AN ONLINE ALGORITHM FOR CONTRASTIVE PRINCIPAL COMPONENT ANALYSIS

Siavash Golkar[*,†]*, David Lipshutz*[*,†]*, Tiberiu Tesileanu*[†]*, Dmitri B. Chklovskii*[†,‡]

[†]Center for Computational Neuroscience, Flatiron Institute
[‡]Neuroscience Institute, NYU Langone Medical School

ABSTRACT

Finding informative low-dimensional representations that can be computed efficiently in large datasets is an important problem in data analysis. Recently, contrastive Principal Component Analysis (cPCA) was proposed as a more informative generalization of PCA that takes advantage of contrastive learning. However, the performance of cPCA is sensitive to hyper-parameter choice and there is currently no online algorithm for implementing cPCA. Here, we introduce a modified cPCA method, which we denote cPCA*, that is more interpretable and less sensitive to the choice of hyper-parameter. We derive an online algorithm for cPCA* and show that it maps onto a neural network with local learning rules, so it can potentially be implemented in energy efficient neuromorphic hardware. We evaluate the performance of our online algorithm on real datasets and highlight the differences and similarities with the original formulation.

Index Terms— Contrastive principal component analysis, online algorithm, neural network, local learning rules

1. INTRODUCTION

Principal Component Analysis (PCA) is a long-standing pillar of dimensionality reduction that is often a first step in data analysis. As a spectral method without hyper-parameters, PCA is robust and interpretable. Furthermore, there exist efficient algorithms for computing the principal subspace projections [1], including online algorithms that map onto networks with local learning rules and can potentially be implemented in energy efficient neuromorphic hardware [2].

However, PCA projects a dataset onto the directions of highest variance, without regard to the source of this variability. In particular, if there are directions with high variability due to noise, PCA projects onto those directions, potentially at the cost of more informative subspaces. Recently, contrastive PCA (cPCA) was proposed as a generalization of PCA that can account for such scenarios [3, 4]. By using a background dataset comprised of domain-relevant 'negative samples' (e.g., a music studio recording sans music), cPCA can distinguish directions of interest from those that have high variance due to background noise. For example, in a drug trial, cPCA picks an informative subspace by discounting the directions of high variance in the control (or placebo) group. This method has proved useful for both analysis and visualization in domains where a background dataset is available [5–7]. It has also been applied to the image and video compression domain [8].

While cPCA has been shown to find more informative projections than PCA [3], this improvement has come at the cost of interpretability and efficiency. In particular, cPCA is highly sensitive to a hyper-parameter that sets the relative contribution of the positive and negative samples. This renders the algorithm less robust and also less interpretable than PCA. Furthermore, online PCA algorithms are not readily adapted to perform cPCA, so cPCA is less amenable to large datasets or in scenarios with limited amount of compute.

In this work, we introduce cPCA*, a more robust variation of cPCA. We show cPCA* is less sensitive to hyper-parameter choice and, for specific generative examples, can be interpreted as finding the subspace that maximizes the signal-to-noise ratio. We derive an online algorithm for cPCA* and map it onto a neural network with local learning rules which can be implemented on efficient neuromorphic chips. In summary,

- We introduce cPCA*, an interpretable contrastive PCA method, and derive an online algorithm that maps onto a neural network with local learning rules.
- We empirically show that cPCA* is more robust to the choice of hyper-parameter than cPCA.

2. cPCA: ORIGINAL AND OUR METHOD

Given a dataset consisting of positive and negative samples, we want to identify directions that have large variance in the positive samples but small variance in the negative samples.

To be precise, let $d \geq 2$ and $(\mathbf{x}_1, \delta_1), \ldots, (\mathbf{x}_T, \delta_T) \in \mathbb{R}^d \times \{0, 1\}$ be a sequence of inputs. The input $\mathbf{x}_t$ is a feature vector, which can either be a *positive sample* or a *negative sample*. The scalar δ_t is an indicator variable such that $\delta_t = 1$ for positive samples and $\delta_t = 0$ for negative samples. We define covariance matrices of the positive and negative samples, as follows:

$$\mathbf{C}_{(+)} := \langle \mathbf{x}_t \mathbf{x}_t^\top | \delta_t = 1 \rangle_t, \quad \mathbf{C}_{(-)} := \langle \mathbf{x}_t \mathbf{x}_t^\top | \delta_t = 0 \rangle_t.$$

*SG and DL contributed equally.

cPCA finds directions (i.e., unit vectors $\mathbf{v} \in \mathbb{R}^d$) that simultaneously maximize $\mathbf{v}^\top \mathbf{C}_{(+)}\mathbf{v}$ and minimize $\mathbf{v}^\top \mathbf{C}_{(-)}\mathbf{v}$.

2.1. Contrastive PCA.

Abid et al. [3,4] proposed projecting the positive samples onto the top k-dimensional eigen-subspace of the matrix difference

$$\mathbf{A}_\alpha := (1-\alpha)\mathbf{C}_{(+)} - \alpha\mathbf{C}_{(-)}, \qquad \alpha \in [0,1], \tag{1}$$

where $1 \leq k < d$ and α is the *contrast parameter*.[1]

To gain intuition, consider the special case where positive samples correspond to noisy signal and negative samples are pure noise; that is,

$$\mathbf{x}_t = \delta_t \times \text{signal} + \text{noise}. \tag{2}$$

If $\delta_t = 1$, then $\mathbf{x}_t$ is signal plus noise, whereas if $\delta_t = 0$, then $\mathbf{x}_t$ is just noise. Under the assumption that the signal and noise are uncorrelated, the covariance of the signal is equal to $2\mathbf{A}_{1/2} = \mathbf{C}_{(+)} - \mathbf{C}_{(-)}$, so the directions that maximize the variance of the signal are the top eigenvectors of $\mathbf{A}_{1/2}$.

The contrast parameter α represents the trade-off between maximizing the target variance and minimizing the background variance. When $\alpha = 0$, cPCA reduces to PCA applied to the positive samples. As $\alpha \to 1$, directions that reduce the variance of the negative samples become more optimal and the contrastive principal subspace is driven towards the minor subspace of the negative samples. Therefore, each value of α yields a direction with a different trade-off between positive and negative samples variance.

In practice, the performance of cPCA is sensitive to the contrast parameter α (see Fig. 2). While the generative model suggests that it should perform optimally when $\alpha = 1/2$, in practice, cPCA often performs better for $\alpha > 1/2$ (Fig. 2) and there is no theory for why this is true. Furthermore, since $\mathbf{A}_\alpha$ is not a covariance matrix for $\alpha \in (0,1)$, online PCA algorithms [1,2] cannot be adapted to cPCA.

2.2. Our method: Contrastive PCA*.

Our method, cPCA*, is based on projecting the positive samples onto the top k-dimensional eigen-subspace of the generalized eigenvalue problem

$$\mathbf{C}_{(+)}\mathbf{v} = \lambda \mathbf{B}_\beta \mathbf{v}, \tag{3}$$

where

$$\mathbf{B}_\beta := (1-\beta)\mathbf{I}_d + \beta\mathbf{C}_{(-)}, \qquad \beta \in [0,1], \tag{4}$$

and the hyper-parameter β is our *contrast parameter*.

To motivate cPCA*, consider the generative model in Eq. (2), under the assumption that the signal and noise are uncorrelated. Let $\mathbf{C}_{\text{signal}} = \mathbf{C}_{(+)} - \mathbf{C}_{(-)}$ and $\mathbf{C}_{\text{noise}} = \mathbf{B}_1 = \mathbf{C}_{(-)}$. Then the direction (i.e., unit vector $\mathbf{v}$) that maximizes the signal-to-noise ratio is the unit vector $\mathbf{v}$ that maximizes the ratio

$$\frac{\mathbf{v}^\top \mathbf{C}_{\text{signal}}\mathbf{v}}{\mathbf{v}^\top \mathbf{C}_{\text{noise}}\mathbf{v}} = \frac{\mathbf{v}^\top \mathbf{C}_{(+)}\mathbf{v}}{\mathbf{v}^\top \mathbf{B}_1\mathbf{v}} - 1.$$

The optimal $\mathbf{v}$ corresponds to the top eigenvector of the generalized eigenvalue problem (3) with $\beta = 1$.

As with the cPCA contrast parameter α, when $\beta = 0$, the problem reduces to PCA applied to positive samples. As $\alpha \to 1$ in cPCA, the projection subspace converges to the negative sample minor subspace, whereas as $\beta \to 1$ in cPCA*, the projection converges to maximize the ratio of the variance of the positive sample projections and the variance of the negative sample projections. For $\beta \in (0,1)$, our method is interpretable as maximizing the ratio of the variance of the positive sample projections and the variance of the negative sample projections when the negative samples have been conditioned with independent isotropic noise with variance $\frac{1-\beta}{\beta}$.

3. ONLINE cPCA* ALGORITHM

To derive an online algorithm, we concatenate the sequence of products $\delta_1\mathbf{x}_1, \ldots, \delta_T\mathbf{x}_T \in \mathbb{R}^d$ into the data matrix:

$$\mathbf{X}_{(+)} := [\delta_1\mathbf{x}_1, \ldots, \delta_T\mathbf{x}_T] \in \mathbb{R}^{d\times T}.$$

Since $\delta_t\mathbf{x}_t$ is zero for negative samples and equal to $\mathbf{x}_t$ for positive samples, the covariance of $\mathbf{X}_{(+)}$ is a scalar multiple of the covariance of the positive samples $\mathbf{C}_{(+)}$.

Similarity matching objective. Our starting point is the similarity matching objective:

$$\min_{\mathbf{Z}\in\mathbb{R}^{k\times T}} \frac{1}{T^2}\|\mathbf{Z}^\top\mathbf{Z} - \mathbf{X}_{(+)}^\top \mathbf{B}_\beta^{-1}\mathbf{X}_{(+)}\|_F^2. \tag{5}$$

The similarity matching objective, which is a special case of a cost function from multidimensional scaling [9], minimizes the difference in similarity (measured using inner products) between the outputs $\mathbf{Z}$ and the positive samples $\mathbf{X}_{(+)}$ normalized by $\mathbf{B}_\beta$. The optimal solution $\mathbf{Z}$ of equation (5) is the projection of $\mathbf{B}_\beta^{-1/2}\mathbf{X}_{(+)}$ onto its k-dimensional principal subspace [9], which is equal to the projection of $\mathbf{X}_{(+)}$ onto the top k-dimensional eigen-subspace of the generalized eigenvalue problem (3).

Legendre transforms. Directly optimizing Eq. (5) does not result in an online algorithm. Rather, we follow the approach in [10] for deriving an online algorithm. Expanding the square in Eq. (5) and dropping terms that do not depend on $\mathbf{Z}$ yields the minimization problem

$$\min_{\mathbf{Z}\in\mathbb{R}^{k\times T}} \frac{1}{T^2}\,\text{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{Z} - 2\mathbf{Z}^\top\mathbf{Z}\mathbf{X}_{(+)}^\top\mathbf{B}_\beta^{-1}\mathbf{X}_{(+)}\right).$$

[1] Abid et al. [4] find the top eigen-subspace of $\mathbf{C}_{(+)} - \alpha\mathbf{C}_{(-)}$ for $\alpha \geq 0$. We reparametrize the contrast parameter for a more direct comparison with our method.

We substitute in with the Legendre transforms:

$$\frac{1}{T^2}\operatorname{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{Z}\right) = \max_{\mathbf{M}\in\mathbb{S}^k_{++}} \frac{2}{T}\operatorname{Tr}\left(\mathbf{Z}^\top\mathbf{M}\mathbf{Z}\right) - \operatorname{Tr}\left(\mathbf{M}^2\right),$$

where $\mathbb{S}^k_{++}$ is the set of $k \times k$ positive definite matrices, and

$$\frac{1}{T^2}\operatorname{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\mathbf{X}_{(+)}^\top\mathbf{B}_\beta^{-1}\mathbf{X}_{(+)}\right) = \max_{\mathbf{W}\in\mathbb{R}^{k\times d}} \frac{2}{T}\operatorname{Tr}\left(\mathbf{Z}^\top\mathbf{W}\mathbf{X}_{(+)}\right) - \operatorname{Tr}\left(\mathbf{W}\mathbf{B}_\beta\mathbf{W}^\top\right),$$

which are respectively optimized at $\mathbf{M}^* = \frac{1}{T}\mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{W}^* = \frac{1}{T}\mathbf{Z}\mathbf{X}_{(+)}^\top\mathbf{B}_\beta^{-1}$. After substitution, we have

$$\min_{\mathbf{Z}\in\mathbb{R}^{k\times T}} \min_{\mathbf{W}\in\mathbb{R}^{k\times d}} \max_{\mathbf{M}\in\mathbb{S}^k_{++}} L(\mathbf{W},\mathbf{M},\mathbf{Z}),$$

where

$$L(\mathbf{W},\mathbf{M},\mathbf{Z}) := \frac{1}{T}\operatorname{Tr}(2\mathbf{Z}^\top\mathbf{M}\mathbf{Z} - 4\mathbf{Z}\mathbf{W}\mathbf{X}_{(+)}) - \operatorname{Tr}(\mathbf{M}^2 + 2\mathbf{W}\mathbf{B}_\beta\mathbf{W}^\top).$$

Since the objective satisfies the saddle-point property with respect to $\mathbf{M}$ and $\mathbf{Z}$, we can interchange the order of optimization:

$$\min_{\mathbf{W}\in\mathbb{R}^{k\times d}} \max_{\mathbf{M}\in\mathbb{S}^k_{++}} \min_{\mathbf{Z}\in\mathbb{R}^{k\times T}} L(\mathbf{W},\mathbf{M},\mathbf{Z}), \tag{6}$$

Offline algorithm. Before deriving an online algorithm, it is instructive to first solve Eq. (6) in the offline setting where we have access to the dataset $\mathbf{X}_{(+)}$ and the matrix $\mathbf{B}_\beta$. In this setting, we first minimize $L(\mathbf{W},\mathbf{M},\mathbf{Z})$ with respect to $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{M}^{-1}\mathbf{W}\mathbf{X}_{(+)}.$$

We then take gradient descent-ascent steps:

$$\mathbf{W} \leftarrow \mathbf{W} + 2\eta\left(\frac{1}{T}\mathbf{Z}\mathbf{X}_{(+)}^\top - \mathbf{W}\mathbf{B}_\beta\right) \tag{7}$$

$$\mathbf{M} \leftarrow \mathbf{M} + \frac{\eta}{\tau}\left(\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top - \mathbf{M}\right). \tag{8}$$

Here $\tau > 0$ denotes the ratio between the updates to $\mathbf{W}$ and $\mathbf{M}$ (which we set to 1 in our experiments) and $\eta \in (0,\tau)$ is the step size.

Online algorithm. To solve the minimax objective (6) in the online setting, we take stochastic gradient descent-ascent steps. At each time step t, we first minimize over the output $\mathbf{z}_t$ by running the continuous dynamics to equilibrium:

$$\dot{\mathbf{z}}_t(\gamma) = \delta_t\mathbf{c}_t - \mathbf{M}\mathbf{z}_t(\gamma) \qquad \Rightarrow \qquad \mathbf{z}_t = \delta_t\mathbf{M}^{-1}\mathbf{c}_t, \tag{9}$$

where we have defined $\mathbf{c}_t := \mathbf{W}\mathbf{x}_t$. Since $\delta_t = 0$ when $\mathbf{x}_t$ is a negative sample, the algorithm only produces a non-trivial output when presented with a positive sample. Next, we take a stochastic gradient descent-ascent step with respect to $(\mathbf{W},\mathbf{M})$. We can replace the averages $\frac{1}{T}\mathbf{Z}\mathbf{X}_{(+)}^\top$ and $\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top$ in equations (7) and (8) with the rank-1 approximations $\mathbf{z}_t\mathbf{x}_t^\top$ and $\mathbf{z}_t\mathbf{z}_t^\top$, respectively. To estimate $\mathbf{B}_\beta$ in the online setting, we note that

$$\mathbf{C}_{(-)} = \langle\mathbf{x}_t\mathbf{x}_t^\top|\delta_t = 0\rangle_t = \frac{\langle(1-\delta_t)\mathbf{x}_t\mathbf{x}_t^\top\rangle_t}{\langle 1-\delta_t\rangle_t}.$$

Therefore, we can replace $\mathbf{B}_\beta$ with the online estimate

$$\beta\frac{1-\delta_t}{p_t}\mathbf{x}_t\mathbf{x}_t^\top + (1-\beta)\mathbf{I}_d,$$

where p_t denotes a running estimate of $\langle 1-\delta_t\rangle_t$ that is initialized at $p_0 = 0.5$ and updated at each time step, as follows:

$$p_t = p_{t-1} + \frac{1}{t}(1 - \delta_t - p_{t-1}).$$

These substitutions result in the online updates

$$\mathbf{W} \leftarrow \mathbf{W} + 2\eta\left(\mathbf{z}_t - \beta\frac{1-\delta_t}{p_t}\mathbf{c}_t\right)\mathbf{x}_t^\top - 2\eta(1-\beta)\mathbf{W} \tag{10}$$

$$\mathbf{M} \leftarrow \mathbf{M} + \frac{\eta}{\tau}\left(\mathbf{z}_t\mathbf{z}_t^\top - \mathbf{M}\right). \tag{11}$$

We can map these onto a neural network, Fig. 1. Assuming that each neuron has access to the running average p_t, the learning rules are local in the sense that the update to a synaptic weight depends only on variables that are available in the pre- and postsynaptic neurons.

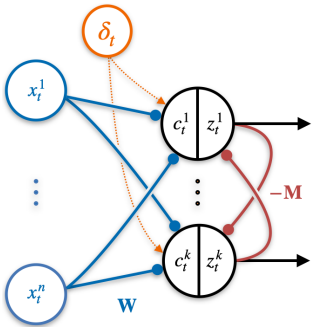

Fig. 1. Neural network implementation of our online cPCA* algorithm. At each t, the inputs $(x_t^1,\ldots,x_t^d)$ are projected onto the feedforward weights $\mathbf{W}$ to obtain $(c_t^1,\ldots,c_t^k)$. Lateral weights $-\mathbf{M}$ connect the output neurons. The neural dynamics in Eq. (9) are run to equilibrium and the output of the network is $(z_t^1,\ldots,z_t^k)$. The synaptic weights $\mathbf{W}$ and $-\mathbf{M}$ are then updated according to Eqs. (10) and (11).

4. NUMERICAL EXPERIMENTS

4.1. Comparison of cPCA and cPCA*

We compare cPCA and cPCA* on two naturalistic datasets used in [3] as well as a synthetic dataset. The results are sum-

marized in Fig. 3.

Artificial dataset. The artificial dataset consists of 200 30-dimensional samples. The negative samples are pure Gaussian noise while the positive samples exhibit a bimodal distribution along some of the dimensions. Due to high variance in the noisy directions, PCA is ineffective for identifying the clusters.

Noisy Digits dataset. In the first naturalistic dataset, the positive samples consist of 5,000 synthetic images generated by randomly superimposing images of handwritten digits 0 and 1 from MNIST dataset [11] on top of natural images taken from the UPenn Natural Image Database [12]. The negative samples consist of natural images. We apply cPCA and cPCA* to find 2-dimensional projections, for varying hyper-parameter choices, Figs. 2 and 3. To measure how well-separated the projections of the noisy digits 0 and 1 are, we plot the symmetrized KL divergence of the distributions of the projections of the noisy 0's and the distributions of the projections of the noisy 1's. We see that our method separates the digits for a much larger range of the hyper-parameter (Fig. 2).

Mouse dataset. The second naturalistic dataset consists of protein expression measurements ($d = 77$) of mice that have received shock therapy [13], some of which develop Down Syndrome. We apply cPCA and cPCA* (with $k = 2$) using negative samples that consist of protein expression measurements from a set of mice that have not been exposed to shock therapy and measure the degree of linear separability between mice that do not have Down Syndrome and mice that have Down Syndrome, Fig. 3.

4.2. Evaluation of online cPCA*

Fig. 4 shows the convergence of the online cPCA* algorithm on the mouse dataset. We see that cPCA* converges to the optimal projector. Here, the learning rate is $\eta = 0.003$ and the ratio between gradient descent and ascent steps is $\tau = 1$.

5. SUMMARY

In this work, we introduced cPCA*, a modified contrastive Principal Component Analysis method. We showed that this method is interpretable as maximizing the signal to noise ratio and leads to an online algorithm which can be mapped onto a neural network with local learning rules. In terms of the scope of applicability, cPCA* has the same requirements as cPCA: as a contrastive algorithm it needs a relevant background dataset. However, we hope that the derivation of this online algorithm with more robust hyper-parameter sensitivity will broaden the possible use-cases of this algorithm to larger datasets.

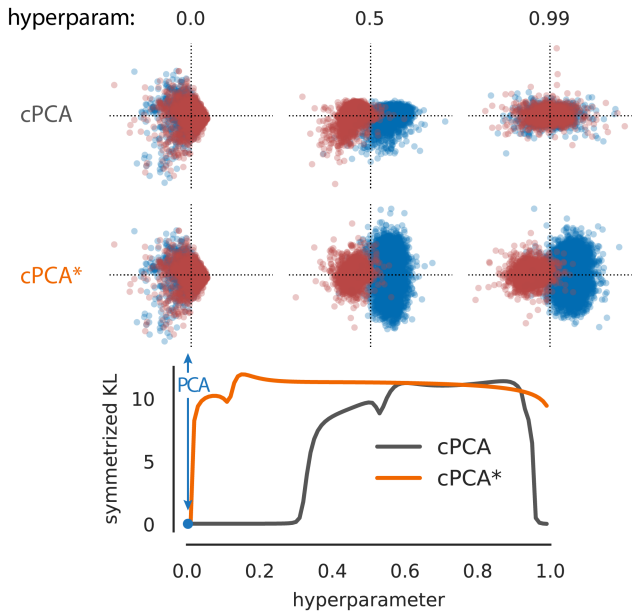

Fig. 2. cPCA* is more robust than cPCA to the choice of hyper-parameters. Separation of noisy MNIST digits 0 and 1 with (top) cPCA vs. (middle) cPCA*. Bottom: Quantitative measure of separation of the MNIST digits vs. value of hyper-parameter (α for cPCA and β for cPCA*).

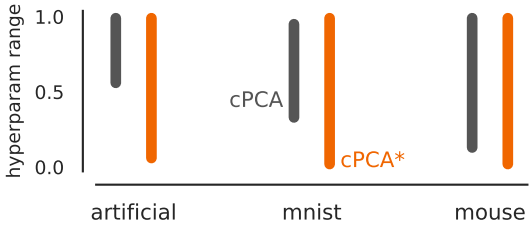

Fig. 3. Range of hyper-parameters (α for cPCA and β for cPCA*) that lead to good performance (greater than 90% classification accuracy using linear discriminant analysis) is broader in cPCA*.

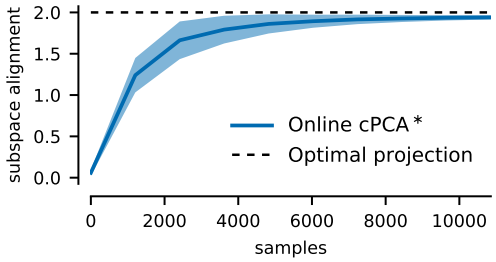

Fig. 4. Convergence of the online cPCA* algorithm on the mouse dataset. The y-axis shows the alignment of the projector found via the online algorithm to the optimal projector, computed by taking the trace of the product of the two projectors. Shaded region shows standard deviation over 5 runs of the experiment.

6. REFERENCES

[1] Zeyuan Allen-Zhu and Yuanzhi Li, "First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2017, pp. 487–492.

[2] Cengiz Pehlevan, Tao Hu, and Dmitri B Chklovskii, "A Hebbian/anti-Hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data," *Neural Computation*, vol. 27, no. 7, pp. 1461–1495, 2015.

[3] Abubakar Abid, Vivek Kumar Bagaria, Martin Jinye Zhang, and James Y. Zou, "Contrastive principal component analysis," *arXiv:1709.06716*, 2017.

[4] Abubakar Abid, Martin J Zhang, Vivek K Bagaria, and James Zou, "Exploring patterns enriched in a dataset with contrastive principal component analysis," *Nature Communications*, vol. 9, no. 1, pp. 1–7, 2018.

[5] Imrul Kaish, Jakir Hossain, Evangelos Papalexakis, and Jia Chen, "COVID-19 or flu? Discriminative knowledge discovery of COVID-19 symptoms from Google Trends data," *4th International Workshop on Epidemiology meets Data Mining and Knowledge discovery*, 2021.

[6] Micol Marchetti-Bowick, *Structured Sparse Regression Methods for Learning from High-Dimensional Genomic Data*, Ph.D. thesis, Carnegie Mellon University, 2020.

[7] Takanori Fujiwara, Oh-Hyun Kwon, and Kwan-Liu Ma, "Supporting analysis of dimensionality reduction results with contrastive learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 45–55, 2020.

[8] Junfeng Xiao and Di Zhang, "Full-reference image/video quality assessment algorithms based on contrastive principal component analysis," in *2022 7th International Conference on Image, Vision and Computing (ICIVC)*, 2022, pp. 648–653.

[9] Michael AA Cox and Trevor F Cox, "Multidimensional scaling," in *Handbook of Data Visualization*, pp. 315–347. Springer, 2008.

[10] David Lipshutz, Yanis Bahroun, Siavash Golkar, Anirvan M Sengupta, and Dmitri B Chklovskii, "A normative framework for deriving neural networks with multi-compartmental neurons and non-Hebbian plasticity," *arXiv preprint arXiv:2302.10051*, 2023.

[11] Yann LeCun, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[12] Gašper Tkačik, Patrick Garrigan, Charles Ratliff, Grega Milčinski, Jennifer M Klein, Lucia H Seyfarth, Peter Sterling, David H Brainard, and Vijay Balasubramanian, "Natural images from the birthplace of the human eye," *PLOS One*, vol. 6, no. 6, pp. e20409, 2011.

[13] Clara Higuera, Katheleen J Gardiner, and Krzysztof J Cios, "Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome," *PLOS One*, vol. 10, no. 6, pp. e0129126, 2015.