F &(62)\\
&= \|\mathbf{X}_t - \hat{\mathbf{X}}_t\|_F + \|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F &(63)\\
&\leq \|\mathbf{X}^\star - \hat{\mathbf{X}}_t\|_F + \|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F \leq 2\|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F. &(64)
\end{aligned}$$

Hence if we can show $\|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F$ is small, then $\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F$ is also small.

Step 3. Showing $\|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F$ is small

Denote the gradient mapping $\Delta\mathbf{Z}^\star := \mathbf{Z}^\star - \Pi_{\mathbf{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star)\right)$ (Recall that this equals zero if $\mathbf{Z}^\star$ were a stationary point of f over $\mathbf{\Theta}$, but we do not make such assumption in this proof). We claim that

$$\|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F \leq \eta\,\|\mathbf{Z}_{t-1} - \mathbf{Z}^\star\|_F + \|\Pi'\left(\Delta\mathbf{Z}^\star\right)\|_F, \tag{65}$$

where $\eta := \max(|1 - \tau L|,\ |1 - \tau\mu|)$.

Below we show (65). Using $\mathbf{Z}^\star \in \mathcal{A} \times \mathbb{R}^{d_3 \times d_4}$ and linearity of the linear projection Π', write

$$\begin{aligned}\mathbf{Z}^\star &= \Pi'(\mathbf{Z}^\star) && (66)\\ &= \Pi'\left(\Pi_{\boldsymbol{\Theta}}(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star))\right) + \Pi'\left(\mathbf{Z}^\star - \Pi_{\boldsymbol{\Theta}}(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star))\right) && (67)\\ &= \Pi'\left(\Pi_{\boldsymbol{\Theta}}(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star))\right) + \Pi'\left(\Delta\mathbf{Z}^\star\right). && (68)\end{aligned}$$

Using the non-expansiveness and linearity of the linear projection Π',

$$\begin{aligned}&\|\hat{\mathbf{Z}}_t - \mathbf{Z}^\star\|_F && (69)\\ &\quad= \left\| \begin{matrix} \Pi'\left(\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}_{t-1} - \tau\nabla f(\mathbf{Z}_{t-1})\right)\right) \\ -\Pi'\left(\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla f(\mathbf{Z}^\star)\right)\right) + \Pi'\left(\Delta\mathbf{Z}^\star\right) \end{matrix} \right\|_F && (70)\\ &\quad\le \|\mathbf{Z}_{t-1} - \tau\nabla f(\mathbf{Z}_{t-1}) - \mathbf{Z}^\star + \tau\nabla f(\mathbf{Z}^\star)\|_F + \|\Pi'\left(\Delta\mathbf{Z}^\star\right)\|_F && (71)\\ &\quad\le \eta\,\|\mathbf{Z}_{t-1} - \mathbf{Z}^\star\|_F + \|\Pi'\left(\Delta\mathbf{Z}^\star\right)\|_F. && (72)\end{aligned}$$

Hence in order to derive (72), it is enough to show that

$$\|\mathbf{Z} - \tau\nabla f(\mathbf{Z}) - \mathbf{Z}' + \tau\nabla f(\mathbf{Z}')\|_F \le \eta\|\mathbf{Z}_{t-1} - \mathbf{Z}^\star\|_F. \tag{73}$$

The above follows from the fact that $\mathbf{Z}_t$ and $\mathbf{Z}^\star$ have rank $\le r$ and the restricted strong convexity and smoothness properties (Definition E.1). Indeed, fix $\mathbf{Z}, \mathbf{Z}' \in \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}^{d_3 \times d_4}$ whose first matrix components have rank $\le r$. Since ∇f is continuous,

$$\begin{aligned}\mathbf{Z} - \tau\nabla f(\mathbf{Z}) - \mathbf{Z}' + \tau\nabla f(\mathbf{Z}') &= (\mathbf{Z} - \mathbf{Z}') - \tau(\nabla f(\mathbf{Z}) - \nabla f(\mathbf{Z}')) && (74)\\ &= \int_0^1 \left(\mathbf{I} - \tau\nabla^2(\mathbf{Z} + s(\mathbf{Z}' - \mathbf{Z}))\right)(\mathbf{Z} - \mathbf{Z}')\,ds. && (75)\end{aligned}$$

Using the inequality $\|\mathbf{AB}\|_F \le \|\mathbf{A}\|_2\|\mathbf{B}\|_F$

$$\|\mathbf{Z} - \tau\nabla f(\mathbf{Z}) - \mathbf{Z}' + \tau\nabla f(\mathbf{Z}')\|_F \le \sup_{\tilde{\mathbf{Z}} = [\mathbf{Z}_1, \mathbf{Z}_2]:\,\mathrm{rank}(\mathbf{Z}_1) \le r} \|\mathbf{I} - \tau\nabla^2 f(\tilde{\mathbf{Z}})\|_2\,\|\mathbf{X} - \mathbf{Y}\|_F. \tag{76}$$

Since the eigenvalues of $\nabla^2 f(\mathbf{Z})$ are contained in $[\mu, L]$, the eigenvalues of $\mathbf{I} - \tau\nabla^2 f(\mathbf{Z})$ are between $\min(1 - \tau L,\, 1 - \tau\mu)$ and $\max(1 - \tau L,\, 1 - \tau\mu)$. Hence the right hand side above is at most

$$\eta\,\|\mathbf{Z} - \mathbf{Z}'\|_F, \tag{77}$$

verifying (74). This shows (72).

Step 4: Bounding the error term

From (64) and (65), we deduce

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \le 2\eta\,\|\mathbf{Z}_{t-1} - \mathbf{Z}^\star\|_F + \|\Pi'\left(\Delta\mathbf{Z}^\star\right)\|_F. \tag{78}$$

Note that $0 \le \eta < 1/2$ if and only if $\tau \in (\frac{1}{2\mu}, \frac{3}{2L})$, and this interval is non-empty if and only if $L/\mu < 3$. Hence for such choice of τ, $0 < 2\eta < 1$, so by a recursive application of the above inequality, we obtain

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \le (2\eta)^t\,\|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{1}{1 - 2\eta}\|\Pi'\left(\Delta\mathbf{Z}^\star\right)\|_F. \tag{79}$$

Note that $\Pi'(\Delta\mathbf{X}^\star, \Delta\boldsymbol{\gamma}^\star) = [\Pi_{\mathcal{A}}(\Delta\mathbf{X}^\star), \Delta\boldsymbol{\gamma}^\star]$ and $\mathrm{rank}(\mathcal{A}) \le 3r$. Thus by triangle inequality,

$$\begin{aligned}\|\Pi'\left(\Delta\mathbf{X}^\star, \Delta\boldsymbol{\gamma}^\star\right)\|_F &\le \|\Pi'(\Delta\mathbf{X}^\star)\|_F + \|\Delta\boldsymbol{\gamma}^\star\|_F && (80)\\ &\le \sqrt{3r}\|\Delta\mathbf{X}^\star\|_2 + \|\Delta\boldsymbol{\gamma}^\star\|_F. && (81)\end{aligned}$$

This completes the proof of **(ii)**. □

Remark E.4. Note that in (80), we could have used the following crude bound

$$\|\Pi'(\Delta\mathbf{X}^\star, \Delta\boldsymbol{\gamma}^\star)\|_F \le \|[\Delta\mathbf{X}^\star, \Delta\boldsymbol{\gamma}^\star]\|_F \le \|\Delta\mathbf{X}^\star\|_F + \|\Delta\boldsymbol{\gamma}^\star\|_F \tag{82}$$

$$\le \sqrt{\text{rank}(\Delta\mathbf{X}^\star)}\|\Delta\mathbf{X}^\star\|_2 + \|\Delta\boldsymbol{\gamma}^\star\|_F, \tag{83}$$

which is also the bound we would have obtained if we choosed the trivial linear subspace $\mathcal{A} = \mathbb{R}^{d_1\times d_2}$ in the proof of Theorem E.2 above. While we know $\text{rank}(\mathbf{X}^\star) \le r$, we do not have an a priori bound on $\text{rank}(\Delta\mathbf{X}^\star)$, which could be much larger then $\sqrt{3r}$. A smarter choice of the subspace $\mathcal{A}$ as we used in the proof of Theorem E.2 ensures that we only need the factor $\sqrt{3r}$ in place of the unknown factor $\sqrt{\text{rank}(\Delta\mathbf{X}^\star)}$ as in (80).

Remark E.5. Suppose f is not only rank-restricted smooth, but also L'-smooth on $\boldsymbol{\Theta}$ for some $L' > 0$. Then we have

$$f(\mathbf{Z}_t) - f(\mathbf{Z}^\star) \le \left(\|\nabla f(\mathbf{Z}^\star)\| + L\rho^t\right)\rho^t\|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F \tag{84}$$

for $t \ge 1$. Indeed, note that

$$|f(\mathbf{Z}_n) - f(\mathbf{Z}^\star)| = \left|\int_0^1 \langle\nabla f\left(\mathbf{Z}_n + s(\mathbf{Z}^\star - \mathbf{Z}_n)\right), \mathbf{Z}_n - \mathbf{Z}^\star\rangle\, ds\right| \tag{85}$$

$$\le \int_0^1 \|\nabla f\left(\mathbf{Z}_n + s(\mathbf{Z}^\star - \mathbf{Z}_n)\right)\|\,\|\mathbf{Z}_n - \mathbf{Z}^\star\|\, ds \tag{86}$$

$$\le \int_0^1 \left(\|\nabla f(\mathbf{Z}^\star)\| + sL'\|\mathbf{Z}_n - \mathbf{Z}^\star\|\right)\|\mathbf{Z}_n - \mathbf{Z}^\star\|\, ds \tag{87}$$

$$\le \left(\|\nabla f(\mathbf{Z}^\star)\| + L'\|\mathbf{Z}_n - \mathbf{Z}^\star\|\right)\|\mathbf{Z}_n - \mathbf{Z}^\star\|. \tag{88}$$

Then (84) follows from Theorem E.2 **(ii)**.

Remark E.6. A similar approach as in our proof of Theorem E.2 was used in (Wang et al., 2017) for analyzing a similar problem without auxiliary covariates and under a stronger assumption that the gradient $\nabla f(\mathbf{Z}^\star)$ is small. Our analysis is for a more general setting but is a bit simpler and gives a weaker requirement $L/\mu < 3$ for the well-conditioning of the objective f instead of $L/\mu < 4/3$ in (Wang et al., 2017).

F. Proof of Theorems 2.1 and C.5

In this section, we prove the main results for SDL, Theorems 2.1 and C.5. In the main text, we explained that our algorithm for SDL (Alg. 1) is exactly an LPGD for the reformulated problems (9) (for SDL-$\mathbf{H}$) and 10 (for SDL-$\mathbf{H}$). Therefore, our proofs of Theorems 2.1 and C.5 are essentially verifying the well-conditioning hypothesis $L/\mu < 3$ of the general result for the LPGD algorithm (Theorem E.2).

F.1. Proof of Theorem 2.1 and its generalization

We begin with some preliminary computations. Let $\mathbf{a}_s$ denote the activation corresponding to the sth sample (see (1)). More precisely, $\mathbf{a}_s = \mathbf{A}^T\mathbf{x}_s + \boldsymbol{\gamma}^T\mathbf{x}'_s$ for the filter-based model with $\mathbf{A} \in \mathbb{R}^{p\times\kappa}$, and $\mathbf{a}_s = \mathbf{A}[:, s] + \boldsymbol{\gamma}^T\mathbf{x}'_s$ with $\mathbf{A} \in \mathbb{R}^{\kappa\times n}$. In both cases, $\mathbf{B} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{q\times\kappa}$. Then the objective function f in (2) can be written as

$$f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) := \left(-\sum_{s=1}^{n}\sum_{j=0}^{\kappa}\mathbf{1}(y_i = j)\log g_j(\mathbf{a}_s)\right) + \xi\|\mathbf{X}_{\text{data}} - \mathbf{B}\|_F^2 + \lambda\left(\|\mathbf{A}\|_F^2 + \|\boldsymbol{\gamma}\|_F^2\right) \tag{89}$$

$$= \sum_{s=1}^{n}\left(\log\left(1 + \sum_{c=1}^{\kappa}h(\mathbf{a}_s[c])\right) - \sum_{j=1}^{\kappa}\mathbf{1}(y_i = j)\log h(\mathbf{a}_s[j])\right) + \tag{90}$$

$$\xi\|\mathbf{X}_{\text{data}} - \mathbf{B}\|_F^2 + \lambda\left(\|\mathbf{A}\|_F^2 + \|\boldsymbol{\gamma}\|_F^2\right), \tag{91}$$

where $\mathbf{a}_s[i] \in \mathbb{R}$ denotes the ith component of $\mathbf{a}_s \in \mathbb{R}^\kappa$. In the proofs we provided below, we compute the Hessian of f above explicitly for the filter- and the feature-based SDL models and use Theorem E.2 to derive the result.

For each label $y \in \{0, \dots, \kappa\}$ and activation $\mathbf{a} \in \mathbb{R}^\kappa$, recall the negative log likelihood

$$\ell(y, \mathbf{a}) = \log\sum_{c=1}^{\kappa}h(a_c) - \sum_{c=1}^{\kappa}\mathbf{1}_{\{y_i=c\}}\log h(a_c) \tag{92}$$

of observing label y from the probability distribution $\mathbf{g}(\mathbf{a})$ defined in (5). An easy computation shows

$$\nabla_{\mathbf{a}}\ell(y,\mathbf{a}) = \dot{\mathbf{h}}(y,\mathbf{a}) = (\dot{h}_1,\dots,\dot{h}_\kappa) \in \mathbb{R}^\kappa, \qquad \nabla_{\mathbf{a}}\nabla_{\mathbf{a}^T}\ell(y,\mathbf{a}) = \ddot{\mathbf{H}}(y,\mathbf{a}) = (\ddot{h}_{ij}) \in \mathbb{R}^{\kappa\times\kappa}, \tag{93}$$

where

$$\dot{h}_j = \dot{h}_j(y,\mathbf{a}) := \left(\frac{h'(a_j)}{1+\sum_{c=1}^\kappa h(a_c)} - \mathbf{1}(y=j)\frac{h'(a_j)}{h(a_j)}\right) \tag{94}$$

$$\ddot{h}_{ij} := \left(\frac{h''(a_j)\mathbf{1}(i=j)}{1+\sum_{c=1}^\kappa h(a_c)} - \frac{h'(a_i)h'(a_j)}{\left(1+\sum_{c=1}^\kappa h(a_c)\right)^2}\right) - \mathbf{1}_{\{y=i=j\}}\left(\frac{h''(a_j)}{h(a_j)} - \frac{(h'(a_j))^2}{(h(a_j))^2}\right). \tag{95}$$

***Proof of Theorem** C.4 for SDL-*$\mathbf{W}$. Let $f = F$ denote the loss function for the filter-based SDL model in (2). Fix $\mathbf{Z}_1,\mathbf{Z}_2 \in \mathbf{\Theta} \subseteq \mathbb{R}^{d_1\times d_2}\times\mathbb{R}^{d_3\times d_4}$. Since the constraint set $\mathbf{\Theta}$ is convex (see Algorithm 1), $t\mathbf{Z}_1+(1-t)\mathbf{Z}_2 \in \mathbf{\Theta}$ for all $t\in[0,1]$. Then by the mean value theorem, there exists $t^*\in[0,1]$ such that for $\mathbf{Z}^* = t^*\mathbf{Z}_1+(1-t^*)\mathbf{Z}_2$,

$$f(\mathbf{Z}_2) - f(\mathbf{Z}_1) - \langle\nabla f(\mathbf{Z}_1),\, \mathbf{Z}_2-\mathbf{Z}_1\rangle \tag{96}$$

$$= (\mathrm{vec}(\mathbf{Z}_2)-\mathrm{vec}(\mathbf{Z}_1))^T\,\nabla_{\mathrm{vec}(\mathbf{Z})}\nabla_{\mathrm{vec}(\mathbf{Z})^T}f(\mathbf{Z}^*)\,(\mathrm{vec}(\mathbf{Z}_2)-\mathrm{vec}(\mathbf{Z}_1))\,. \tag{97}$$

Hence, according to Theorem E.2, it suffices to verify that for some $\mu, L>0$ such that $L/\mu<3$,

$$\frac{\mu}{2}\mathbf{I} \preceq \nabla_{\mathrm{vec}(\mathbf{Z})}\nabla_{\mathrm{vec}(\mathbf{Z})^T}f(\mathbf{Z}^*) \preceq \frac{L}{2}\mathbf{I} \tag{98}$$

for all $\mathbf{Z}^* = [\mathbf{X},\boldsymbol{\gamma}]$ with $\mathrm{rank}(\mathbf{X}^*)\le r$.

To this end, let $\mathbf{a}_s$ denote the activation corresponding to the sth sample (see (2)). More precisely, $\mathbf{a}_s = \mathbf{A}^T\mathbf{x}_s + \boldsymbol{\gamma}^T\mathbf{x}'_s$ for the filter-based model we consider here. We discussed that the objective function f in (2) can be written as (89). Denote

$$\mathbf{a}_s = \mathbf{A}^T\mathbf{x}_s+\boldsymbol{\gamma}^T\mathbf{x}'_s =: \left[\left\langle \underbrace{\begin{bmatrix}\mathbf{A}[:,j]\\ \boldsymbol{\gamma}[:,j]\end{bmatrix}}_{=:\mathbf{u}_j}, \underbrace{\begin{bmatrix}\mathbf{x}_s\\ \mathbf{x}'_s\end{bmatrix}}_{=:\boldsymbol{\phi}_s}\right\rangle;\ j=1,\dots,\kappa\right]^T \in \mathbb{R}^\kappa, \tag{99}$$

where we have introduced the notations $\mathbf{u}_j\in\mathbb{R}^{(p+q)\times 1}$ for $j=1,\dots,\kappa$ and $\boldsymbol{\phi}_s\in\mathbb{R}^{(p+q)\times 1}$ for $s=1,\dots,n$. Denote $\mathbf{U} := [\mathbf{u}_1,\dots,\mathbf{u}_\kappa] = [\mathbf{A} \parallel \boldsymbol{\gamma}] \in \mathbb{R}^{(p+q)\times\kappa}$, which is a matrix parameter that combines $\mathbf{A}$ and $\boldsymbol{\gamma}$. Also denote $\mathbf{\Phi} = (\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_n)\in\mathbb{R}^{(p+q)\times n}$ that combined feature matrix of n observations. Then we can compute the gradient and the Hessian of f above as follows:

$$\nabla_{\mathrm{vec}(\mathbf{U})}f(\mathbf{U},\mathbf{B}) = \left(\sum_{s=1}^n \dot{\mathbf{h}}(y_s,\mathbf{U}^T\boldsymbol{\phi}_s)\otimes\boldsymbol{\phi}_s\right) + 2\lambda\,\mathrm{vec}(\mathbf{U}), \quad \nabla_{\mathbf{B}}f(\mathbf{U},\mathbf{B}) = 2\xi(\mathbf{B}-\mathbf{X}_{\text{data}}) \tag{100}$$

$$\nabla_{\mathrm{vec}(\mathbf{U})}\nabla_{\mathrm{vec}(\mathbf{U})^T}f(\mathbf{U},\mathbf{B}) = \left(\sum_{s=1}^n \ddot{\mathbf{H}}(y_s,\mathbf{U}^T\boldsymbol{\phi}_s)\otimes\boldsymbol{\phi}_s\boldsymbol{\phi}_s^T\right) + 2\lambda\mathbf{I}_{(p+q)\kappa}, \tag{101}$$

$$\nabla_{\mathrm{vec}(\mathbf{B})}\nabla_{\mathrm{vec}(\mathbf{B})^T}f(\mathbf{U},\mathbf{B}) = 2\xi\mathbf{I}_{pn}, \qquad \nabla_{\mathrm{vec}(\mathbf{B})}\nabla_{\mathrm{vec}(\mathbf{U})^T}f(\mathbf{U},\mathbf{B}) = O, \tag{102}$$

where $\otimes$ above denotes the Kronecker product and the functions $\dot{\mathbf{h}}$ and $\ddot{\mathbf{H}}$ are defined in (94).

Recall that the eigenvalues of $\mathbf{A}\otimes\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are two square matrices, are given by $\lambda_i\mu_j$, where λ_i and μ_j run over all eigenvalues of $\mathbf{A}$ and $\mathbf{B}$, respectively. Hence denoting $\mathbf{H}_{\mathbf{U}} := \sum_{s=1}^N \ddot{\mathbf{H}}(y_s,\mathbf{U}^T\boldsymbol{\phi}_s,)\otimes\boldsymbol{\phi}_s\boldsymbol{\phi}_s^T$ and using C.1-C.2, we can deduce

$$\lambda_{\min}(\mathbf{H}_{\mathbf{U}}) \ge n\lambda_{\min}\left(n^{-1}\mathbf{\Phi}\mathbf{\Phi}^T\right)\min_{1\le s\le N,\,\mathbf{U}}\lambda_{\min}\left(\ddot{\mathbf{H}}(y_s,\boldsymbol{\phi}_s,\mathbf{U})\right) \ge n\delta^-\alpha^- \ge n\mu^* > 0, \tag{103}$$

$$\lambda_{\max}(\mathbf{H}_{\mathbf{U}}) \le n\lambda_{\max}\left(n^{-1}\mathbf{\Phi}\mathbf{\Phi}^T\right)\max_{1\le s\le N,\,\mathbf{U}}\lambda_{\min}\left(\ddot{\mathbf{H}}(y_s,\boldsymbol{\phi}_s,\mathbf{U})\right) \le n\delta^+\alpha^+ \le nL^*. \tag{104}$$

This holds for all $\mathbf{A},\mathbf{B},\boldsymbol{\gamma}$ such that $\mathrm{rank}([\mathbf{A},\mathbf{B}])\le r$ and under the convex constraint (also recall that $\mathbf{U}$ is the vertical stack of $\mathbf{A}$ and $\boldsymbol{\gamma}$). Hence we conclude that the objective function F in (2) verifies RSC and RSM properties (Def. E.1) with parameters $\mu=\min(2\xi,2\lambda+n\mu^*)$ and $L=\max(2\xi,2\lambda+nL^*)$. This verifies (98) for the chosen parameters μ and L. Then the rest follows from Theorem E.2. □

Next, we prove Theorem C.4 for SDL-$\mathbf{H}$, the exponential convergence of Algorithm 1 for the feature-based SDL.

Proof of Theorem C.4 for SDL-$\mathbf{H}$. We will use the same setup as in the Proof of Theorem C.4 for SDL-$\mathbf{W}$. The main part of the argument is the computation of the Hessian of loss function $f := F$ for SDL-$\mathbf{H}$ in (9), which is straightforward but substantially more involved than the corresponding computation for the filter-based case in the proof of Theorem C.4. Let $\mathbf{a}_s := \mathbf{A}[:,s] + \boldsymbol{\gamma}^T \mathbf{x}'_s$ denote the activation corresponding to the sth sample, where in this case $\mathbf{A} \in \mathbb{R}^{\kappa \times n}$ (see (2)). Denote

$$\mathbf{a}_s = \mathbf{I}_\kappa \mathbf{A}[:,s] + \boldsymbol{\gamma}^T \mathbf{x}'_s =: \left[\left\langle \underbrace{\begin{bmatrix} \mathbf{I}_\kappa[:,j] \\ \boldsymbol{\gamma}[:,j] \end{bmatrix}}_{=:\mathbf{v}_j}, \underbrace{\begin{bmatrix} \mathbf{A}[:,s] \\ \mathbf{x}'_s \end{bmatrix}}_{=:\boldsymbol{\psi}_s} \right\rangle ;\ j = 1, \ldots, \kappa \right]^T \in \mathbb{R}^\kappa. \tag{105}$$

Note that in the above representation we have concatenated $\mathbf{A}[:,s]$ with the auxiliary covariate $\mathbf{x}'_s$, whereas previously for SDL-$\mathbf{W}$ (see (99)), we concatenated $\mathbf{A}[:,j]$ with classificaiton parameter $\boldsymbol{\gamma}[:,j]$ for the auxiliary covarate for the jth class[2]. A straightforward computation shows the following gradient formulas:

$$\nabla_{\mathrm{vec}(\boldsymbol{\gamma})} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = \left(\sum_{s=1}^n \dot{\mathbf{h}}(y_s, \mathbf{a}_s) \otimes \mathbf{x}'_s \right) + 2\lambda\, \mathrm{vec}(\boldsymbol{\gamma}), \tag{106}$$

$$\nabla_{\mathrm{vec}(\mathbf{A})} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = \left(\sum_{s=1}^n \dot{\mathbf{h}}(y_s, \mathbf{a}_s) \otimes \mathbf{I}_n[:,s] \right) + 2\lambda\, \mathrm{vec}(\mathbf{A}) = \begin{bmatrix} \dot{\mathbf{h}}(y_1, \mathbf{a}_1) \\ \vdots \\ \dot{\mathbf{h}}(y_n, \mathbf{a}_n) \end{bmatrix} + 2\lambda\, \mathrm{vec}(\mathbf{A}), \tag{107}$$

$$\nabla_{\mathbf{B}} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = 2\xi(\mathbf{B} - \mathbf{X}_{\text{data}}) \tag{108}$$

$$\nabla_{\mathrm{vec}(\boldsymbol{\gamma})} \nabla_{\mathrm{vec}(\boldsymbol{\gamma})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = \left(\sum_{s=1}^n \ddot{\mathbf{H}}(y_s, \mathbf{a}_s) \otimes \mathbf{x}'_s (\mathbf{x}'_s)^T \right) + 2\lambda \mathbf{I}_{q\kappa}, \tag{109}$$

$$\nabla_{\mathrm{vec}(\mathbf{A})} \nabla_{\mathrm{vec}(\mathbf{A})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = \mathrm{diag}\left(\ddot{\mathbf{H}}(y_1, \mathbf{a}_1), \ldots, \ddot{\mathbf{H}}(y_n, \mathbf{a}_n) \right) + 2\lambda \mathbf{I}_{\kappa n} \tag{110}$$

$$\nabla_{\mathrm{vec}(\boldsymbol{\gamma})} \nabla_{\mathrm{vec}(\mathbf{A})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = \left[\ddot{\mathbf{H}}(y_1, \mathbf{a}_1) \otimes \mathbf{x}'_1, \ldots, \ddot{\mathbf{H}}(y_1, \mathbf{a}_n) \otimes \mathbf{x}'_n \right] \in \mathbb{R}^{\kappa q \times \kappa n} \tag{111}$$

$$\nabla_{\mathrm{vec}(\mathbf{B})} \nabla_{\mathrm{vec}(\mathbf{B})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = 2\xi \mathbf{I}_{pn}, \qquad \nabla_{\mathrm{vec}(\mathbf{B})} \nabla_{\mathrm{vec}(\mathbf{V})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) = O. \tag{112}$$

From this we will compute the eigenvalues of the Hessian $\mathbf{H}_{\text{feat}}$ of the loss function f. In order to illustrate our computation in a simple setting, we first assume $\kappa = 1 = q$, which corresponds to binary classification $\kappa = 1$ with one-dimensional auxiliary covariates $q = 1$. In this case, we have

$$\mathbf{H}_{\text{feat}} := \nabla_{\mathrm{vec}(\mathbf{A},\boldsymbol{\gamma},\mathbf{B})} \nabla_{\mathrm{vec}(\mathbf{A},\boldsymbol{\gamma},\mathbf{B})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) \tag{113}$$

$$= \begin{bmatrix} \ddot{h}(y_1, \mathbf{a}_1) + 2\lambda & 0 & \ldots & 0 & \ddot{h}(y_1, \mathbf{a}_1) x'_1 & O \\ 0 & \ddot{h}(y_2, \mathbf{a}_2) + 2\lambda & \ldots & 0 & \ddot{h}(y_2, \mathbf{a}_2) x'_2 & O \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & \ddot{h}(y_n, \mathbf{a}_n) + 2\lambda & \ddot{h}(y_n, \mathbf{a}_n) x'_n & O \\ \ddot{h}(y_1, \mathbf{a}_1) x'_1 & \ddot{h}(y_2, \mathbf{a}_2) x'_2 & \ldots & \ddot{h}(y_n, \mathbf{a}_n) x'_n & \left(\frac{1}{n} \sum_{s=1}^n \ddot{h}(y_s, \mathbf{a}_s) (x'_s)^2 \right) + 2\lambda & O \\ O & O & \ldots & O & O & 2\xi \mathbf{I}_{pn} \end{bmatrix}, \tag{114}$$

where we denoted $\ddot{h} = \ddot{h}_{11} \in \mathbb{R}$ and $x'_s = \mathbf{x}'_s \in \mathbb{R}$ for $s = 1, \ldots, n$. In order to compute the eigenvalues of the above matrix, we will use the following formula for determinant of 3×3 block matrix: (O representing matrices of zero entries with appropriate sizes)

$$\det \left(\begin{bmatrix} A & B & O \\ B^T & C & O \\ O & O & D \end{bmatrix} \right) = \det \left(C - B^T A^{-1} B \right) \det(A) \det(D). \tag{115}$$

[2]This is because for the feature-based model, the column $\mathbf{A}[:,s] \in \mathbb{R}^\kappa$ for $s = 1, \ldots, n$ represent a feature of the sth sample, whereas for the filter-based model, $\mathbf{A}[:,j]$ for $j = 1, \ldots, \kappa$ represents the jth filter that is applied to the feature $\mathbf{x}_s$ of the sth sample.

This yields the following simple formula for the characteristic polynomial of $\mathbf{H}_{\text{feat}}$:

$$\det(\mathbf{H}_{\text{feat}} - \lambda \mathbf{I}) \tag{116}$$

$$= \left(\sum_{s=1}^{n} \ddot{h}(y_s, \mathbf{a}_s)(x_s')^2 - \sum_{s=1}^{n} \frac{(\ddot{h}(y_s, \mathbf{a}_s))^2 (x_s')^2}{\ddot{h}(y_s, \mathbf{a}_s) + 2\lambda} + 2\lambda - \lambda \right) (2\xi - \lambda) \prod_{s=1}^{n} \left(\ddot{h}(\mathbf{y}_s, \mathbf{a}_s) + 2\lambda - \lambda \right) \tag{117}$$

$$= \left(\sum_{s=1}^{n} \frac{2\lambda \ddot{h}(y_s, \mathbf{a}_s)(x_s')^2}{\ddot{h}(y_s, \mathbf{a}_s) + 2\lambda} + 2\lambda - \lambda \right) (2\xi - \lambda)^{pn} \prod_{s=1}^{n} \left(\ddot{h}(\mathbf{y}_s, \mathbf{a}_s) + 2\lambda - \lambda \right). \tag{118}$$

By Assumption C.3, we know that $\ddot{h}(y_s, \mathbf{a}_s) > 0$ for all $s = 1, \ldots, n$, so the first term in the parenthesis in the above display is lower bounded by $2\lambda - \lambda$. It follows that

$$\lambda_{\min}(\mathbf{H}_{\text{feat}}) \geq \min(2\xi, \alpha^- + 2\lambda), \tag{119}$$

$$\lambda_{\max}(\mathbf{H}_{\text{feat}}) \leq \max \left(2\lambda + \alpha^+ \sum_{s=1}^{n} (x_s')^2,\ 2\xi,\ \alpha^+ + 2\lambda \right). \tag{120}$$

Now we generalize the above computation for general $\kappa, q \geq 1$ case. First note the general form of the Hessian as below:

$$\mathbf{H}_{\text{feat}} := \nabla_{\text{vec}(\mathbf{A},\boldsymbol{\gamma},\mathbf{B})} \nabla_{\text{vec}(\mathbf{A},\boldsymbol{\gamma},\mathbf{B})^T} f(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) \tag{121}$$

$$= \begin{bmatrix} \ddot{\mathbf{H}}(y_1, \mathbf{a}_1) + 2\lambda \mathbf{I}_\kappa & 0 & \ldots & 0 & (\ddot{\mathbf{H}}(y_1, \mathbf{a}_1) \otimes \mathbf{x}_1')^T & O \\ 0 & \ddot{\mathbf{H}}(y_2, \mathbf{a}_2) + 2\lambda \mathbf{I}_\kappa & \ldots & 0 & (\ddot{\mathbf{H}}(y_2, \mathbf{a}_2) \otimes \mathbf{x}_2')^T & O \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & \ddot{\mathbf{H}}(y_n, \mathbf{a}_n) + 2\lambda \mathbf{I}_\kappa & (\ddot{\mathbf{H}}(y_n, \mathbf{a}_n) \otimes \mathbf{x}_n')^T & O \\ \ddot{\mathbf{H}}(y_1, \mathbf{a}_1) \otimes \mathbf{x}_1' & \ddot{\mathbf{H}}(y_2, \mathbf{a}_2) \otimes \mathbf{x}_2' & \ldots & \ddot{\mathbf{H}}(y_n, \mathbf{a}_n) \otimes \mathbf{x}_n' & \sum_{s=1}^{n} \ddot{\mathbf{H}}(y_s, \mathbf{a}_s) \otimes \mathbf{x}_s'(\mathbf{x}_s)^T + 2\lambda \mathbf{I}_{q\kappa} & O \\ O & O & \ldots & O & O & 2\xi \mathbf{I}_{pn} \end{bmatrix}. \tag{122}$$

Note that for any square symmetric matrix B and a column vector $\mathbf{x}$ of matching size,

$$B \otimes \mathbf{x}\mathbf{x}^T - (B \otimes \mathbf{x})^T (B + \lambda \mathbf{I})^{-1} (B \otimes \mathbf{x}) = \left(B - B(B + \lambda \mathbf{I})^{-1} B \right) \otimes (\mathbf{x}\mathbf{x}^T) \tag{123}$$

$$= (B + \lambda I)^{-1} B \otimes \mathbf{x}\mathbf{x}^T \tag{124}$$

$$\preceq \mathbf{I} \otimes \mathbf{x}\mathbf{x}^T, \tag{125}$$

where the last diagonal dominace is due to the Woodbury identity for matrix inverse (e.g., see (Horn & Johnson, 2012)). Hence by a similar computation as before, we obtain

$$\det(n\mathbf{H}_{\text{feat}} - \lambda \mathbf{I}) \tag{126}$$

$$= \det \left(\sum_{s=1}^{n} 2\lambda \left(\ddot{\mathbf{H}}(y_s, \mathbf{a}_s) + 2\lambda \mathbf{I}_\kappa \right)^{-1} \ddot{\mathbf{H}}(y_s, \mathbf{a}_s) \otimes \mathbf{x}_s'(\mathbf{x}_s')^T + (2\lambda - \lambda) \mathbf{I}_{q\kappa} \right) (2\xi n - \lambda)^{pn} \tag{127}$$

$$\times \prod_{s=1}^{n} \det \left(\ddot{\mathbf{H}}(\mathbf{y}_s, \mathbf{a}_s) + (2\lambda - \lambda) \mathbf{I}_\kappa \right). \tag{128}$$

It follows that

$$\lambda_{\min}(\mathbf{H}_{\text{feat}}) \geq \min(2\xi, \alpha^- + 2\lambda), \tag{129}$$

$$\lambda_{\max}(\mathbf{H}_{\text{feat}}) \leq \max \left(2\lambda + \alpha^+ n \lambda_{\max} \left(n^{-1} \mathbf{X}_{\text{aux}} \mathbf{X}_{\text{aux}}^T \right),\ 2\xi,\ \alpha^+ + 2\lambda \right). \tag{130}$$

Then the rest follows from Theorem E.2. □

F.2. Proof of Theorem C.5

In this section, we prove the statistica estimation guarantee for SDL in Theorem C.5. Recall the generative model for SDL in (20). Our proof is based in Theorem 2.1 we have established previously and standard matrix concentration bounds, which we provide below:

Lemma F.1 (Generalized Hoeffding's inequality for sub-gaussian variables)**.** *Let $X_1, \dots, X_n$ denote i.i.d. random vectors in $\mathbb{R}^d$ such that $\mathbb{E}[X_k[i]^2/K^2] \leq 2$ for some constant $K > 0$ for all $1 \leq k \leq n$ and $1 \leq i \leq d$. Fix a vector $\mathbf{a} = (a_1, \dots, a_n)^T \in \mathbb{R}^n$. Then for each $t > 0$,*

$$\mathbb{P}\left(\left\|\sum_{k=1}^{n} a_k X_k\right\|_1 > t\right) \leq 2d \exp\left(\frac{-t^2}{K^2 d^2 \|\mathbf{a}\|_2^2}\right) \tag{131}$$

Proof. Follows from Theorem 2.6.2 in (Vershynin, 2018) and using a union bound over d coordinates. □

Lemma F.2. *(2-norm of matrices with independent sub-gaussian entries) Let $\mathbf{A}$ be an $m \times n$ random matrix with independent subgaussian entries $\mathbf{A}_{ij}$ of mean zero. Denote K to be the maximum subgaussian norm of $\mathbf{A}_{ij}$, that is, $K > 0$ is the smallest number such that $\mathbb{E}[\exp(\mathbf{A}_{ij})^2/K^2] \leq 2$. Then for each $t > 0$,*

$$\mathbb{P}\left(\|\mathbf{A}\|_2 \geq 3K(\sqrt{m} + \sqrt{n} + t)\right) \leq 2\exp(-t^2). \tag{132}$$

Proof. See Theorem 4.4.5 in (Vershynin, 2018). □

Now we prove Theorem C.5 for SDL-$\mathbf{W}$.

Recall that the (L_2-regularized) normalized negative log-likelihood of observing triples $(y_i, \mathbf{x}_i, \mathbf{x}_i')$ for $i = 1, \dots, n$ is given as

$$\mathcal{L}_n := F(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) + \frac{1}{2(\sigma')^2}\|\mathbf{X}_{\text{aux}} - \mathbf{C}\|_F^2 + c, \tag{133}$$

where c is a constant and F is as in (9) or (10) depending on the activation type with tuning parameter $\xi = \frac{1}{2\sigma^2}$.

Proof of Theorem C.5 for SDL-$\mathbf{W}$. Let $\mathcal{L}_n$ denote the L_2-regularized negative joint negative log likelihood function in (133) without the last three terms, and define the expected loss function $\bar{\mathcal{L}}_n(\mathbf{Z}) := \mathbb{E}_{\boldsymbol{\varepsilon}_i, \boldsymbol{\varepsilon}_i', 1 \leq i \leq n}[\mathcal{L}_n(\mathbf{Z})]$. We omit the constant terms in these functions. Define the following gradient mappings of $\mathbf{Z}^\star$ with respect to the empirical f_n and the expected $\bar{f}_n$ loss functions:

$$G(\mathbf{Z}^\star, \tau) = \frac{1}{\tau}\left(\mathbf{Z}^\star - \Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\mathcal{L}_n(\mathbf{Z}^\star)\right)\right), \quad \bar{G}(\mathbf{Z}^\star, \tau) := \frac{1}{\tau}\left(\mathbf{Z}^\star - \Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\bar{\mathcal{L}}_n(\mathbf{Z}^\star)\right)\right). \tag{134}$$

It is elementary to show that the true parameter $\mathbf{Z}^\star$ is a stationary point of $\bar{\mathcal{L}} - \lambda(\|\mathbf{A}\|_F^2 + \|\boldsymbol{\gamma}\|_F^2)$ over $\boldsymbol{\Theta} \subseteq \mathbb{R}^{p \times (\kappa + n)} \times \mathbb{R}^{q \times \kappa}$. Hence we have $\bar{G}(\mathbf{Z}^\star, \tau) = 2\lambda[\mathbf{A}^\star, O, \boldsymbol{\gamma}^\star]$, so we may write

$$G(\mathbf{Z}^\star, \tau) = G(\mathbf{Z}^\star, \tau) - \bar{G}(\mathbf{Z}^\star, \tau) + 2\lambda[\mathbf{A}^\star, O, \boldsymbol{\gamma}^\star] \tag{135}$$

$$= \frac{1}{\tau}\left[\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\mathcal{L}_n(\mathbf{Z}^\star)\right) - \Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\bar{\mathcal{L}}_n(\mathbf{Z}^\star)\right)\right] + 2\lambda[\mathbf{A}^\star, O, \boldsymbol{\gamma}^\star] \tag{136}$$

First, suppose $\mathbf{Z}^\star - \tau\nabla\mathcal{L}_n(\mathbf{Z}^\star) \in \boldsymbol{\Theta}$ (In particular, this is the case whe $\boldsymbol{\Theta}$ equals the whole space). Then we can disregard the projection $\Pi_{\boldsymbol{\Theta}}$ in the above display so we get

$$G(\mathbf{Z}^\star, \tau) - 2\lambda[\mathbf{A}^\star, O, \boldsymbol{\gamma}^\star] = \nabla\mathcal{L}_n(\mathbf{Z}^\star) - \nabla\bar{\mathcal{L}}(\mathbf{Z}^\star) =: [\Delta\mathbf{X}^\star, \Delta\boldsymbol{\gamma}^\star]. \tag{137}$$

According to Theorem 2.1, it now suffices show that $G(\mathbf{Z}^\star, \tau)$ above is small with high probability. We use the notation $\mathbf{U} = [\mathbf{A}^T, \boldsymbol{\gamma}^T]^T$, $\mathbf{U}^\star = [(\mathbf{A}^\star)^T, (\boldsymbol{\gamma}^\star)^T]^T$, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n] = [\mathbf{X}_{\text{data}}^T, \mathbf{X}_{\text{aux}}^T]^T$ (see also the proof of Theorem 2.1). Denote $\mathbf{a}_s = \mathbf{U}^T\boldsymbol{\phi}_s$ and $\mathbf{a}_s^\star = (\mathbf{U}^\star)^T\boldsymbol{\phi}_s$ for $s = 1, \dots, n$ and introduce the following random quantities

$$\mathtt{Q}_1 := \sum_{s=1}^{n} \dot{\mathbf{h}}(y_s, \mathbf{a}_s^\star) \in \mathbb{R}^\kappa, \quad \mathtt{Q}_2 := \sum_{s=1}^{n} \boldsymbol{\varepsilon}_s \in \mathbb{R}^p, \quad \mathtt{Q}_3 := \sum_{s=1}^{n} \boldsymbol{\varepsilon}_s' \in \mathbb{R}^q, \quad \mathtt{Q}_4 := [\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n] \in \mathbb{R}^{p \times n}. \tag{138}$$

Recall that

$$\nabla_{\mathrm{vec}(\mathbf{U})}\mathcal{L}_n(\mathbf{U},\mathbf{B}) = \left(\sum_{s=1}^{n} \dot{\mathbf{h}}(y_s,\mathbf{a}_s)\otimes\boldsymbol{\phi}_s\right) + 2\lambda\,\mathrm{vec}(\mathbf{U}), \quad \nabla_{\mathbf{B}}\mathcal{L}_n(\mathbf{U},\mathbf{B}) = \frac{2}{2\sigma^2}(\mathbf{B}-\mathbf{X}_{\text{data}}), \tag{139}$$

$$\nabla_{\mathrm{vec}(\mathbf{U})}\bar{\mathcal{L}}_n(\mathbf{U},\mathbf{B}) = \left(\sum_{s=1}^{n} \mathbb{E}\left[\dot{\mathbf{h}}(y_s,\mathbf{a}_s)\otimes\boldsymbol{\phi}_s\right]\right) + 2\lambda\,\mathrm{vec}(\mathbf{U}), \quad \nabla_{\mathbf{B}}\bar{\mathcal{L}}_n(\mathbf{U},\mathbf{B}) = \frac{2}{2\sigma^2}(\mathbf{B}-\mathbf{B}^\star), \tag{140}$$

where $\dot{\mathbf{h}}$ is defined in (94). Note that

$$\mathbb{E}\left[\dot{\mathbf{h}}(y_s,\mathbf{a}_s)\,\middle|\,\boldsymbol{\phi}_s\right] = \left[\left(\frac{h'(\mathbf{a}[j])}{1+\sum_{c=1}^{\kappa}h(\mathbf{a}[c])} - g_j(\mathbf{a}_s^\star)\frac{h'(\mathbf{a}[j])}{h(\mathbf{a}[j])}\right)_{\mathbf{a}=\mathbf{a}_s}; j=1,\dots,\kappa\right] \tag{141}$$

$$= \left[\left(\frac{h'(\mathbf{a}[j])}{1+\sum_{c=1}^{\kappa}h(\mathbf{a}[c])} - \frac{h(\mathbf{a}_s^\star[j])}{1+\sum_{c=1}^{\kappa}h(\mathbf{a}_s^\star[c])}\frac{h'(\mathbf{a}[j])}{h(\mathbf{a}[j])}\right)_{\mathbf{a}=\mathbf{a}_s}; j=1,\dots,\kappa\right], \tag{142}$$

so the above vanishes when $\mathbf{a}_s = \mathbf{a}_s^\star$. Hence

$$\mathbb{E}\left[\dot{\mathbf{h}}(y_s,\mathbf{a}_s^\star)\otimes\boldsymbol{\phi}_s\right] = \mathbb{E}\left[\mathbb{E}\left[\dot{\mathbf{h}}(y_s,\mathbf{a}_s^\star)\otimes\boldsymbol{\phi}_s\,\middle|\,\boldsymbol{\phi}_s\right]\right] = \mathbf{0}, \tag{143}$$

Hence we can compute the following gradients

$$\nabla_{\mathrm{vec}(\mathbf{A})}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \left(\sum_{s=1}^{n}\dot{\mathbf{h}}(y_s,\mathbf{a}_s)\otimes\mathbf{x}_s\right) \tag{144}$$

$$\nabla_{\mathrm{vec}(\boldsymbol{\gamma})}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \left(\sum_{s=1}^{n}\dot{\mathbf{h}}(y_s,\mathbf{a}_s)\otimes\mathbf{x}_s'\right) \tag{145}$$

$$\nabla_{\mathbf{B}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \frac{2}{2\sigma^2)}(\mathbf{B}^\star-\mathbf{X}_{\text{data}}) = \frac{2}{2\sigma^2}[\boldsymbol{\varepsilon}_1,\dots,\boldsymbol{\varepsilon}_n] \tag{146}$$

$$\nabla_{\boldsymbol{\lambda}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \frac{2}{2\sigma^2}\sum_{s=1}^{n}\boldsymbol{\varepsilon}_s'. \tag{147}$$

It follows that (recall the definition of $\gamma_{\max}$in C.3)

$$\|\nabla_{\mathbf{A}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A}^\star,\mathbf{B}^\star,\boldsymbol{\gamma}^\star)\|_2 = \left\|\sum_{s=1}^{n}(\mathbf{B}^\star[:,s]+\boldsymbol{\varepsilon}_s)\dot{\mathbf{h}}(y_s,\mathbf{a}_s^\star)^T\right\|_2 \tag{148}$$

$$\leq \left\|\sum_{s=1}^{n}\mathbf{B}^\star[:,s]\dot{\mathbf{h}}(y_s,\mathbf{a}_s^\star)^T\right\|_2 + \left\|\sum_{s=1}^{n}\boldsymbol{\varepsilon}_s\dot{\mathbf{h}}(y_s,\mathbf{a}_s^\star)^T\right\|_2 \tag{149}$$

$$\leq \|\mathbf{B}^\star\|_\infty\|\mathtt{Q}_1\|_2 + \gamma_{\max}\|\mathtt{Q}_2\|_2\,. \tag{150}$$

Similarly, we have

$$\|\Delta\boldsymbol{\gamma}^\star\|_F = \|\nabla_{\boldsymbol{\gamma}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A}^\star,\mathbf{B}^\star,\boldsymbol{\gamma}^\star)\|_F = \|\nabla_{\mathrm{vec}(\boldsymbol{\gamma})}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A}^\star,\mathbf{B}^\star,\boldsymbol{\gamma}^\star)\|_2 \tag{151}$$

$$\leq q\|\boldsymbol{\lambda}^\star\|_\infty\|\mathtt{Q}_1\|_2 + q\gamma_{\max}\|\mathtt{Q}_3\|_2 \tag{152}$$

Using the fact that $\|[A,B]\|_2 \leq \|A\|_2 + \|B\|_2$ for two matrices A, B with the same number of rows, we have

$$\|\Delta\mathbf{X}^\star\|_2 = \left\|\nabla_{\mathbf{A}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A}^\star,\mathbf{B}^\star,\boldsymbol{\gamma}^\star)\right\|_2 + \left\|\nabla_{\boldsymbol{\gamma}}(\mathcal{L}_n-\bar{\mathcal{L}}_n)(\mathbf{A}^\star,\mathbf{B}^\star,\boldsymbol{\gamma}^\star)\right\|_2 \tag{153}$$

$$\leq \|\mathbf{B}^\star\|_\infty\|\mathtt{Q}_1\|_2 + n\gamma_{\max}\|\mathtt{Q}_2\|_2 + \frac{2}{2\sigma^2}\|\mathtt{Q}_4\|_2\,. \tag{154}$$

Thus, combining the above bounds, we obtain

$$S := \sqrt{3r}\|\Delta\mathbf{X}^\star\|_2 + \|\Delta\boldsymbol{\gamma}^\star\|_F \leq \sum_{i=1}^{4} c_i\|\mathtt{Q}_i\|_2, \tag{155}$$

where the constants $c_1, \ldots, c_4 > 0$ are given by

$$c_1 = \left(\sqrt{3r}\|\mathbf{B}^\star\|_\infty + q\|\boldsymbol{\lambda}^\star\|_\infty\right), \quad c_2 = \gamma_{\max}\left(q + \sqrt{3r}\right), \quad c_3 = q\gamma_{\max}, \quad c_4 = \frac{2\sqrt{3}r}{2\sigma^2}. \tag{156}$$

Next, we will use concentration inequalities to argue that the right hand side in (155) is small with high probability and obtain the following tail bound on S:

$$\mathbb{P}\left(S > c\sqrt{n}\log n + 3C\sigma(\sqrt{p} + \sqrt{n} + c\sqrt{\log n})\right) \leq \frac{1}{n}, \tag{157}$$

where $C > 0$ is an absolute constant and $c > 0$ can be written explicitly in terms of the constants we use in this proof. Recall that for a random variable Z, its sub-Gaussian norm, denoted as $\|Z\|_{\psi_2}$, is the smalleset number $K > 0$ such that $\mathbb{E}[\exp(Z^2/K^2)] \leq 2$. The constant $C >$ above is the sub-gaussian norm of the standard normal variable, which can be taken as $C \leq 36e/\log 2$. Using union bound with Lemmas F.1 and F.2, for each $t, t > 0$, we get

$$\begin{aligned}
&\mathbb{P}\left(S > (c_1 + c_2 + c_3 + c_4)t + 3C\sigma(\sqrt{p} + \sqrt{n} + t')\right) && (158)\\
&\quad \leq \left(\sum_{i=1}^{3}\mathbb{P}\left(\|\mathtt{Q}_i\|_2 > t\right)\right) + \mathbb{P}\left(\|n\mathtt{Q}_4\|_2 > 3C\sigma(\sqrt{p} + \sqrt{n} + t')\right) && (159)\\
&\quad \leq 2\kappa\exp\left(\frac{-t^2}{C_1^2\kappa^2 n}\right) + 2p\exp\left(\frac{-t^2}{(C\sigma)^2p^2n}\right) + 2q\exp\left(\frac{-t^2}{(C\sigma')^2q^2n}\right) + \exp(-(t')^2). && (160)
\end{aligned}$$

Indeed, for bounding $\mathbb{P}(\mathtt{Q}_1 > t)$, we used Lemma F.1 with sub-Gaussian norm $C_1 = K = \gamma_{\max}/\sqrt{\log 2}$ for the bounded random vector $\dot{\mathbf{h}}(y_s, \mathbf{a}_s)$ (see Ex. 2.5.8 in (Vershynin, 2018)); for $\mathbb{P}(\mathtt{Q}_2 > t)$ and $\mathbb{P}(\mathtt{Q}_3 > t)$, we used Lemma F.1 with $K = C\sigma$ and $K = C\sigma'$, respectively; for the last term involving $\mathtt{Q}_4$, we used Lemma F.2 with $K = C/\sigma$. Observe that in order to make the last expression in (158) small, we will chose $t = c_5\sqrt{n}\log n$ and $t' = c_5\sqrt{\log n}$, where $c_5 > 0$ is a constant to be determined. This yields

$$\mathbb{P}\left(S > c\sqrt{n}\log n + 3C\sigma(\sqrt{p} + \sqrt{n} + c\sqrt{\log n})\right) \leq n^{-c_6}, \tag{161}$$

where $c = c_5\sum_{i=1}^{4}c_i$ and $c_6 > 0$ is an explicit constant that grows in c_5. We assume $c_5 > 0$ is such that $c_6 \geq 1$. This shows (157).

To finish, we use Theorem 2.1 to deduce that with probability at least $1/n$,

$$\begin{aligned}
\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \leq \rho^t\|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{\tau}{1-\rho}\left(c\sqrt{n}\log n + 3C\sigma(\sqrt{p} + \sqrt{n} + c\sqrt{\log n})\right) && (162)\\
+ \frac{2\lambda\tau}{1-\rho}\left(\|\mathbf{A}^\star\|_2 + \|\boldsymbol{\gamma}^\star\|_F\right) && (163)
\end{aligned}$$

Note that $\tau < \frac{3}{2L}$ with $L = \max(2\xi,\, 2\lambda + nL^*) \geq nL^*$, so $\tau < \frac{3}{2nL^*}$. So this yields the desired result.

Second, suppose $\mathbf{Z}^\star - \tau\nabla F(\mathbf{Z}^\star) \notin \boldsymbol{\Theta}$. Then we cannot direcly simplify the expression (135). In this case, we take the Frobenius norm and use non-expansiveness of the projection operator (onto convex set $\boldsymbol{\Theta}$):

$$\begin{aligned}
\|G(\mathbf{Z}^\star, \tau)\|_F &= \frac{1}{\tau}\left\|\left[\Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\mathcal{L}_n(\mathbf{Z}^\star)\right) - \Pi_{\boldsymbol{\Theta}}\left(\mathbf{Z}^\star - \tau\nabla\bar{\mathcal{L}}_n(\mathbf{Z}^\star)\right)\right]\right\|_F && (164)\\
&\leq \|\nabla\mathcal{L}_n(\mathbf{Z}^\star) - \nabla\bar{\mathcal{L}}_n(\mathbf{Z}^\star)\|_F && (165)\\
&\leq \|\Delta\mathbf{X}^\star\|_F + \|\Delta\boldsymbol{\gamma}^\star\|_F. && (166)
\end{aligned}$$

According to Remark E.4, we also have Theorem E.2 (and hence Theorem 2.1) with $\sqrt{3r}\|\Delta\mathbf{X}^\star\|_2$ replaced with $\|\Delta\mathbf{X}^\star\|_F$. Then an identical argument shows

$$S' := \|\Delta\mathbf{X}^\star\|_F + \|\Delta\boldsymbol{\gamma}\|_F \leq c_1\|\mathtt{Q}_1\|_2 + c_2\|\mathtt{Q}_2\|_2 + c_3\|\mathtt{Q}_3\|_2 + c_4\|\mathtt{Q}_4\|_F, \tag{167}$$

where the constants $c_1, \dots, c_4 > 0$ are the same as in (156). So we have

$$\|\mathbf{Z}_t - \mathbf{Z}^\star\|_F \leq \rho^t \|\mathbf{Z}_0 - \mathbf{Z}^\star\|_F + \frac{\tau}{1-\rho}(S' + 2\lambda(\|\mathbf{A}^\star\|_2 + \|\mathbf{A}^\star\|_F)). \tag{168}$$

Then an identical argument with the inequality $\|\mathtt{Q}_4\|_F \leq \sqrt{\min(p,n)}\|\mathtt{Q}_4\|_2$ shows

$$\mathbb{P}\left(S' > (c_1+c_2+c_3+c_4)t + 3C\sigma(\sqrt{p}+\sqrt{n}+t')\sqrt{\min(p,n)}\right) \tag{169}$$

$$\leq \left(\sum_{i=1}^{3}\mathbb{P}\left(\|\mathtt{Q}_i\|_2 > t\right)\right) + \mathbb{P}\left(\|\mathtt{Q}_4\|_2 > \frac{3C(\sqrt{p}+\sqrt{n}+t')}{\sigma}\right), \tag{170}$$

and the assertion follows similarly as before. □

It remains to show Theorem C.5 for SDL-$\mathbf{H}$.

Proof of Theorem C.5 for SDL-H. The argument is entirely similar to the proof of Theorem C.5 for SDL-$\mathbf{W}$. Indeed, denoting $\mathbf{a}_s = \mathbf{A}[:,s] + \boldsymbol{\gamma}^T\mathbf{x}'_s$ for $s = 1, \dots, n$ and keeping the other notations the same as in the proof of Theorem C.5, we can compute the following gradients

$$\nabla_{\mathbf{A}}(\mathcal{L}_n - \bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \left[\dot{\mathbf{h}}(y_1,\mathbf{a}_1), \dots, \dot{\mathbf{h}}(y_n,\mathbf{a}_n)\right] \tag{171}$$

$$\nabla_{\mathrm{vec}(\boldsymbol{\gamma})}(\mathcal{L}_n - \bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \left(\sum_{s=1}^{n}\dot{\mathbf{h}}(y_s,\mathbf{a}_s)\otimes\mathbf{x}'_s\right) \tag{172}$$

$$\nabla_{\mathbf{B}}(\mathcal{L}_n - \bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \frac{2}{2\sigma^2}(\mathbf{B}^\star - \mathbf{X}_{\text{data}}) = \frac{2}{2\sigma^2}[\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n] \tag{173}$$

$$\nabla_{\boldsymbol{\lambda}}(\mathcal{L}_n - \bar{\mathcal{L}}_n)(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}) = \frac{2}{2\sigma^2}\sum_{s=1}^{n}\boldsymbol{\varepsilon}'_s. \tag{174}$$

Hence repeating the same argument as before, using concentration inequalities for the following random quantities

$$\mathtt{Q}_1 := \left[\dot{\mathbf{h}}(y_1,\mathbf{a}_1), \dots, \dot{\mathbf{h}}(y_n,\mathbf{a}_n)\right] \in \mathbb{R}^{p\times n}, \quad \mathtt{Q}_2 := \sum_{s=1}^{n}\boldsymbol{\varepsilon}_s \in \mathbb{R}^p, \tag{175}$$

$$\mathtt{Q}_3 := \sum_{s=1}^{n}\boldsymbol{\varepsilon}'_s \in \mathbb{R}^q, \quad \mathtt{Q}_4 := [\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n] \in \mathbb{R}^{p\times n}, \tag{176}$$

one can bound the size of $G(\mathbf{Z}^\star, \tau)$ with high probability. The rest of the details are omitted. □

G. Auxiliary computations

Remark G.1. Denoting $\xi = \xi' n$ and $\lambda = \lambda' n$, the condition L/μ in Theorem 2.1 for SDL-$\mathbf{W}$ with $\lambda = 0$ reduces to

$$\frac{L^*}{\mu^*} < 3 \;\Rightarrow\; \left(\frac{L^*}{6} < \xi' < \frac{3\mu^*}{2}, \quad 0 \leq \lambda' < \frac{6\xi' - L^*}{2}\right) \cup \left(\xi' > \frac{3\mu^*}{2}, \quad \frac{2\xi' - 3\mu^*}{6} < \lambda' < \frac{6\xi' - L^*}{2}\right) \tag{177}$$

$$\frac{L^*}{\mu^*} \geq 3 \;\Rightarrow\; \left(\frac{L^* - \mu^*}{4} < \xi' < \frac{3(L^* - \mu^*)}{4}, \; \frac{L^* - 3\mu^*}{4} < \lambda' < \frac{6\xi' - L^*}{2}\right) \tag{178}$$

$$\cup \left(\xi' > \frac{3(L^* - \mu^*)}{2}, \; \frac{2\xi' - 3\mu^*}{6} < \lambda' < \frac{6\xi' - L^*}{2}\right). \tag{179}$$

H. Auxiliary lemmas

Lemma H.1. *Fix a differentiable function $f : \mathbb{R}^p \times \mathbb{R}$ and a convex set $\boldsymbol{\Theta} \subseteq \mathbb{R}^p$. Fix $\tau > 0$ and*

$$G(\mathbf{Z}, \tau) := \frac{1}{\tau}(\mathbf{Z} - \Pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta} - \tau\nabla f(\boldsymbol{\theta}))). \tag{180}$$

Then for each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\|G(\boldsymbol{\theta}, \tau)\| \leq \|\nabla f(\boldsymbol{\theta})\|$.

Proof. The assertion is clear if $\|G(\boldsymbol{\theta}, \tau)\| = 0$, so we may assume $\|G(\boldsymbol{\theta}, \tau)\| > 0$. Denote $\hat{\boldsymbol{\theta}} := \Pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta} - \tau \nabla f(\boldsymbol{\theta}))$. Note that

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}'} \|\boldsymbol{\theta} - \tau \nabla f(\boldsymbol{\theta}) - \boldsymbol{\theta}'\|^2, \tag{181}$$

so by the first-order optimality condition,

$$\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta} + \tau \nabla f(\boldsymbol{\theta}),\, \boldsymbol{\theta}' - \hat{\boldsymbol{\theta}} \rangle \geq 0 \quad \forall \boldsymbol{\theta}' \in \boldsymbol{\Theta}. \tag{182}$$

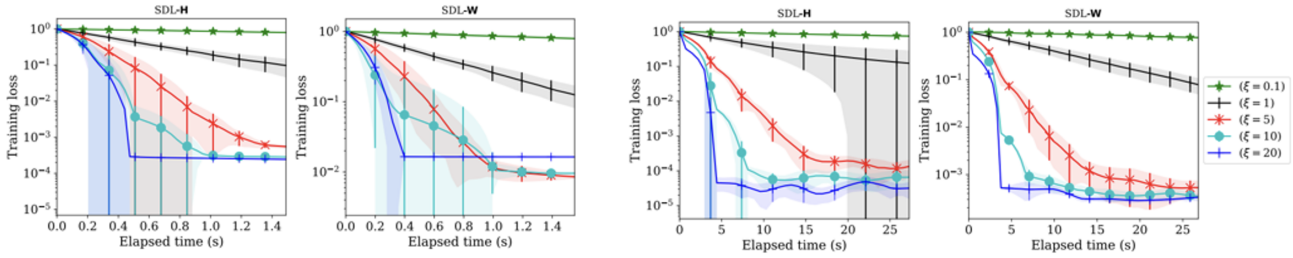
Plugging in $\boldsymbol{\theta}' = \boldsymbol{\theta}$ and using Cauchy-Schwarz inequality,

$$\tau^2 \|G(\boldsymbol{\theta}, \tau)\|^2 = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2 \leq \tau \langle \nabla f(\boldsymbol{\theta}),\, \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \rangle \leq \tau \|\nabla f(\boldsymbol{\theta})\|\, \tau \|G(\boldsymbol{\theta}, \tau)\|. \tag{183}$$

Hence the assertion follows by dividing both sides by $\tau^2 \|G(\boldsymbol{\theta}, \tau)\| > 0$. □

I. Simulation and Numerical Validation

We numerically verify Theorem 2.1 on a semi-synthetic dataset generated by using MNIST image dataset (LeCun & Cortes, 2010) and a text dataset named 'Real / Fake Job Posting Prediction' (fak). All procedures were performed on a 2021 Macbook Air with M1 chip and 16 GB of RAM. For MNIST dataset, we generate low-rank image data by taking a random linear combination of randomly selected images of digits '2' and '5' and adding some noise.



(a) MNIST data

(b) Job posting data

Figure 3. Training loss vs. elapsed CPU time for Algorithm 1 (with binary logistic classifier) on the semi-synthetic MNIST dataset ($p = 28^2 = 784$, $q = 0$, $n = 500$, $\kappa = 1$) for several values of ξ in log scale. We used L_2-regularization coefficient $\lambda = 2$ and fixed stepsize $\tau = 0.01$. Average training loss over ten runs and the shades representing the standard deviation shown.

We validate our theoretical exponential convergence results of Algorithm 1 using Figures 3. Note that the convexity and smoothness parameters μ and L in Theorem 2.1 are difficult to compute exactly. In practice, cross-validation of hyperparameters is usually employed. For $\xi \in \{0.1, 1, 5, 10, 20\}$ in Figures 3, we indeed observe exponential decay of training loss as dictated by our theoretical results for Algorithm 1. We also observe that the exponential rate of decay in training loss increases as ξ increases. According to Theorem 2.1, the contraction coefficient is $\rho = (1 - \tau\mu)$, which decreases in ξ since μ increases in ξ (see (16)). The decay for large $\xi \in \{10, 20\}$ seems even superexponential.

Here we give more details on the semi-synthetic MNIST data we used in the experiment in Figure 3. Denote $p = 28^2 = 784$, $n = 500$, $\bar{r} = 20$, $r = 2$, and $\kappa = 1$. First, we randomly select 10 images each from digits '2' and '5'. Vectorizing each image as a column in $p = 784$ dimension, we obtain a true dictionary matrix for features $\mathbf{W}_{\text{true},X} \in \mathbb{R}^{p \times \bar{r}}$. Similarly, we randomly sample 10 images of each from digits '4' and '7' and obtain the true dictionary matrix of labels $\mathbf{W}_{\text{true},\mathbf{Y}} \in \mathbb{R}^{p \times \bar{r}}$. Next, we sample a code matrix $\mathbf{H}_{\text{true}} \in \mathbb{R}^{\bar{r} \times n}$ whose entries are i.i.d. with the uniform distribution $U([0, 1])$. Then the 'pre-feature' matrix $\mathbf{X}_0 \in \mathbb{R}^{p \times n}$ of vectorized synthetic images is generated by $\mathbf{W}_{\text{true},X} \mathbf{H}_{\text{true}}$. The feature matrix $\mathbf{X}_{\text{data}} \in \mathbb{R}^{p \times n}$ is then generated by adding an independent Gaussian noise $\varepsilon_j \sim N(\mathbf{0}, \sigma^2 I_p)$ to the jth column of $\mathbf{X}_0$, for $j = 1, \ldots, n$, with $\sigma = 0.5$. We generate the binary label matrix $\mathbf{Y} = [y_1, \ldots, y_n] \in \{0, 1\}^{1 \times n}$ (recall $\kappa = 1$) as follows: Each entry y_i is an independent Bernoulli variable with probability $p_i = \left(1 + \exp\left(-\boldsymbol{\beta}_{\text{true},\mathbf{Y}}^T \mathbf{W}_{\text{true},\mathbf{Y}}^T \mathbf{X}_{\text{data}}[:, i]\right)\right)^{-1}$, where $\boldsymbol{\beta}_{\text{true},\mathbf{Y}} = [1, -1]$.