Learning Low-Dimensional Structure via Closed-Loop Transcription: Equilibria and Optimization

Druv Pai

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-74
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-74.html

May 9, 2023

Learning Low-Dimensional Structure via Closed-Loop Transcription: Equilibria and Optimization

by

Druv Pai

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

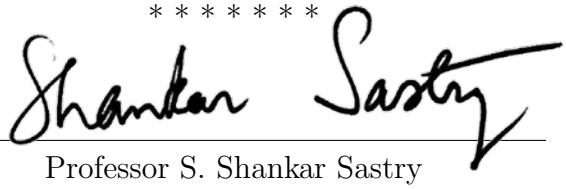
Committee:

Professor Yi Ma
Research Advisor

May 3, 2023

(Date)

* * * * * * *



Professor S. Shankar Sastry
Second Reader

05/9/23

(Date)

Abstract

We consider the problem of learning maximally informative representations for data in a high-dimensional space with distribution supported on or around a single or multiple low-dimensional geometric structures, with or without labels. That is, we wish to compute a linear injective map (i.e., an "encoder") such that the image of the data (i.e., the "representations") have maximum "information gain"; we also want to compute a a suitable notion of inverse for the encoder (i.e., a "decoder"). We formulate this family of learning problems as a class of two-player games. For a broad notion of game-theoretic equilibria which is learnable via standard gradient-based optimization techniques, we show that the equilibrium solutions to games within the class indeed result in maximally informative representations and a consistent autoencoding. We then apply this framework to several instances of the closed-loop transcription (CTRL) framework, which has been recently proposed for learning discriminative and generative representations for data lying on low-dimensional submanifolds, obtaining desirable representations which provably emulate and extend ones given by classical theory. Finally, we present a novel optimization algorithm to obtain the particular equilibria which our theory desires, and prove its correctness in a restricted case.

Acknowledgments

First, I would like to thank my advisor, Professor Yi Ma. Since I started working with him as a third-year undergraduate, Professor Ma has always pushed me to improve as a researcher and spurred my growth. In addition, he has been a fount of good ideas and excellent problems. For this, and many other reasons, I am very grateful.

I would also like to thank Professor Shankar Sastry for his continual support, both on the technical level and on a big-picture level, as well as for being the second reader for this report.

The ideas in this work were contained in a couple of publications. I would like to thank all of my collaborators on those publications, including (but not limited to) Professors Edgar Dobriban and Manxi Wu; and fellow students Michael Psenka and Chih-Yuan Chiu. I also worked with several other collaborators on various other projects; in particular, I am grateful to have collaborated with Dr. Sam Buchanan, Xili Dai, Peter Tong, Vishal Raman, Brent Yi, and Chinmay Maheshwari.

This thesis is dedicated to my friends and family, whose support is invaluable.

Contents

Notice

This work is based primarily on the paper (Pai et al., 2022) as well as a final project for CS 294-182 in Spring 2023, but contains additional material not submitted to either venue.

1 Motivation and Context

Learning representations of complex high-dimensional data with low underlying complexity is a central goal in machine learning, with applications to compression, sampling, out-of-distribution detection, classification, etc. For example, in the context of image data, one may perform clustering (Prasad et al., 2020), and generate or detect fake images (Huang et al., 2018). There are a number of recently popular methods for representation learning, several proposed in the context of image generation; one such example is generative adversarial networks (GANs) (Goodfellow et al., 2014), giving promising results (Karras et al., 2021; Mino and Spanakis, 2018). Despite empirical successes, theoretical understanding of representation learning of high-dimensional data with underlying low complexity is still rather primitive. Classical methods with theoretical guarantees (Jolliffe, 2002), such as principal component analysis (PCA), are divorced from modern methods such as GANs whose justifications are mostly empirical and whose theoretical properties remain poorly understood (Feizi et al., 2020; Farnia and Ozdaglar, 2020).

A challenge for our theoretical understanding is that *high-dimensional data often has low-dimensional structure*, such as belonging to multiple subspaces and even nonlinear manifolds (Wright and Ma, 2022; Li and Bresler, 2018; Zhang et al., 2019; Shen et al., 2020; Zhai et al., 2020, 2019; Qu et al., 2019; Lau et al., 2020; Fefferman et al., 2016). This hypothesis can be difficult to account for theoretically.[1] In fact, our understanding of this setting, and knowledge of principled and generalizable solutions, is still incomplete, even in the case when the data lies on multiple linear subspaces (Vidal et al., 2016), and the representation map is linear. In this work, we aim to bridge this gap.

1.1 Our Contributions

Our contributions are four-fold:

1. We propose a new game-theoretic framework, called CTRL-PG, for learning injective and discriminative representations for geometrically structured high-dimensional data.
2. We mathematically characterize the equilibrium representations of CTRL-PG games and show they fulfill a set of commonly-desired properties of representations, proving that our framework is well-posed and theoretically principled.
3. We apply our framework to classical but complex subspace learning problems, and show that our framework recovers the optimal results constructed by classical theory in this setting.
4. We propose an algorithm to learn such equilibria, with a proof of correctness under benign assumptions.

Our results demonstrate an instance of classical machine learning problems being solved optimally using modern deep learning tools, thus unifying the classical and modern perspectives on machine learning. Our analysis is tailored to fit the assumption of high-dimensional data with low-dimensional structure.

1.2 Related Works

PCA, Subspace Clustering, and Autoencoding Principal component analysis (PCA) and its probabilistic versions (Hotelling, 1933; Tipping and Bishop, 1999) are a classical tool for learning low-dimensional representations. One finds the best ℓ^2-approximating subspace of a given dimension for the data. Thus, PCA can be viewed as *seeking to learn the linear subspace structure of the data.* Several generalizations of PCA exist. Generalized PCA (GPCA) (Vidal et al., 2003) *seeks to learn multiple linear subspace structure* by clustering. Unlike PCA and this work, GPCA does not learn transformed representations of the data. PCA has also been adapted to recover nonlinear structures in many ways (Van Der Maaten et al., 2009), e.g., via principal curves (Hastie and Stuetzle, 1989) or autoencoders (Kramer, 1991).

[1] One assumption which violates this hypothesis implicitly is the existence of a probability density for the data. For instance, the analysis in several prominent works on representation learning, such as Kingma and Welling (2014) and Feizi et al. (2020) critically requires this assumption to hold. Probability densities with respect to the Lebesgue measure on $\mathbb{R}^n$ do not exist if the underlying probability measure has a Lebesgue measure zero support, e.g., for lower-dimensional structures such as subspaces (Kallenberg, 2021). Thus, this assumption excludes a lower dimensionality of the data.

GAN Generative Adversarial Networks (GANs) are a recently popular representation learning method (Goodfellow et al., 2014; Arjovsky et al., 2017). GANs simultaneously learn a generator function, which maps low-dimensional noise to the data distribution, and a discriminator function, which maps the data to discriminative representations from which one can classify the data as authentic or synthetic with a simple predictor. The generator and discriminator are trained adversarially; the generator is trained to generate data which is distributionally close to real data, in order to fool the discriminator, while the discriminator is simultaneously trained to identify discrepancies between the generator output and empirical data.

While GANs enjoy certain empirical success (see e.g. Karras et al. (2021); Mino and Spanakis (2018)), their theoretical properties are less well developed, especially in the context of high-dimensional data with intrinsic structure. More specifically, the most prominent works of GAN analysis use the simplifying assumption of full-rank data (Feizi et al., 2020), require explicit computation of objective functions which are intractable to even estimate using a finite sample (Arjovsky et al., 2017; Zhu et al., 2020), or show that GANs have poor theoretical behavior, such as their training game not having Nash equilibria (Farnia and Ozdaglar, 2020). In this work, we adopt the more realistic assumption of low-dimensional data in a high-dimensional space, use explicit, closed-form objective functions which are more convenient to optimize, and demonstrate the existence of global equilibria of the training game corresponding to our method.

1.3 Notation

In this section we give an accounting of common notations which we use.

- We denote by $\mathbb{R}_{\geq 0}$ the set of all non-negative real numbers.
- For $n \geq 1$ a positive integer, we denote by $[n]$ the set $\{1, \dots, n\}$.
- For two sets U, V, a map $h \colon U \to V$, and a subset $W \subseteq U$, we denote by $h(W) \doteq \{h(u) \mid u \in W\}$ the image of W under h.
- For a vector space U and subspaces $V, W \subseteq U$, we denote by $V + W \doteq \{\boldsymbol{v} + \boldsymbol{w} \mid \boldsymbol{v} \in V, \boldsymbol{w} \in W\}$ the Minkowski sum of V and W.
- For a set U, we denote by $\mathrm{id}_U \colon U \to U$ the identity map on U.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we denote by $\mathrm{Col}(\boldsymbol{A}) \doteq \{\boldsymbol{y} \in \mathbb{R}^m \mid \exists \boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$ the image of $\boldsymbol{A}$.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we denote by $\mathrm{Null}(\boldsymbol{A}) \doteq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$ the kernel of $\boldsymbol{A}$.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we denote $\|\boldsymbol{A}\|_F \doteq \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$ to be the Frobenius norm of $\boldsymbol{A}$.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we denote $\boldsymbol{A}^+$ to be the Moore-Penrose psuedoinverse of $\boldsymbol{A}$.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a positive integer $i \leq \min\{m, n\}$, we denote by $\sigma_i(\boldsymbol{A})$ the i^{th} largest singular value of $\boldsymbol{A}$.
- For a symmetric matrix $\boldsymbol{A} \in \mathbb{S}^n$, we denote by $\lambda_{\max}(\boldsymbol{A})$ and $\lambda_{\min}(\boldsymbol{A})$ the minimum and maximum singular values of $\boldsymbol{A}$ respectively.
- For a symmetric matrix $\boldsymbol{A} \in \mathbb{S}^{n \times n}$ and a positive integer $i \leq n$, we denote by $\lambda_i(\boldsymbol{A})$ as the i^{th} largest eigenvalue of $\boldsymbol{A}$.
- For vector spaces U, V, we denote $\mathsf{L}(U, V)$ to be the set of linear maps $U \to V$.
- For a vector space U and a positive integer $d \leq \dim(U)$, we denote by $\mathsf{Gr}(V, d)$ the vector space of d-dimensional subspaces of V.
- For a normed vector space U and subspace $V \subseteq U$, we denote the projection onto V as $\mathcal{P}_V \colon U \to V$.
- For inner product spaces U, V and a linear map $h \colon U \to V$, we denote the adjoint of h by $h^* \colon V \to U$.
- For inner product spaces U, V, we denote $\mathsf{O}(U, V)$ to be the set of orthogonal linear maps $U \to V$:

$$\mathsf{O}(U, V) \doteq \left\{ h \in \mathsf{L}(U, V) \;\middle|\; \begin{array}{ll} h^* \circ h = \mathrm{id}_U, & \text{if } \dim(U) \leq \dim(V) \\ h \circ h^* = \mathrm{id}_V, & \text{if } \dim(U) \geq \dim(V) \end{array} \right\}. \tag{1}$$

2 Preliminaries

2.1 Representation Learning

Let $\boldsymbol{x}$ be a random variable taking values in $\mathbb{R}^D$. Let $\boldsymbol{X} = \left[\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\right] \in \mathbb{R}^{D \times N}$ be a sample matrix whose columns are $N \geq 1$ samples $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N \in \mathbb{R}^D$ which are i.i.d. realizations of $\boldsymbol{x}$.

Our goal is to learn an encoder mapping $f_\theta \colon \mathbb{R}^D \to \mathbb{R}^d$ from some parametric function family $\{f_\theta \colon \mathbb{R}^D \to \mathbb{R}^d \mid \theta \in \Theta\}$, given $\boldsymbol{X}$ and any auxiliary data (such as labels, as we will see later). Normally, we want $d \leq D$ and $f_\theta(\boldsymbol{x})$ to have better geometric and statistical properties than $\boldsymbol{x}$ itself. Moreover, we want to learn an approximate *inverse* or decoder mapping $g_\xi \colon \mathbb{R}^d \to \mathbb{R}^D$ from another parametric function family $\{g_\xi \colon \mathbb{R}^d \to \mathbb{R}^D \mid \xi \in \Xi\}$, such that the distributions of $\boldsymbol{x}$ and $(g_\xi \circ f_\theta)(\boldsymbol{x})$ are close.

2.2 Closed-Loop Transcription

To learn the encoder/decoder mappings f_θ and g_ξ, we use the Closed-Loop Transcription (CTRL) framework, a recent method which was proposed for representation learning of low-dimensional submanifolds in high-dimensional space and has had good empirical results (Dai et al., 2022). This framework generalizes *both* autoencoders and GANs; f_θ has dual roles as an encoder and discriminator, and g_ξ has dual roles as a decoder and a generator.

For the sample matrix $\boldsymbol{X}$, we define $f_\theta(\boldsymbol{X}) \doteq \left[f_\theta(\boldsymbol{x}^1), \ldots, f_\theta(\boldsymbol{x}^N)\right] \in \mathbb{R}^{d \times N}$. The training process follows a *closed loop*: starting with the data $\boldsymbol{X}$ and the autoencoded data $(g_\xi \circ f_\theta)(\boldsymbol{X})$, the data representations $f_\theta(\boldsymbol{X})$ and the autoencoded data representations $(f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{X})$ are used to train θ and ξ. This approach has a crucial advantage over the GAN formulation: contrary to GANs (Arjovsky et al., 2017; Zhu et al., 2020), since $f_\theta(\boldsymbol{X})$ and $(f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{X})$ both live in the structured representation space $\mathbb{R}^d$, interpretable quantifications of representation quality and of the difference between $f_\theta(\boldsymbol{X})$ and $(f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{X})$ exist and may be computed *efficiently in closed form*.

For convenience, hereafter we use the following notations to denote encoder-decoder compositions:

$$\begin{array}{lll} \boldsymbol{z}(\theta) \doteq f_\theta(\boldsymbol{x}), & \hat{\boldsymbol{x}}(\theta, \xi) \doteq (g_\xi \circ f_\theta)(\boldsymbol{x}), & \hat{\boldsymbol{z}}(\theta, \xi) \doteq (f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{x}) \\ \boldsymbol{z}^i(\theta) \doteq f_\theta(\boldsymbol{x}^i), & \hat{\boldsymbol{x}}^i(\theta, \xi) \doteq (g_\xi \circ f_\theta)(\boldsymbol{x}^i), & \hat{\boldsymbol{z}}^i(\theta, \xi) \doteq (f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{x}^i), \quad \forall i \in [N] \\ \boldsymbol{Z}(\theta) \doteq f_\theta(\boldsymbol{X}), & \hat{\boldsymbol{X}}(\theta, \xi) \doteq (g_\xi \circ f_\theta)(\boldsymbol{X}), & \hat{\boldsymbol{Z}}(\theta, \xi) \doteq (f_\theta \circ g_\xi \circ f_\theta)(\boldsymbol{X}). \end{array} \tag{2}$$

2.3 Rate Reduction

These tractable quantities are based on the information-theoretic and statistical paradigm of *rate reduction* discussed in the CTRL literature (Dai et al., 2022; Yu et al., 2020) as well as previous works (Ma et al., 2007). Here we review the main principles, as they are central to an information-theoretic interpretation of our objective functions.

Let $\boldsymbol{z}$ be a random variable taking values in $\mathbb{R}^d$. Let $\mathrm{RD}(\cdot \mid \boldsymbol{z})$ be the rate distortion function of $\boldsymbol{z}$ with respect to the Euclidean squared distance distortion (Cover and Thomas, 2006). Information-theoretically, this is the *coding rate* of the data; that is, the average number of bits required to encode $\boldsymbol{z}$, such that the expected Euclidean squared distance between $\boldsymbol{z}$ and its encoding is at most the first argument of the function.

If $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Gamma})$ is a multivariate Gaussian random vector with mean $\boldsymbol{0}_d$ and covariance $\boldsymbol{\Gamma}$, then

$$\mathrm{RD}(\varepsilon \mid \boldsymbol{u}) = \frac{1}{2} \log_2 \det\left(\frac{d}{\varepsilon^2}\boldsymbol{\Gamma}\right) \qquad \forall \varepsilon \in \left[0, \sqrt{d \cdot \lambda_{\min}(\boldsymbol{\Gamma})}\right]. \tag{3}$$

For larger ε, the rate distortion function becomes more complicated and can be found by the water-filling algorithm on the eigenvalues of $\boldsymbol{\Gamma}$. However, Ma et al. (2007) proposes the following approximation of the rate distortion. For $\boldsymbol{w}_\varepsilon \sim \mathcal{N}\left(\boldsymbol{0}_d, \frac{\varepsilon^2}{d}\boldsymbol{I}_d\right)$ independent of $\boldsymbol{z}$, let

$$R_\varepsilon(\boldsymbol{z}) \doteq \mathrm{RD}(\varepsilon \mid \boldsymbol{z} + \boldsymbol{w}_\varepsilon). \tag{4}$$

If $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma})$, then we may derive a closed form expression for $R_\varepsilon(\boldsymbol{z})$ for *all* $\varepsilon > 0$. Since $\boldsymbol{z}$ and $\boldsymbol{w}_\varepsilon$ are normally distributed and independent, so is $\boldsymbol{z} + \boldsymbol{w}_\varepsilon$, and

$$\boldsymbol{z} + \boldsymbol{w}_\varepsilon \sim \mathcal{N}\left(\boldsymbol{0}_d, \frac{\varepsilon^2}{d}\boldsymbol{I}_d + \boldsymbol{\Sigma}\right). \tag{5}$$

Thus,

$$\sqrt{d \cdot \lambda_{\min}\left(\frac{\varepsilon^2}{d}\boldsymbol{I}_d + \boldsymbol{\Sigma}\right)} = \sqrt{d \cdot \left(\frac{\varepsilon^2}{d} + \lambda_{\min}(\boldsymbol{\Sigma})\right)} = \sqrt{\varepsilon^2 + d\lambda_{\min}(\boldsymbol{\Sigma})} \geq \varepsilon. \tag{6}$$

Therefore, we have the following closed form expression for $R_\varepsilon(\boldsymbol{z})$ for *all* $\varepsilon > 0$.

$$R_\varepsilon(\boldsymbol{z}) = \mathrm{RD}(\varepsilon \mid \boldsymbol{z} + \boldsymbol{w}_\varepsilon) = \frac{1}{2}\log_2 \det\left(\frac{d}{\varepsilon^2}\left(\boldsymbol{\Sigma} + \frac{\varepsilon^2}{d}\boldsymbol{I}_d\right)\right) \tag{7}$$

$$= \frac{1}{2}\log_2 \det\left(\boldsymbol{I}_d + \frac{d}{\varepsilon^2}\boldsymbol{\Sigma}\right). \tag{8}$$

In information-theoretic terms, $R_\varepsilon(\boldsymbol{z})$ is a *regularized rate distortion* function. Heuristically, it counts the average number of bits required to encode $\boldsymbol{z}$ up to ε precision, and thus it quantifies the expansiveness of the distribution of $\boldsymbol{z}$, or in other words how "spread out" the distribution is.

From this quantity we can also define a difference function[2] between distributions of two possibly-correlated random vectors $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^d$. This function approximately computes the average number of bits saved by encoding $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ separately and independently compared to encoding them together, say by encoding a mixture random variable $\boldsymbol{z}$ which is $\boldsymbol{z}_1$ with probability $\frac{1}{2}$ and $\boldsymbol{z}_2$ with probability $\frac{1}{2}$, up to precision ε. In this notation, we have

$$\Delta R_\varepsilon(\boldsymbol{z}_1, \boldsymbol{z}_2) \doteq R_\varepsilon(\boldsymbol{z}) - \frac{1}{2}R_\varepsilon(\boldsymbol{z}_1) - \frac{1}{2}R_\varepsilon(\boldsymbol{z}_2). \tag{9}$$

This difference function has several advantages over Wasserstein or Jensen-Shannon distances. It is a principled quantification of difference which is computable in closed-form for the widely representative class of Gaussian distributions. In particular, due to the existence of the closed-form representation, it is much simpler to do analysis on the solutions of optimization problems involving this function.

We may generalize the difference function to several random vectors. Specifically, define probabilities $\pi_1, \ldots, \pi_K \in [0, 1]$ such that $\sum_{j=1}^K \pi_j = 1$, arranged in a vector $\boldsymbol{\pi} \in [0, 1]^K$, and let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$ be random variables taking values in $\mathbb{R}^d$. Define $\boldsymbol{z}$ to be the mixture random vector which equals $\boldsymbol{z}_j$ with probability π_j. Then the *coding rate reduction* of $\boldsymbol{z}$ given $\boldsymbol{\pi}$ is given by

$$\Delta R_\varepsilon(\boldsymbol{z} \mid \boldsymbol{\pi}) \doteq R_\varepsilon(\boldsymbol{z}) - \sum_{j=1}^K \pi_j R_\varepsilon(\boldsymbol{z}_j). \tag{10}$$

Heuristically, this again approximates the average number of bits saved by encoding each $\boldsymbol{z}_j$ separately as opposed to encoding $\boldsymbol{z}$ as a whole, and thus it quantifies how compact the distribution of each $\boldsymbol{z}_j$ is and how expansive, or "spread out" the distribution of $\boldsymbol{z}$ as a whole is. More precisely, it was shown by Yu et al. (2020) that, subject to rank and Frobenius norm constraints on the $\boldsymbol{z}_j$, this expression is maximized when the $\boldsymbol{z}_j$ are distributed on pairwise orthogonal subspaces, and also each $\boldsymbol{z}_j$ has isotropic (or nearly isotropic) covariance on its subspace.

In practice, we do not know the distribution of the data, and the features are not perfectly a mixture of Gaussians. Still, the mixture of Gaussians is often a reasonable model for lower-dimensional feature distributions (Ma et al., 2007; Yu et al., 2020; Dai et al., 2022), so we use the Gaussian form for the approximate coding rate.

Also, in practice we may not have access to the full distribution of data, and so we need to estimate all relevant quantities via a finite sample. For Gaussians, R_ε is only a function of $\boldsymbol{z}$ through its covariance $\boldsymbol{\Sigma}$;

[2]Unfortunately, it is not a true distance function; for starters, it can be zero for random variables with non-identical distributions.

in practice, this covariance is estimated via a finite sample $\boldsymbol{Z} \in \mathbb{R}^{d \times N}$, assumed to be centered, as $\boldsymbol{Z}\boldsymbol{Z}^*/N$. This also allows us to estimate $\Delta R_\varepsilon(\cdot, \cdot)$ from a finite sample. To estimate $\Delta R_\varepsilon(\cdot \mid \cdot)$, we also need to estimate $\boldsymbol{\pi}$. For this, we require finite sample label information $\boldsymbol{y}$ telling us which samples correspond to which random vector $\boldsymbol{z}_j$. Denote by $N_j \geq 1$ the number of samples in $\boldsymbol{Z}$ which correspond to $\boldsymbol{z}_j$. Then $\boldsymbol{\pi}$ may be estimated via plug-in as $\hat{\pi}_j = \frac{N_j}{N}$.

This set of approximations yields estimates $R_\varepsilon(\boldsymbol{Z})$, $\Delta R_\varepsilon(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$, and $\Delta R_\varepsilon(\boldsymbol{Z} \mid \boldsymbol{y})$. Henceforth, we drop the ε subscript and use the natural logarithm instead of the base-2 logarithm. In this notation, the expressions for Gaussian $\boldsymbol{z}$ and $\boldsymbol{z}_j$, which we use in practice, are:

$$R(\boldsymbol{Z}) = \frac{1}{2} \operatorname{logdet}\left(\boldsymbol{I}_d + \frac{d}{N\varepsilon^2}\boldsymbol{Z}\boldsymbol{Z}^*\right), \tag{11}$$

$$\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2) = R(\big[\boldsymbol{Z}_1, \boldsymbol{Z}_2\big]) - \frac{1}{2}R(\boldsymbol{Z}_1) - \frac{1}{2}R(\boldsymbol{Z}_2), \tag{12}$$

$$\Delta R(\boldsymbol{Z} \mid \boldsymbol{y}) = R(\big[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_K\big]) - \sum_{j=1}^{K} \frac{N_j}{N} R(\boldsymbol{Z}_j). \tag{13}$$

We note here that although the assumption that $\boldsymbol{z}$ and $\boldsymbol{z}_j$ are Gaussian provides an information-theoretic interpretation of the coding rate and rate reduction, our results in this work *do not* rely on anything being exactly distributed according to a mixture of Gaussians. This is because the proofs use purely the *algebraic* properties of the coding rate approximations.

We finish this section with two key deterministic results which provide intuition to the behavior of optimizing with respect to rate reduction functions.

Proposition 2.1. *Let $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \in \mathbb{R}^{d \times N}$. Then $\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2) \geq 0$. Furthermore, if $\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2) = 0$, then $\operatorname{Col}(\boldsymbol{Z}_1) = \operatorname{Col}(\boldsymbol{Z}_2)$.*

Proof. By (Yu et al., 2020, Lemma A.4), we have $\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2) \geq 0$, with equality if and only if $\boldsymbol{Z}_1\boldsymbol{Z}_1^* = \boldsymbol{Z}_2\boldsymbol{Z}_2^*$, implying that $\operatorname{Col}(\boldsymbol{Z}_1) = \operatorname{Col}(\boldsymbol{Z}_2)$. □

This result says that minimizing $\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ matches the underlying span of the two inputs' columns.

Proposition 2.2 (Theorem A.6 of Yu et al. (2020))**.** *Let $\boldsymbol{y} \in [K]^N$ be finite sample label information, and let $N_1, \ldots, N_K$ be positive integers such that*

$$N_j \doteq \sum_{i=1}^{N} \mathbf{1}[y_i = j], \qquad \forall j \in [K]. \tag{14}$$

Let $d_1, \ldots, d_K$ be positive integers. Consider the following optimization problem:

$$\boldsymbol{Z}_1^\star, \ldots, \boldsymbol{Z}_K^\star \in \underset{\substack{\boldsymbol{Z}_j \in \mathbb{R}^{d \times N_j} \\ \forall j \in [K]}}{\operatorname{argmin}} \quad \Delta R(\big[\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_j\big] \mid \boldsymbol{y}) \tag{15}$$

$$s.t. \quad \|\boldsymbol{Z}_j\|_F^2 = N_j, \qquad \forall j \in [K] \tag{16}$$

$$\operatorname{rank}(\boldsymbol{Z}_j) \leq d_j, \qquad \forall j \in [K]. \tag{17}$$

Suppose that the following conditions hold:

(i) (Large ambient dimension.) $d \geq \sum_{j=1}^{K} d_j$.

(ii) (High coding precision.) $\varepsilon^4 < \min_{j \in [K]} \left\{ \frac{N_j}{N} \cdot \frac{d^2}{d_j^2} \right\}$.

Then the optimal solutions $\boldsymbol{Z}_1^\star, \ldots, \boldsymbol{Z}_K^\star$ satisfy:

(i) Between-class discriminative: $\operatorname{Col}(\boldsymbol{Z}_j^\star)$ and $\operatorname{Col}(\boldsymbol{Z}_k^\star)$ are orthogonal subspaces for all $j \neq k$.

(ii) Within-class diverse: For each $j \in [K]$, we have $\sigma_1(\boldsymbol{Z}_j) = \cdots = \sigma_{d_j - 1}(\boldsymbol{Z}_j) \geq \sigma_{d_j}(\boldsymbol{Z}_j) > 0$.

This result says that maximizing $\Delta R(\boldsymbol{Z} \mid \boldsymbol{y})$ subject to norm and dimension constraints means that each $\boldsymbol{Z}_j$ is distributed nearly isotropically on a subspace of the largest possible dimension.

2.4 Game Theory and Equilibria

We now discuss how to train the encoder parameters θ and decoder parameters ξ. Let the parameter spaces of the encoder and decoder be Θ and Ξ respectively. For simplicity we assume that they are metric subspaces of Euclidean space, with ρ_Θ and ρ_Ξ denoting their respective (Euclidean) metrics.

Several methods, e.g., PCA, GANs (Goodfellow et al., 2014), and the original CTRL formulation (Dai et al., 2022), can be viewed as learning the encoder (or discriminator) parameters θ and decoder (or generator) parameters ξ via finding the Nash equilibria of an appropriate two-player zero-sum game between the encoder and decoder. In this work, we approach this problem from a more general perspective; we learn the encoder parameters θ and decoder parameters ξ via finding the so-called *proximal equilibria* of the training game. We now cover the basics of two-player zero-sum game theory in the context of our representation learning problem; a more complete treatment is found in Başar and Olsder (1998).

In a two-player zero-sum game between the encoder — whose move corresponds to picking $\theta \in \Theta$ — and decoder — whose move corresponds to picking $\xi \in \Xi$ — the encoder attempts to *maximize* a value function $\mathcal{V}\colon \Theta \times \Xi \to \mathbb{R}$, while the decoder attempts to *minimize* it. In a *simultaneous* game, both players make their moves at the same time, with no information about the others' play. In a *sequential game*, the players make their moves one at a time; in our formulation, the *encoder* would move first, since conceptually the decoder solely wishes to invert the encoder and can only do effectively so with knowledge of the encoder's play.

We now introduce the traditional solution concepts — that is, the equilibria which may be learned by an algorithm — for simultaneous and sequential games. The solution concept for simultaneous games is the celebrated *Nash equilibrium* (Başar and Olsder, 1998), defined below.

Definition 2.3 (Nash Equilibrium)**.** The pair $(\theta^\star, \xi^\star) \in \Theta \times \Xi$ is a *Nash equilibrium* if

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \Theta} \mathcal{V}(\theta, \xi^\star) \qquad \text{and} \qquad \xi^\star \in \operatorname*{argmin}_{\xi \in \Xi} \mathcal{V}(\theta^\star, \xi). \tag{18}$$

We denote by NE the set of Nash equilibria.

In words, neither the encoder nor the decoder wish to *unilaterally* deviate from $(\theta^\star, \xi^\star)$. This reflects the simultaneous notion of the game in that neither player in the game knows the other player's actions.

The solution concept for sequential games is the Stackelberg equilibrium (Başar and Olsder, 1998; Fiez et al., 2019; Jin et al., 2020), defined below in the case that the encoder moves first.

Definition 2.4 (Stackelberg Equilibrium)**.** The pair $(\theta^\star, \xi^\star) \in \Theta \times \Xi$ is a *Stackelberg equilibrium* if

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \Theta} \inf_{\xi \in \Xi} \mathcal{V}(\theta, \xi) \qquad \text{and} \qquad \xi^\star \in \operatorname*{argmin}_{\xi \in \Xi} \mathcal{V}(\theta^\star, \xi). \tag{19}$$

We denote by SE the set of Stackelberg equilibria.

The sequential notion of the game is reflected in the definition of the equilibrium; the decoder, going second, may play ξ to minimize $\mathcal{V}(\theta, \cdot)$ with full knowledge of the encoder's play θ (assuming the encoder plays rationally), while the encoder plays θ to maximize $\mathcal{V}(\cdot, \cdot)$ with only the knowledge that the decoder will play optimally in response.

2.5 Proximal Equilibria

To accommodate the wide variety of optimization strategies which are used in practice to learn (θ, ξ) pairs, we instead study a generalization of both Nash and Stackelberg equilibria — the so-called *proximal equilibrium*, first defined in Farnia and Ozdaglar (2020).

Definition 2.5 (Proximal Equilibrium)**.** Let $\lambda > 0$. We define the *proximal value function* $\mathcal{V}_\lambda\colon \Theta \times \Xi \to \mathbb{R}$ by

$$\mathcal{V}_\lambda(\theta, \xi) \doteq \inf_{\zeta \in \Xi} \left\{ \mathcal{V}(\theta, \zeta) + \frac{\lambda}{2} \rho_\Xi(\xi, \zeta)^2 \right\}. \tag{20}$$

The pair $(\theta^\star, \xi^\star) \in \Theta \times \Xi$ is a λ-*proximal equilibrium* if

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \Theta} \mathcal{V}_\lambda(\theta, \xi^\star) \qquad \text{and} \qquad \xi^\star \in \operatorname*{argmin}_{\xi \in \Xi} \mathcal{V}(\theta^\star, \xi). \tag{21}$$

We denote by $\mathsf{PE}(\lambda)$ the set of λ-proximal equilibria.

Remark 2.6. *The original definition of the proximal equilibrium by Farnia and Ozdaglar (2020) replaces the condition* $\xi^\star \in \operatorname{argmin}_{\xi\in\Xi} \mathcal{V}(\theta^\star, \xi)$ *with the condition* $\xi^\star \in \operatorname{argmin}_{\xi\in\Xi} \mathcal{V}_\lambda(\theta^\star, \xi)$*, but these are equivalent since it is easy to see that for any* $\lambda > 0$ *and* $\theta \in \Theta$ *we have*

$$\operatorname*{argmin}_{\xi\in\Xi} \mathcal{V}_\lambda(\theta, \xi) = \operatorname*{argmin}_{\xi\in\Xi} \mathcal{V}(\theta, \xi). \tag{22}$$

We present the proximal equilibrium definition as in Definition 2.5 in order to make clear the connections between proximal equilibria, Nash equilibria, and Stackelberg equilibria.

Proximal equilibria encourage some optimization around the Nash equilibrium solution, which is consistent with the convergence of alternating gradient descent-ascent to stable points in minimax training, so there is reason to believe that such equilibria are obtained in practice using standard minimax optimization algorithms. In the remainder of the work, we analyze the proximal equilibrium.

The key way to think about proximal equilibria is that *proximal equilibria interpolate between Nash equilibria and Stackelberg equilibria.* This is formalized in the following result.

Proposition 2.7 (Adaptation of Proposition 3 of Farnia and Ozdaglar (2020))**.**

(i) If $\lambda_1 \le \lambda_2$ *then* $\mathsf{PE}(\lambda_1) \supseteq \mathsf{PE}(\lambda_2)$.

(ii) $\bigcap_{\lambda>0} \mathsf{PE}(\lambda) = \mathsf{NE}$.

(iii) $\bigcup_{\lambda>0} \mathsf{PE}(\lambda) = \mathsf{SE}$.

Proof. Item (i) is proved directly in Proposition 3 of Farnia and Ozdaglar (2020). For items (ii) and (iii), we have

$$\lim_{\lambda\nearrow\infty} \mathcal{V}_\lambda(\theta, \xi) = \mathcal{V}(\theta, \xi), \tag{23}$$

$$\lim_{\lambda\searrow 0} \mathcal{V}_\lambda(\theta, \xi) = \inf_{\zeta\in\Xi} \mathcal{V}(\theta, \zeta), \tag{24}$$

which, in conjunction with (i), prove (ii) and (iii) respectively. □

3 CTRL-PG: A Framework for Training Games

3.1 Formulating the Training Game

In this section, we introduce a new conceptual framework for training games within the Closed-Loop Transcription (CTRL) framework (Dai et al., 2022), which we call CTRL-PG (for "proximal games")[3]. Recall that we seek to learn encoder parameters $\theta^\star \in \Theta$ and decoder parameters $\xi^\star \in \Xi$ with the following desiderata:

- (*High-quality representations.*) The representation $\boldsymbol{z}(\theta^\star)$ has good geometric and statistical properties.
- (*Consistent autoencoding.*) The autoencoding $\hat{\boldsymbol{x}}(\theta^\star, \xi^\star)$ is close to $\boldsymbol{x}$ itself.

The exact qualities that we wish for our representations $\boldsymbol{z}(\theta)$ and the exact notion of closeness of $\hat{\boldsymbol{x}}(\theta, \xi)$ to $\boldsymbol{x}$ will change depending on the exact application or task which we want to learn. Thus, let us suppose for now that we generically quantify the representation *quality* by a function $\mathcal{Q}\colon \Theta \to \mathbb{R}_{\ge 0}$, such that *higher* values of $\mathcal{Q}$ indicate higher quality representations $\boldsymbol{z}(\theta)$. Also, let us quantify the *consistency* of the autoencoding as $\mathcal{C}\colon \Theta \times \Xi \to \mathbb{R}_{\ge 0}$, such that *lower* values of $\mathcal{C}$ indicate a more self-consistent autoencoding (i.e., $\hat{\boldsymbol{x}}(\theta, \xi)$ and $\boldsymbol{x}$ are close).

As discussed in Section 2.4, we seek to learn $f_\star$ and $g_\star$ as proximal equilibria for a two-player zero-sum game. Here we develop the value function $\mathcal{V}\colon \Theta \times \Xi$ of this game, which enables us to completely define the game. Recall that the encoder attempts to maximize $\mathcal{V}$, while the decoder attempts to minimize $\mathcal{V}$.

[3] In (Pai et al., 2022), a very similar formulation was labeled CTRL-SG (for "sequential games", and results were shown only for Stackelberg equilibria.

To ensure high-quality representations, the encoder should want to maximize $\mathcal{Q}(\theta)$ over $\theta \in \Theta$. On the other hand, to encourage a self-consistent autoencoding, the decoder should wish to minimize $\mathcal{C}(\theta, \xi)$ over $\xi \in \Xi$. In line with the CTRL framework, to encourage the decoder to be maximally powerful, the encoder should play a dual role as a GAN-type discriminator and thus should seek to distinguish the true data $\boldsymbol{x}$ from the autoencoding $\hat{\boldsymbol{x}}(\theta, \xi)$, say by *maximizing* $\mathcal{C}(\theta, \xi)$ over $\theta \in \Theta$. Thus, the encoder should seek to maximize $\mathcal{Q}(\theta) + \mathcal{C}(\theta, \xi)$ while the decoder should seek to minimize $\mathcal{C}(\theta, \xi)$. This yields the following game.

Definition 3.1 (CTRL-PG Game)**.** The CTRL-PG game is a two-player zero-sum game between:

- the encoder, playing $\theta \in \Theta$ to *maximize* the value function $\mathcal{V}$;
- the decoder, playing $\xi \in \Xi$ to *minimize* the value function $\mathcal{V}$;

where the value function $\mathcal{V} \colon \Theta \times \Xi \to \mathbb{R}$ has the form

$$\mathcal{V}(\theta, \xi) \doteq \mathcal{Q}(\theta) + \mathcal{C}(\theta, \xi) \tag{25}$$

for a given *quality* function $\mathcal{Q} \colon \Theta \to \mathbb{R}_{\geq 0}$ and *consistency* function $\mathcal{C} \colon \Theta \times \Xi \to \mathbb{R}_{\geq 0}$.

Thus in the CTRL-PG formulation, the encoder wishes to maximize $\mathcal{Q}(\theta) + \mathcal{C}(\theta, \xi)$, while the decoder wishes to minimize $\mathcal{Q}(\theta) + \mathcal{C}(\theta, \xi)$ or equivalently (since $\mathcal{Q}$ is not a function of ξ) just minimize $\mathcal{C}(\theta, \xi)$, which is perfectly in line with the above discussion.

This system generalizes the proposed setting of learning from a fixed finite dataset that we first discussed in Section 2.1. Since it is a purely game-theoretic formulation, in principle, one may adapt it to many learning contexts. In Section 4 we learn from a fixed finite (possibly labelled) dataset, but one could adapt this framework to e.g., semi-supervised learning and online/incremental learning.

3.2 Characterizing the Proximal Equilibria

In the following theorem, we compute the properties of the proximal equilibria of CTRL-PG games.

Theorem 3.2. *Suppose that the following assumptions hold:*

(i) (Quality can be maximized.) The set $\operatorname{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta)$ *is nonempty.*

(ii) (Consistency can be achieved.) For every $\theta \in \Theta$*, the set* $\operatorname{argmin}_{\xi \in \Xi} \mathcal{C}(\theta, \xi)$ *is nonempty.*

(iii) (The decoder can do equally well for any encoder play.) The function $\theta \mapsto \min_{\xi \in \Xi} \mathcal{C}(\theta, \xi)$ *is constant.*

Let $\lambda > 0$*, let* $\mathsf{PE}(\lambda)$ *be the set of* λ*-proximal equilibria for the CTRL-PG game, and let* $(\theta^\star, \xi^\star) \in \mathsf{PE}(\lambda)$*. Then:*

(a) (Quality is maximized.) We have $\theta^\star \in \operatorname{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta)$*.*

(b) (Consistency is achieved.) We have $\xi^\star \in \operatorname{argmin}_{\xi \in \Xi} \mathcal{C}(\theta^\star, \xi)$*.*

Proof. We use property (iii) from Proposition 2.7, i.e., $\mathsf{PE}(\lambda) \subseteq \mathsf{SE}$. Then we have

$$\begin{aligned}
\operatorname*{argmax}_{\theta \in \Theta} \inf_{\xi \in \Xi} \mathcal{V}(\theta, \xi) &= \operatorname*{argmax}_{\theta \in \Theta} \inf_{\xi \in \Xi} \{\mathcal{Q}(\theta) + \mathcal{C}(\theta, \xi)\} && (26)\\
&= \operatorname*{argmax}_{\theta \in \Theta} \left\{ \mathcal{Q}(\theta) + \inf_{\xi \in \Xi} \mathcal{C}(\theta, \xi) \right\} && (27)\\
&= \operatorname*{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta). && (28)\\
\operatorname*{argmin}_{\xi \in \Xi} \mathcal{V}(\theta^\star, \xi) &= \operatorname*{argmin}_{\xi \in \Xi} \{\mathcal{Q}(\theta^\star) + \mathcal{C}(\theta^\star, \xi)\} && (29)\\
&= \operatorname*{argmin}_{\xi \in \Xi} \mathcal{C}(\theta^\star, \xi). && (30)
\end{aligned}$$

Thus we have that $(\theta^\star, \xi^\star) \in \mathsf{PE}(\lambda)$ implies $(\theta^\star, \xi^\star) \in \mathsf{SE}$, which in turn implies

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta) \qquad \text{and} \qquad \xi^\star \in \operatorname*{argmin}_{\xi \in \Xi} \mathcal{C}(\theta^\star, \xi) \tag{31}$$

as desired. □

In the above result, the only condition which is conceptually interesting is condition (iii), which says that the function $\theta \mapsto \min_{\xi \in \Xi} \mathcal{C}(\theta, \xi)$ is constant. This says that for each choice of f_θ, there exists a g_ξ which inverts it on the support of the data distribution, at least in the sense of ensuring self-consistency as measured by $\mathcal{C}$. We argue that this property holds (at least approximately) for function classes used for autoencoders in practice (Kingma and Welling, 2014), especially within the closed-loop transcription framework (Dai et al., 2022, 2023), and so it is a reasonable assumption to make for our theory.

The key (albeit slightly technical) innovation of Theorem 3.2 is to show that the game-theoretic optimization procedure in multiple variables can be connected to a set of (possibly sequentially-solved) optimization problems in each variable. Thus CTRL-PG games are completely mathematically interpretable in the language of (conventional) optimization. This decomposition also opens the door to alternative implementations of the game, i.e., pretraining first an encoder (to maximize $\mathcal{Q}(\theta)$) then a decoder (to minimize $\mathcal{C}(\theta, \xi)$) before training them jointly, though we leave further exploration of this issue for future work in favor of developing an end-to-end optimization scheme in Section 5.

Thus, the general CTRL-PG game system allows us to use the CTRL framework for representation learning, choose principled objective functions to encourage the desired representation and autoencoding properties, and then explicitly characterize the optimal learned encoder and decoder for that algorithm. It also suggests principled optimization strategies and algorithms, such as the one we propose in Section 5, for obtaining these optimal functions.

4 Learning to Represent Structured Data

In this section, we discuss two applications of the CTRL-PG framework to concrete unsupervised and supervised representation learning problems. Our analysis of these problems may be viewed as verifications that the closed-loop transcription framework produces information-theoretically and geometrically desirable representations, in that the optimal solutions for simple cases agree with those found by classical theory. Thus, a large part of what we do in this section is to verify a connection and unification between classical and modern representation learning via the closed-loop transcription framework.

4.1 Unsupervised Learning for Data on a Low-Dimensional Subspace

Our first application of the CTRL-PG framework is to learn informative representations for data on a single subspace. Our framework is called CTRL-SSP (*single subspace pursuit*).

Our setting is that we have samples $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N \in \mathbb{R}^D$ which are $N \geq 1$ i.i.d. samples of some data random variable $\boldsymbol{x}$. To model that the data has statistical and geometric structure, suppose $\boldsymbol{x}$ is supported on a subspace $\mathcal{S}$ of dimension $d_{\mathcal{S}} \ll D$. We collect the samples into a sample matrix $\boldsymbol{X} = \left[\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\right] \in \mathbb{R}^{D \times N}$.

We have the following desiderata for CTRL-SSP, which result in useful representations and a self-consistent autoencoding.

1. The map $f_{\theta^\star}$ is injective on the support of the data distribution $\mathcal{S}$.
2. The maps $f_{\theta^\star}$ and $g_{\xi^\star}$ form an internally self-consistent closed-loop autoencoding on the support of the data distribution $\mathcal{S}$.

We now discuss how to quantify these desiderata. We look towards PCA, which has long been held as the de-facto standard for subspace learning. A common formulation of PCA is to find the best approximating subspace $\mathcal{S}_{\text{PCA}}$ of dimension d for the data. Suppose $f_{\text{PCA}} \colon \mathbb{R}^D \to \mathbb{R}^d$ projects its input $\boldsymbol{x}$ onto the coordinates of this subspace, and $g_{\text{PCA}} \colon \mathbb{R}^d \to \mathcal{S}_{\text{PCA}} \subseteq \mathbb{R}^D$ reconstructs its input $\boldsymbol{z}$ from its coordinates on the subspace. Now if $d \geq d_{\mathcal{S}}$, we have $\mathcal{S}_{\text{PCA}} \supseteq \mathcal{S}$, so by the earlier descriptions of f_{PCA} and g_{PCA}, we have $(g_{\text{PCA}} \circ f_{\text{PCA}}) \mid_{\mathcal{S}} = \text{id}_{\mathcal{S}}$. Thus the maps f_{PCA} and g_{PCA} are ℓ^2-isometries on $\mathcal{S}$ and $f_{\text{PCA}}(\mathcal{S})$ respectively, and $(g_{\text{PCA}} \circ f_{\text{PCA}})(\mathcal{S}) = \mathcal{S}$. Thus, we choose to preserve in CTRL-SSP exactly these essential properties of PCA:

1. f_{PCA} is an ℓ^2-isometry on $\mathcal{S}$.
2. $(f_{\text{PCA}} \circ g_{\text{PCA}} \circ f_{\text{PCA}})(\mathcal{S}) = f_{\text{PCA}}(\mathcal{S})$.

In particular, we desire that our encoder, more than just being injective on $\mathcal{S}$, must preserve all distances and thus all the structure of the data distribution on $\mathcal{S}$. We also desire our decoder to learn the linear structure of the representation space imposed by the encoder, so we desire subspace-level autoencoding consistency in the representation space.

As per the CTRL-PG formulation, we want to encode each of the desiderata in our choices of Θ, Ξ, $\mathcal{Q}$ and $\mathcal{C}$. Because the maps in PCA are orthogonal, we choose Θ and Ξ to represent full sets of orthogonal maps, i.e., we choose $f_\theta \colon \boldsymbol{x} \mapsto \theta \boldsymbol{x}$ and $g_\xi \colon \boldsymbol{z} \mapsto \xi \boldsymbol{z}$ where $\theta \in \Theta = \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)$ and $\xi \in \Xi = \mathsf{O}(\mathbb{R}^d, \mathbb{R}^D)$.

Before we continue with more intuition-building, let us make a few sensible assumptions that reduce the complexity of the problem. These assumptions will be re-used, so we store them for shorthand reference later.

Assumption 4.1.

(i) (Informative data.) $\mathrm{Col}(\boldsymbol{X}) = \mathcal{S}$.

(ii) (Large enough representation space.) $d \geq d_{\mathcal{S}}$.

To choose $\mathcal{Q}$, we are motivated by the following lemma.

Lemma 4.2. *Suppose that Assumption 4.1 holds, and that*

$$f_\star \in \operatorname*{argmax}_{f \in \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)} R(f(\boldsymbol{X})). \tag{32}$$

Then $f_\star(\mathcal{S})$ is a subspace of dimension $d_{\mathcal{S}}$, and $f_\star$ is an ℓ^2-isometry on $\mathcal{S}$.

Proof. Since $f_\star$ is a linear map and $\mathcal{S}$ is a linear subspace, $f_\star(\mathcal{S})$ is a linear subspace, and furthermore $\dim(f_\star(\mathcal{S})) \leq d_{\mathcal{S}}$. We now claim that $f_\star$ is an ℓ^2-isometry on $\mathcal{S}$. We show this by calculating an upper bound for $R(f(\boldsymbol{X}))$ and show that it is achieved if and only if f is an ℓ^2-isometry on $\mathcal{S}$.

Indeed, for any $f \in \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)$, we have that f has operator norm and Lipschitz constant equal to unity, so $\|f(\boldsymbol{x})\|_{\ell^2} \leq \|\boldsymbol{x}\|_{\ell^2}$ for any $\boldsymbol{x} \in \mathbb{R}^D$. By the Courant-Fischer min-max theorem for singular values, we have, for each $1 \leq p \leq d_{\mathcal{S}}$, that

$$\sigma_p(f(\boldsymbol{X})) = \sup_{S \in \mathsf{Gr}(p, \mathbb{R}^D)} \inf_{\substack{\boldsymbol{u} \in S \\ \|\boldsymbol{u}\|_{\ell^2} = 1}} \|f(\boldsymbol{X}) \cdot \boldsymbol{u}\|_{\ell^2} \tag{33}$$

$$\leq \sup_{S \in \mathsf{Gr}(p, \mathbb{R}^D)} \inf_{\substack{\boldsymbol{u} \in S \\ \|\boldsymbol{u}\|_{\ell^2} = 1}} \|\boldsymbol{X} \cdot \boldsymbol{u}\|_{\ell^2} = \sigma_p(\boldsymbol{X}). \tag{34}$$

Thus

$$R(f(\boldsymbol{X})) = \frac{1}{2} \operatorname{logdet}\left(\boldsymbol{I}_d + \frac{d}{N\varepsilon^2} f(\boldsymbol{X}) f(\boldsymbol{X})^* \right) \tag{35}$$

$$= \frac{1}{2} \sum_{p=1}^{d_{\mathcal{S}}} \log\left(1 + \frac{d}{N\varepsilon^2} \sigma_p(f(\boldsymbol{X}))^2 \right) \leq \frac{1}{2} \sum_{p=1}^{d_{\mathcal{S}}} \log\left(1 + \frac{d}{N\varepsilon^2} \sigma_p(\boldsymbol{X})^2 \right). \tag{36}$$

As such, f is an ℓ^2 isometry on $\mathcal{S}$ if and only if $\sigma_p(f(\boldsymbol{X})) = \sigma_p(\boldsymbol{X})$ for all $1 \leq p \leq d_{\mathcal{S}}$.

Thus, any $f_\star \in \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)$ which fulfills the upper bound for $R(f(\boldsymbol{X}))$, i.e., any maximizer for $R(f(\boldsymbol{X}))$, is an ℓ^2 isometry on $\mathcal{S}$. Therefore, $\dim(f_\star(\mathcal{S})) = d_{\mathcal{S}}$. □

This lemma suggests that maximizing $R(\boldsymbol{Z}(\theta))$ will achieve the desired encoder property, i.e., $f_{\theta^\star} \mid_{\mathcal{S}}$ is an ℓ^2-isometry. To choose $\mathcal{C}$, we are motivated by another lemma.

Lemma 4.3. *Suppose that Assumption 4.1 holds. Let $f \in \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)$, and suppose that*

$$\widetilde{g} \in \operatorname*{argmin}_{g \in \mathsf{O}(\mathbb{R}^d, \mathbb{R}^D)} \Delta R(f(\boldsymbol{X}), (f \circ g \circ f)(\boldsymbol{X})). \tag{37}$$

Then $f(\mathcal{S}) = (f \circ g \circ f)(\mathcal{S})$.

Proof. First, note that by taking $g = f^*$, we obtain that $f(\boldsymbol{X}) = (f \circ g \circ f)(\boldsymbol{X})$, and so

$$\min_{g \in \mathsf{O}(\mathbb{R}^d, \mathbb{R}^D)} \Delta R(f(\boldsymbol{X}), (f \circ g \circ f)(\boldsymbol{X})) = 0. \tag{38}$$

Thus if

$$\widetilde{g} \in \operatorname*{argmin}_{g \in \mathsf{O}(\mathbb{R}^d, \mathbb{R}^D)} \Delta R(f(\boldsymbol{X}), (f \circ g \circ f)(\boldsymbol{X})) \tag{39}$$

then we have

$$\Delta R(f(\boldsymbol{X}), (f \circ \widetilde{g} \circ f)(\boldsymbol{X})) = 0 \tag{40}$$

so that, by Proposition 2.1, we have

$$f(\mathcal{S}) = \operatorname{Col}(f(\boldsymbol{X})) = \operatorname{Col}((f \circ \widetilde{g} \circ f)(\boldsymbol{X})) = (f \circ \widetilde{g} \circ f)(\mathcal{S}) \tag{41}$$

as desired. □

This lemma suggests that minimizing $\Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi))$ will achieve the desired decoder property, i.e., $f_{\theta^\star}(\mathcal{S}) = (f_{\theta^\star} \circ g_{\xi^\star} \circ f_{\theta^\star})(\mathcal{S})$. With these motivations in mind, we apply the CTRL-PG formula to construct an appropriate value function, thus establishing the CTRL-SSP game.

Definition 4.4 (CTRL-SSP Game)**.** The CTRL-SSP game is a two-player zero-sum game between:

- the encoder, playing $\theta \in \Theta = \mathsf{O}(\mathbb{R}^D, \mathbb{R}^d)$ to *maximize* the value function $\mathcal{V}$;
- the decoder, playing $\xi \in \Xi = \mathsf{O}(\mathbb{R}^d, \mathbb{R}^D)$ to *minimize* the value function $\mathcal{V}$;

where the value function $\mathcal{V} \colon \Theta \times \Xi \to \mathbb{R}$ has the form

$$\mathcal{V}(\theta, \xi) \doteq R(\boldsymbol{Z}(\theta)) + \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi)) \tag{42}$$

for the linear function parameterizations $f_\theta \colon \boldsymbol{x} \mapsto \theta \boldsymbol{x}$ and $g_\xi \colon \boldsymbol{z} \mapsto \xi \boldsymbol{z}$.

We now explicitly characterize the proximal equilibria of CTRL-SSP games.

Theorem 4.5 (Proximal Equilibria of CTRL-SSP Game)**.** *Suppose that Assumption 4.1 holds. Let $\lambda > 0$ and suppose that $(\theta^\star, \xi^\star)$ is a λ-proximal equilibrium of the CTRL-SSP game. Then:*

(a) (Injective encoder.) $f_{\theta^\star}(\mathcal{S})$ is a linear subspace of dimension $d_{\mathcal{S}}$, and $f_{\theta^\star}$ is an ℓ^2-isometry on $\mathcal{S}$.

(b) (Consistent autoencoding.) $f_{\theta^\star}(\mathcal{S}) = (f_{\theta^\star} \circ g_{\xi^\star} \circ f_{\theta^\star})(\mathcal{S})$.

Proof. We attempt to invoke Theorem 3.2. Since $\theta \mapsto \boldsymbol{Z}(\theta)$ is a continuous mapping, and the mapping $\boldsymbol{Z} \mapsto R(\boldsymbol{Z})$ is a continuous mapping, the composition $\theta \mapsto R(\boldsymbol{Z}(\theta))$ is a continuous mapping over the compact domain Θ. By the extreme value theorem, $\operatorname{argmax}_{\theta \in \Theta} R(\boldsymbol{Z}(\theta)) = \operatorname{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta)$ is nonempty. Now for any θ one can take $\xi = \theta^*$, so that

$$\Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi)) = \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \theta^*)) = \Delta R(\boldsymbol{Z}(\theta), \boldsymbol{Z}(\theta)) = 0. \tag{43}$$

Since by Proposition 2.1 we have $\Delta R(\cdot, \cdot) \geq 0$, we have that $\operatorname{argmin}_{\xi \in \Xi} \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi)) = \operatorname{argmin}_{\xi \in \Xi} \mathcal{C}(\theta, \xi)$ is nonempty (since it contains $\xi = \theta^*$) and for every θ we have $\min_{\xi \in \Xi} \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi)) = \min_{\xi \in \Xi} \mathcal{C}(\theta, \xi) = 0$. Thus, all assumptions of Theorem 3.2 hold. Applying the theorem, we have

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \Theta} \mathcal{Q}(\theta) \in \operatorname*{argmax}_{\theta \in \Theta} R(Z(\theta)) \tag{44}$$

$$\xi^\star \in \operatorname*{argmin}_{\xi \in \Xi} \mathcal{C}(\theta^\star, \xi) = \operatorname*{argmin}_{\xi \in \Xi} \Delta R(\boldsymbol{Z}(\theta^\star), \hat{\boldsymbol{Z}}(\theta^\star, \xi)). \tag{45}$$

Then the theorem is proved using the characterizations of the optimizers in Lemmas 4.2 and 4.3. □

CTRL-SSP replicates the essential isometry aspect of the PCA solution, but it learns all principal components simultaneously, unlike the common greedy algorithms. Thus, it does not require any model selection beyond a choice of d, which can be set to *any* integer greater than $d_{\mathcal{S}}$ without loss of efficacy. This is one concrete benefit of the "deep learning inspired" continuous optimization framework that we use here.

4.2 Supervised Learning for Data on Multiple Low-Dimensional Subspaces

Our second application of the CTRL-PG framework is to learn informative representations for data on multiple non-intersecting subspaces. Our framework is called CTRL-MSP (*multi subspace pursuit*).

Our setting is as follows. We have samples $\boldsymbol{x}^1, \dots, \boldsymbol{x}^N \in \mathbb{R}^D$ which are $N \geq 1$ i.i.d. samples of some data random variable $\boldsymbol{x}$. Here $\boldsymbol{x}$ obeys a mixture distribution $\boldsymbol{\pi} \in [0,1]^K$ such that $\boldsymbol{x} = \boldsymbol{x}_j$ with probability π_j, where each $\boldsymbol{x}_j$ is a random variable taking values in $\mathbb{R}^D$, and the $\boldsymbol{x}_j$ are independent. To model the statistical and geometric structure of the data, we assume that each $\boldsymbol{x}_j$ is supported on a subspace $\mathcal{S}_j$ of dimension $d_j \ll D$. We collect the samples associated with each $\boldsymbol{x}_j$ into a matrix $\boldsymbol{X}_j \in \mathbb{R}^{D \times N_j}$. Let $\boldsymbol{y} \in [K]^N$ be a vector encoding the label of each sample $\boldsymbol{x}^i$, which is really an assignment of each sample $\boldsymbol{x}^i$ to a random variable $\boldsymbol{x}_j$. Finally, we collect all $N \doteq \sum_{j=1}^K N_j$ samples into a sample matrix $\boldsymbol{X} \doteq \left[\boldsymbol{x}^1, \dots, \boldsymbol{x}^N\right] \in \mathbb{R}^{D \times N}$.

We have the following desiderata for CTRL-MSP, which result in useful representations and a self-consistent autoencoding.

1. The map $f_{\theta^\star}$ is injective on the support of the data distribution $\bigcup_{j=1}^K \mathcal{S}_j$.
2. The map $f_{\theta^\star}$ is discriminative between the supports of each data distribution $\mathcal{S}_j$.
3. The maps $f_{\theta^\star}$ and $g_{\xi^\star}$ form an internally self-consistent closed-loop autoencoding on the support of the data distribution $\bigcup_{j=1}^K \mathcal{S}_j$.

Unlike with CTRL-SSP, there is no classical theory which purports to exactly solve this problem — the closest probably being GPCA (Vidal et al., 2003) — so we cannot lift quantifications for our desiderata from there. Instead, we present a new set of ways to quantify our desiderata.

- To enforce the *injectivity* of the encoder, we aim to ensure that each $f_{\theta^\star}(\mathcal{S}_j)$ is a linear subspace of dimension equal to that of $\mathcal{S}_j$, and furthermore, we aim to enforce that the covariance matrix of each $\boldsymbol{Z}_j(\theta^\star)$ should have no small nonzero singular values. The first property means that the encoder is mathematically injective, i.e., $f_{\theta^\star}$ does not map two points to the same representation. The second property means that the representations $\boldsymbol{Z}_j(\theta^\star)$ are spread out across all directions of the subspace, thus ensuring that $f_{\theta^\star}$ does not map two *distant* points in the same subspace to *close* representations, ensuring *well-behaved* (i.e., not pathological) injectivity.
- To enforce the *discriminativeness* of the encoder, we aim to ensure that the $f_{\theta^\star}(\mathcal{S}_j)$ are pairwise orthogonal subspaces. This property means that the $f_{\theta^\star}(\mathcal{S}_j)$ are statistically incoherent, ensuring that a given sample $\boldsymbol{x}^i$ can be cleanly assigned to one of the subspaces $\mathcal{S}_j$ based on the statistical correlations between its representation $\boldsymbol{z}^i(\theta^\star)$ and vectors from each representation subspace $f_{\theta^\star}(\mathcal{S}_j)$.
- To enforce internal *self-consistency of the closed-loop autoencoding*, we aim to have $f_\star(\mathcal{S}_j) = (f_\star \circ g_\star \circ f_\star)(\mathcal{S}_j)$ for each $j \in [K]$. This property means that the decoder has accurately learned the linear structure of the representation space induced by the encoder.

As per the CTRL-PG formulation, we want to encode each of the desiderata in our choices of Θ, Ξ, $\mathcal{Q}$, and $\mathcal{C}$. Because we wish to provide discriminative representations for multiple subspaces, orthogonal maps no longer suffice, though we are fine with linear maps, i.e., we choose $f_\theta \colon \boldsymbol{x} \mapsto \theta \boldsymbol{x}$ and $g_\xi \colon \boldsymbol{z} \mapsto \xi \boldsymbol{z}$ where $\Theta \subseteq \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d)$ and $\Xi \subseteq \mathsf{L}(\mathbb{R}^d, \mathbb{R}^D)$.

Similar to the case of CTRL-SSP, let us first lay out a few simplifying assumptions.

Assumption 4.6.

(i) (Multiple classes.) $K \geq 2$.

(ii) (Informative data.) For each $j \in [K]$*, we have* $\operatorname{Col}(\boldsymbol{X}_j) = \mathcal{S}_j$.

(iii) (Large enough representation space.) $\sum_{j=1}^K d_j \leq d$.

(iv) (Incoherent class data.) $\sum_{j=1}^K d_j = \dim(\sum_{j=1}^K \mathcal{S}_j)$.[4]

[4] An intuitive understanding of this condition is that if we take a linearly independent set from each $\mathcal{S}_j$, the union of all these sets is still linearly independent.

(v) (High coding precision.) $\varepsilon^4 \leq \min_{j=1}^K (N_j/N \cdot d^2/d_j^2)$.

To find the precise desired set of Θ, recall that Proposition 2.2 shows that maximizing $\Delta R(\boldsymbol{Z} \mid \boldsymbol{y})$ over $\boldsymbol{Z}$ subject to normalization on the $\boldsymbol{Z}_j$ provides the first two desiderata: in particular we have that, at optimum, $\operatorname{rank}(\boldsymbol{Z}_j) = d_j$, each $\boldsymbol{Z}_j$ has d_j large singular values where at least $d_j - 1$ of them are equal, and the spans of the columns of the $\boldsymbol{Z}_j$ form orthogonal subspaces. This motivates that our encoder should maximize $\Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{y})$ over $\theta \in \Theta$, where Θ denotes an appropriate set of functions with normalization constraints. Indeed, the following lemma provides one choice of Θ and characterizes the optima obtained by maximizing $\Delta R(Z(\theta) \mid \boldsymbol{y})$ over $\theta \in \Theta$.

Lemma 4.7. *Suppose that Assumption 4.6 holds, and let*

$$\theta^\star \in \operatorname*{argmax}_{\substack{\theta \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d) \\ \|\boldsymbol{Z}_j(\theta)\|_F^2 \leq N_j \\ \forall j \in [K]}} \Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{y}). \tag{46}$$

Then:

(a) (Injective encoder.) For each $j \in [K]$, we have that $f_{\theta^\star}(\mathcal{S}_j)$ is a linear subspace of dimension d_j. Further, for each $j \in [K]$, exactly one of the following holds:

i. $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \sigma_2(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) = \frac{N_j}{d_j}$*; or*

ii. $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \sigma_2(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j-1}(\boldsymbol{Z}_j(\theta^\star)) \in (\frac{N_j}{d_j}, \frac{N_j}{d_j - 1})$ *and* $\sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) > 0$*, where if* $d_j = 1$ *then* $\frac{N_j}{d_j - 1}$ *is interpreted as* $+\infty$.

(b) (Discriminative encoder.) The subspaces $\{f_{\theta^\star}(\mathcal{S}_j)\}_{j=1}^K$ are orthogonal.

Proof. First, since $f_{\theta^\star}$ is linear, $f_{\theta^\star}(\mathcal{S}_j)$ is a linear subspace; further, $\dim(f_{\theta^\star}(\mathcal{S}_j)) \leq d_j$. We now claim that the subspaces $\{f_{\theta^\star}(\mathcal{S}_j)\}_{j=1}^K$ are orthogonal. Since $f_{\theta^\star}(\mathcal{S}_j) = \operatorname{Col}(\boldsymbol{Z}_j(\theta^\star))$, this is equivalent to the columns of $\boldsymbol{Z}_j(\theta^\star)$ being orthogonal to the columns of $\boldsymbol{Z}_\ell(\theta^\star)$ for all $\ell \neq j$, i.e., $\boldsymbol{Z}_j(\theta^\star)^* \boldsymbol{Z}_\ell(\theta^\star) = \boldsymbol{0}$.

The essential tool we use to show that the $\boldsymbol{Z}_j(\theta^\star)$ have orthogonal columns is (Yu et al., 2020, Lemma A.5), which states that, for matrices $\boldsymbol{Z}_j \in \mathbb{R}^{d \times N_j}$ which are collected in a matrix $\boldsymbol{Z} \in \mathbb{R}^{d \times N}$, we have

$$\Delta R(\boldsymbol{Z} \mid \boldsymbol{y}) \leq \frac{1}{2N} \sum_{j=1}^K \sum_{p=1}^{d_j} \log \left(\frac{\left(1 + \frac{d}{N\varepsilon^2} \sigma_p(\boldsymbol{Z}_j)^2\right)^N}{\left(1 + \frac{d}{N_j \varepsilon^2} \sigma_p(\boldsymbol{Z}_j)^2\right)^{N_j}} \right) \tag{47}$$

with equality if and only if $\boldsymbol{Z}_j^* \boldsymbol{Z}_\ell = \boldsymbol{0}$ for all $1 \leq j < \ell \leq K$.

Suppose for the sake of contradiction that $\boldsymbol{Z}_j(\theta^\star)^* \boldsymbol{Z}_\ell(\theta^\star) \neq \boldsymbol{0}$ for some $1 \leq j < \ell \leq K$. Since $d \geq \sum_{j=1}^K d_j$ and the subspaces $\mathcal{S}_j$ for $j = 1, \ldots, K$ have linearly independent bases, one can construct via the SVD another $\widetilde{\theta} \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d)$ such that

- $\sigma_p(\boldsymbol{Z}_j(\theta^\star)) = \sigma_p(\boldsymbol{Z}_j(\widetilde{\theta}))$, for $1 \leq p \leq d_j$ and $1 \leq j \leq K$.
- $\boldsymbol{Z}_j(\widetilde{\theta})^* \boldsymbol{Z}_\ell(\widetilde{\theta}) = \boldsymbol{0}$ for all for $1 \leq j < \ell \leq K$.

Then for each $j \in [K]$ we have

$$\left\| \boldsymbol{Z}_j(\widetilde{\theta}) \right\|_F^2 = \sum_{p=1}^{d_j} \sigma_p^2(\boldsymbol{Z}_j(\widetilde{\theta})) = \sum_{p=1}^{d_j} \sigma_p^2(\boldsymbol{Z}_j(\theta^\star)) = \|\boldsymbol{Z}_j(\theta^\star)\|_F^2 \leq N_j \tag{48}$$

so we have that $\widetilde{\theta}$ is feasible for the problem. Further, since equality holds in the inequality (47) of (Yu et al., 2020, Lemma A.5) for $\widetilde{\theta}$ but not for $\theta^\star$, we have

$$\Delta R(\boldsymbol{Z}(\widetilde{\theta}) \mid \boldsymbol{y}) = \frac{1}{2N}\sum_{j=1}^{K}\sum_{p=1}^{d_j}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\widetilde{\theta}))^2\right)^N}{\left(1+\frac{d}{N_j\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\widetilde{\theta}))^2\right)^{N_j}}\right) \tag{49}$$

$$= \frac{1}{2N}\sum_{j=1}^{K}\sum_{p=1}^{d_j}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^N}{\left(1+\frac{d}{N_j\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^{N_j}}\right) \tag{50}$$

$$> \Delta R(\boldsymbol{Z}(\theta^\star) \mid \boldsymbol{y}). \tag{51}$$

Thus $\theta^\star$ does not maximize $\theta \mapsto \Delta R(Z(\theta) \mid \boldsymbol{y})$ over feasible θ, a contradiction. Thus, we must have that $\boldsymbol{Z}_j(\theta^\star)^*\boldsymbol{Z}_\ell(\theta^\star) = \boldsymbol{0}$ for all $1 \le j < \ell \le K$, and so the $\{f_{\theta^\star}(\mathcal{S}_j)\}_{j=1}^K$ are orthogonal subspaces.

Now, we claim that either $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) = \frac{N_j}{d_j}$, or $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j-1}(\boldsymbol{Z}_j(\theta^\star)) \in (\frac{N_j}{d_j}, \frac{N_j}{d_j-1})$ and $\sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) > 0$. To show this, the general approach is to isolate the effect of $f_{\theta^\star}$ on each $\boldsymbol{X}_j$. In particular, fix $t \in [K]$. We claim that

$$\theta^\star \in \operatorname*{argmax}_{\theta \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d)} \sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta))^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta))^2\right)^{N_t}}\right). \tag{52}$$

Indeed, suppose that this does not hold, and there exists a feasible $\widehat{\theta}$ such that

$$\sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta^\star))^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta^\star))^2\right)^{N_t}}\right) < \sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widehat{\theta}))^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widehat{\theta}))^2\right)^{N_t}}\right). \tag{53}$$

Then, again since $d \ge \sum_{j=1}^K d_j$ and the subspaces $\mathcal{S}_j$ have linearly independent bases, one can construct another $\widetilde{\theta}$ such that

- $\sigma_p(\boldsymbol{Z}_t(\widetilde{\theta})) = \sigma_p(\boldsymbol{Z}_t(\widehat{\theta}))$, for $1 \le p \le d_t$.
- $\sigma_p(\boldsymbol{Z}_j(\widetilde{\theta})) = \sigma_p(\boldsymbol{Z}_j(\theta^\star))$, for $1 \le p \le d_j$, $1 \le j \le K$ with $j \ne t$.
- $\boldsymbol{Z}_j(\widetilde{\theta})^*\boldsymbol{Z}_\ell(\widetilde{\theta}) = \boldsymbol{0}$ for $1 \le j < \ell \le K$.

For the same reason as in the previous claim, $\widetilde{\theta}$ is feasible. Moreover, $\Delta R(\boldsymbol{Z}(\widetilde{\theta})) \mid \boldsymbol{\Pi}) > \Delta R(\boldsymbol{Z}(\theta^\star) \mid \boldsymbol{\Pi})$, because

$$2n \cdot \Delta R(\boldsymbol{Z}(\widetilde{\theta})) \mid \boldsymbol{\Pi}) \tag{54}$$

$$= \sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widetilde{\theta})^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widetilde{\theta})^2\right)^{N_t}}\right) + \sum_{\substack{j=1\\j\ne t}}^{K}\sum_{p=1}^{d_j}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\widetilde{\theta}))^2\right)^N}{\left(1+\frac{d}{N_j\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\widetilde{\theta}))^2\right)^{N_j}}\right) \tag{55}$$

$$= \sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widehat{\theta}))^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\widehat{\theta}))^2\right)^{N_t}}\right) + \sum_{\substack{j=1\\j\ne t}}^{K}\sum_{p=1}^{d_j}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^N}{\left(1+\frac{d}{N_j\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^{N_j}}\right). \tag{56}$$

This is strictly lower bounded by

$$\sum_{p=1}^{d_t}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta^\star))^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p(\boldsymbol{Z}_t(\theta^\star))^2\right)^{N_t}}\right) + \sum_{\substack{j=1\\j\ne t}}^{K}\sum_{p=1}^{d_j}\log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^N}{\left(1+\frac{d}{N_j\varepsilon^2}\sigma_p(\boldsymbol{Z}_j(\theta^\star))^2\right)^{N_j}}\right) \tag{57}$$

$$= 2n \cdot \Delta R(\boldsymbol{Z}(\theta^\star) \mid \boldsymbol{\Pi}). \tag{58}$$

Thus $\theta_\star$ is not a maximizer of $\theta \mapsto \Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{\Pi})$, which is a contradiction. Hence, (52) follows.

To finish, we may now solve the problem in terms of the singular values of $\boldsymbol{Z}_t(\theta)$. Indeed, from the above optimization problem and the definition of the feasible set, the singular values $\sigma_p(\boldsymbol{Z}_t(\theta^\star))$ are the solutions of the scalar optimization problem

$$\max_{\sigma_1,\dots,\sigma_{d_t}\in\mathbb{R}} \quad \sum_{p=1}^{d_t} \log\left(\frac{\left(1+\frac{d}{N\varepsilon^2}\sigma_p^2\right)^N}{\left(1+\frac{d}{N_t\varepsilon^2}\sigma_p^2\right)^{N_t}}\right) \tag{59}$$

$$\text{s.t.} \quad \sigma_1 \geq \cdots \geq \sigma_{d_t} \geq 0, \tag{60}$$

$$\sum_{p=1}^{d_t} \sigma_p^2 = N_j. \tag{61}$$

Given the assumption that ε is small enough, (Yu et al., 2020, Lemma A.7) says that the solutions to this optimization problem either fulfill $\sigma_1 = \cdots = \sigma_{d_t} = \frac{N_j}{d_t}$ or $\sigma_1 = \cdots = \sigma_{d_t-1} \in (\frac{N_t}{d_t}, \frac{N_t}{d_t-1})$ and $\sigma_{d_t} > 0$ as desired, where if $d_t = 1$ then $\frac{N_t}{d_t-1}$ is interpreted as $+\infty$. This also confirms that $\dim(f_\star(\mathcal{S}_t)) = \operatorname{rank}(\boldsymbol{Z}_t(\theta^\star)) = d_t$, so the proof is complete. □

Thus, we should take $\Theta = \{\theta \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d) \colon \|\boldsymbol{Z}_j(\theta)\|_F^2 \leq N_j, \forall j \in [K]\}$ and $\mathcal{Q}(\theta) = \Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{y})$. Notice how none of this argument fundamentally relies on θ being a matrix. With some adjustments it could be considered as the parameter set for a sufficiently expressive deep neural network.

Regarding the third desideratum, the following lemma motivates choosing $\mathcal{C}(\theta, \xi) = \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi))$.

Lemma 4.8. *Suppose that Assumption 4.6 holds. Let $f \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d)$ be such that $\|f(\boldsymbol{X}_j)\|_F^2 \leq N_j$ for each j, and suppose that*

$$\widetilde{g} \in \operatorname*{argmin}_{g\in\mathsf{L}(\mathbb{R}^d,\mathbb{R}^D)} \sum_{j=1}^{K} \Delta R(f(\boldsymbol{X}_j), (f\circ g\circ f)(\boldsymbol{X}_j)). \tag{62}$$

Then $f(\mathcal{S}_j) = (f \circ g \circ f)(\mathcal{S}_j)$ for each $j \in [K]$.

Proof. First, note that by taking $g = f^\dagger$, we obtain that $f(\boldsymbol{X}) = (f \circ g \circ f)(\boldsymbol{X})$, and so

$$\min_{g\in\mathsf{L}(\mathbb{R}^d,\mathbb{R}^D)} \Delta R(f(\boldsymbol{X}_j), (f\circ g\circ f)(\boldsymbol{X}_j)) = 0. \tag{63}$$

Thus if

$$\widetilde{g} \in \operatorname*{argmin}_{g\in\mathsf{L}(\mathbb{R}^d,\mathbb{R}^D)} \sum_{j=1}^{K} \Delta R(f(\boldsymbol{X}_j), (f\circ g\circ f)(\boldsymbol{X}_j)) \tag{64}$$

then we have

$$\Delta R(f(\boldsymbol{X}_j), (f\circ \widetilde{g}\circ f)(\boldsymbol{X}_j)) = 0, \qquad \forall j \in [K] \tag{65}$$

so that, by Proposition 2.1, we have

$$f(\mathcal{S}_j) = \operatorname{Col}(f(\boldsymbol{X}_j)) = \operatorname{Col}((f\circ \widetilde{g}\circ f)(\boldsymbol{X}_j)) = (f\circ \widetilde{g}\circ f)(\mathcal{S}_j), \qquad \forall j \in [K] \tag{66}$$

as desired. □

This lemma suggests that minimizing $\sum_{j=1}^K \Delta R(\boldsymbol{Z}(\theta), \hat{\boldsymbol{Z}}(\theta, \xi))$ over all $\xi \in \mathsf{L}(\mathbb{R}^d, \mathbb{R}^D)$ will achieve the desired decoder property, i.e., $f(\mathcal{S}_j) = (f \circ \widetilde{g} \circ f)(\mathcal{S}_j)$ for each $j \in [K]$. With these motivations in mind, we apply the CTRL-PG formula to construct an appropriate value function, establishing the CTRL-MSP game.

Definition 4.9 (CTRL-MSP Game)**.** The CTRL-MSP game is a two-player zero-sum game between:

- the encoder, playing $\theta \in \Theta = \{\theta \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d) \mid \|\boldsymbol{Z}_j(\theta)\|_F^2 \leq N_j \ \forall j \in [K]\}$ to *maximize* the value function $\mathcal{V}$;

- the decoder, playing $\xi \in \Xi = \mathsf{L}(\mathbb{R}^d, \mathbb{R}^D)$ to *minimize* the value function $\mathcal{V}$;

where the value function $\mathcal{V}\colon \Theta \times \Xi \to \mathbb{R}$ has the form

$$\mathcal{V}(\theta, \xi) \doteq \Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{y}) + \sum_{j=1}^{K} \Delta R(\boldsymbol{Z}_j(\theta), \hat{\boldsymbol{Z}}_j(\theta, \xi)) \tag{67}$$

for the linear function parameterizations $f_\theta\colon \boldsymbol{x} \mapsto \theta \boldsymbol{x}$ and $g_\xi\colon \boldsymbol{z} \mapsto \xi \boldsymbol{z}$.

We now characterize the proximal equilibria of the CTRL-MSP game.

Theorem 4.10 (Proximal Equilibria of CTRL-MSP Game)**.** *Suppose that Assumption 4.6 holds. Let $\lambda > 0$ and suppose that $(\theta^\star, \xi^\star)$ is a λ-proximal equilibrium of the CTRL-MSP game. Then:*

(a) (Injective encoder.) For each $j \in [K]$, we have that $f_{\theta^\star}(\mathcal{S}_j)$ is a linear subspace of dimension d_j. Further, for each $j \in [K]$, exactly one of the following holds:

i. $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \sigma_2(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) = \frac{N_j}{d_j}$; or

ii. $\sigma_1(\boldsymbol{Z}_j(\theta^\star)) = \sigma_2(\boldsymbol{Z}_j(\theta^\star)) = \cdots = \sigma_{d_j-1}(\boldsymbol{Z}_j(\theta^\star)) \in (\frac{N_j}{d_j}, \frac{N_j}{d_j-1})$ and $\sigma_{d_j}(\boldsymbol{Z}_j(\theta^\star)) > 0$, where if $d_j = 1$ then $\frac{N_j}{d_j-1}$ is interpreted as $+\infty$.

(b) (Discriminative encoder.) The subspaces $\{f_{\theta^\star}(\mathcal{S}_j)\}_{j=1}^K$ are orthogonal.

(c) (Consistent autoencoding.) $f_{\theta^\star}(\mathcal{S}_j) = (f_{\theta^\star} \circ g_{\theta^\star} \circ f_{\theta^\star})(\mathcal{S}_j)$ for each $j \in [K]$.

Proof. We attempt to invoke Theorem 3.2. First, we claim that $\operatorname{argmax}_{\theta\in\Theta} \mathcal{Q}(\theta) = \operatorname{argmax}_{\theta\in\Theta} \Delta R(\boldsymbol{Z}($ *theta*$) \mid \boldsymbol{y})$ is nonempty. Let $\mathcal{S} \doteq \mathsf{Span}(\bigcup_{j=1}^k \mathcal{S}_j)$. While $\mathcal{Q}$ is continuous in θ, compactness (required for the usual argument showing the existence of maxima) is not immediate from the definition: linear maps in Θ are controlled only on $\mathcal{S}$ and may have arbitrarily large operator norms on $\mathcal{S}^\perp$, thus making Θ an unbounded set and not compact. To remedy this, consider the related problem of optimization over the set

$$\Theta' \doteq \Theta \cap \left\{\theta \in \mathsf{L}(\mathbb{R}^D, \mathbb{R}^d) \middle| \, f_\theta(\mathcal{S}^\perp) = \{\boldsymbol{0}\}\right\}.$$

Now we have

$$\max_{\theta\in\Theta} \mathcal{Q}(\theta) = \max_{\theta\in\Theta'} \mathcal{Q}(\theta) \qquad \text{and} \qquad \operatorname*{argmax}_{\theta\in\Theta'} \mathcal{Q}(\theta) \subseteq \operatorname*{argmax}_{\theta\in\Theta} \mathcal{Q}(\theta).$$

Thus, it suffices to show that $\operatorname{argmax}_{\theta\in\Theta'} \mathcal{Q}(\theta)$ is nonempty. Clearly Θ' is compact. The extreme value theorem holds for optimizing $\mathcal{Q}$ over Θ', and the claim is proved.

Now we claim that $\operatorname{argmin}_{\xi\in\Xi} \mathcal{C}(\theta, \xi) = \operatorname{argmin}_{\xi\in\Xi} \sum_{j=1}^K \Delta R(\boldsymbol{Z}_j(\theta), \hat{\boldsymbol{Z}}_j(\theta, \xi))$ exists for every θ. Indeed, Proposition 2.1 shows that $\Delta R(\boldsymbol{Z}_1, \boldsymbol{Z}_2) \geq 0$ with equality if and only if $\boldsymbol{Z}_1 \boldsymbol{Z}_1^* = \boldsymbol{Z}_2 \boldsymbol{Z}_2^*$, so $\mathcal{C}(\theta, \xi) \leq 0$ for all $(\theta, \xi) \in \Theta \times \Xi$. If θ^+ is taken to be the Moore-Penrose pseudoinverse of θ, then $\boldsymbol{Z}(\theta) = \hat{\boldsymbol{Z}}(\theta, \theta^+) = \hat{\boldsymbol{Z}}(\theta, \xi)$ so $\mathcal{C}(\theta, \xi) = 0$, which implies $\theta^+ \in \operatorname{argmin}_{\xi\in\Xi} \mathcal{C}(\theta, \xi)$. This implies that the set of maximizers is nonempty, proving the claim.

Finally, we claim that the function $\theta \mapsto \min_{\xi\in\Xi} \mathcal{C}(\theta, \xi)$ is constant. Indeed, by the choice of $\xi = \theta^+$ which is well-defined for all linear maps θ, this function is constantly zero, as desired. Thus, all assumptions of Theorem 3.2 hold. Applying the theorem, we have

$$\theta^\star \in \operatorname*{argmax}_{\theta\in\Theta} \mathcal{Q}(\theta) = \operatorname*{argmax}_{\theta\in\Theta} \Delta R(\boldsymbol{Z}(\theta) \mid \boldsymbol{y}) \tag{68}$$

$$\xi^\star \in \operatorname*{argmin}_{\xi\in\Xi} \mathcal{C}(\theta^\star, \xi) = \operatorname*{argmin}_{\xi\in\Xi} \sum_{j=1}^{K} \Delta R(\boldsymbol{Z}_j(\theta^\star), \hat{\boldsymbol{Z}}_j(\theta^\star, \xi)). \tag{69}$$

Then the theorem is proved using the characterizations of the optimizers in Lemmas 4.7 and 4.8. □

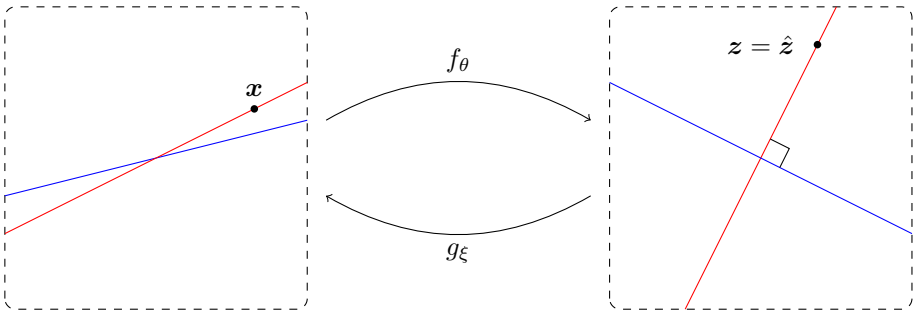


Figure 1: Visualization of the geometry of CTRL-MSP.

A diagram of the idealized operation of CTRL-MSP is visualized in Figure 1.

As the theorem indicates, the earlier check-list of desired quantitative properties can be achieved by CTRL-MSP. That is, CTRL-MSP provably learns injective and discriminative representations of multiple-subspace structure.

We now discuss an implication of the CTRL-MSP method. The original problem statement of learning discriminative representations for multiple subspace structure may be solved directly via orthogonalizing the representations produced by using PCA on each data subspace. This solution is a discrete and ad-hoc procedure that is far divorced from modern representation learning. However, CTRL-MSP provides an alternative approach: simultaneously learning and representing the subspaces via the modern representation learning toolkit within a continuous optimization framework. This gives a unifying perspective on classical and modern representation learning, by showing that classical methods can be viewed as *special cases* of modern methods, and that they may be formulated to learn the same types of representations. A major benefit is that the new formulation computationally can be generalized to much broader families of structures, beyond subspaces to submanifolds, as compelling empirical evidence from Dai et al. (2022) demonstrates.

5 Learning Proximal Equilibria

In this section we propose an algorithm for learning proximal equilibria using stochastic gradient descent. We use the toolkit of multiple-timescale stochastic approximation (Borkar, 2009), which has been used to show convergence to Nash equilibria in other instances of game-theoretic optimization (Heusel et al., 2017; Fiez et al., 2019, 2021; Sayin et al., 2021; Maheshwari et al., 2022).

5.1 Problem Formulation

We know that λ-proximal equilibria are Nash equilibria corresponding to the nonzero-sum game where the encoder utility is $\mathcal{V}_\lambda(\theta, \xi)$ and the decoder loss is $\mathcal{V}(\theta, \xi)$. We know how to characterize and compute (local) Nash equilibria of two-player games in computationally efficient ways (Ratliff et al., 2016; Heusel et al., 2017), provided that we have oracle access to $\nabla_\theta \mathcal{V}_\lambda$ and $\nabla_\xi \mathcal{V}$. While it is reasonable to assume that the original value function $\mathcal{V}$ is differentiable with known derivatives in both coordinates, the proximal value function $\mathcal{V}_\lambda$ is defined variationally, and thus it is not clear whether $\mathcal{V}_\lambda$ is even differentiable, much less what its derivative is. Happily, under broad assumptions which we outline below, we are able to compute the gradient using Danskin's theorem (Danskin, 1966).

Assumption 5.1.

(i) Ξ *is a compact set.*

(ii) For each $\xi \in \Xi$*, the function* $\mathcal{V}(\cdot, \xi)$ *is continuous.*

(iii) For each $(\theta, \xi) \in \Theta \times \Xi$*, the function* $\zeta \mapsto \mathcal{V}(\theta, \zeta) + \frac{\lambda}{2}\rho_\Xi(\xi, \zeta)^2$ *is strictly convex.*

Theorem 5.2 (Danskin's Theorem, Restated)**.** *Suppose Assumption 5.1 holds. Then $\mathcal{V}_\lambda(\cdot, \xi)$ is differentiable in its first argument, and moreover,*

$$\nabla_\theta \mathcal{V}_\lambda(\theta, \xi) = \nabla_\theta \mathcal{V}(\theta, \zeta^\star(\theta, \xi)) \tag{70}$$

where $\zeta^\star(\theta, \xi) \in \Xi$ is the unique solution to the problem $\min_{\zeta \in \Xi}\{\mathcal{V}(\theta, \zeta) + \frac{\lambda}{2}\rho_\Xi(\xi, \zeta)^2\}$.

Danskin's theorem motivates that we optimize θ, ξ, ζ jointly and ensure that ζ is roughly equilibrated with respect to θ and ξ at each update of θ and ξ. This motivates that we use the framework of multiple-timescale stochastic approximation (Borkar, 2009), which updates θ, ξ, and ζ at different timescales such that ζ heuristically appears equilibrated with respect to (θ, ξ) and ξ heuristically appears equilibrated with respect to θ.

5.2 Three-Timescale Proximal Gradient Descent-Ascent

Most previous works in this area (Heusel et al., 2017; Sayin et al., 2021; Maheshwari et al., 2022) use two timescales as they are only solving problems in two variables. In contrast, we propose the following three-timescale algorithm for learning proximal equilibria:

Algorithm 1 Three-Timescale Proximal Gradient Descent-Ascent

Require: Value function $\mathcal{V}\colon \Theta \times \Xi \to \mathbb{R}$, learning rates $(\alpha_n)_{n\in\mathbb{N}}, (\beta_n)_{n\in\mathbb{N}}, (\gamma_n)_{n\in\mathbb{N}}$, initializations $(\theta_0, \xi_0, \zeta_0) \in \Theta \times \Xi \times \Xi$.

for $m \geq 0$ **do**

$\theta_{n+1} \leftarrow \theta_n + \alpha_n \nabla_\theta \mathcal{V}(\theta_n, \zeta_n)$

$\xi_{n+1} \leftarrow \xi_n - \beta_n \nabla_\xi \mathcal{V}(\theta_n, \xi_n)$

$\zeta_{n+1} \leftarrow \zeta_n - \gamma_n \nabla_\zeta \mathcal{V}(\theta_n, \zeta_n) - \gamma_n \lambda(\zeta_n - \xi_n)$

return θ_n, ξ_n

This algorithm only requires oracle access to $\nabla\mathcal{V}$, which is easy compared to oracle access to $\nabla\mathcal{V}_\lambda$. Note that because ρ_Ξ is the Euclidean metric, we have $\nabla_\zeta \rho_\Xi(\xi, \zeta)^2 = 2(\zeta - \xi)$.

We are able to prove the correctness of this algorithm under restricted conditions, going along with Borkar (2009). To our knowledge, this is the first generalization of two-timescale stochastic approximation to three or more timescales, and so our analysis may be of independent interest.[5] For the sake of clarity we study a simplified dynamics i.e., not adding stochasticity to the updates, which would unfortunately make our analysis not directly applicable to stochastic gradient descent. However, we believe that the full-generality analysis is also possible in a straightforward manner with more work.

For brevity, let us make the following definitions:[6]

$$f(\theta, \xi, \zeta) = \nabla_\theta \mathcal{V}(\theta, \zeta) \tag{71}$$
$$g(\theta, \xi, \zeta) = -\nabla_\xi \mathcal{V}(\theta, \xi) \tag{72}$$
$$h(\theta, \xi, \zeta) = -[\nabla_\zeta \mathcal{V}(\theta, \zeta) + \lambda(\zeta - \xi)]. \tag{73}$$

In this notation, the update becomes

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_n f(\theta_n, \xi_n, \zeta_n) \\ \xi_{n+1} &= \xi_n + \beta_n g(\theta_n, \xi_n, \zeta_n) \\ \zeta_{n+1} &= \zeta_n + \gamma_n h(\theta_n, \xi_n, \zeta_n). \end{aligned} \tag{74}$$

Going along with Borkar (2009), we make the following strong assumptions about the underlying continuous dynamics. Note that assumptions of this form can be drastically weakened (Karmakar and Bhatnagar, 2018), but we choose these assumptions for simplicity and clarity of the underlying analysis.

[5] Borkar (2009) alludes to this possibility but does not study it.

[6] Note that we add some extraneous parameters; this is to make our analysis broadly generalizable to three-timescale dynamics instead of the restricted special case of proximal gradient descent-ascent dynamics. Essentially we do not need the extra structure that proximal gradient descent-ascent gives us over the generic three-timescale dynamics because our assumptions are strong. Refining this analysis is left to future work.

Assumption 5.3.

(i) The step sizes are strictly positive and such that

$$\sum_{n=0}^{\infty} \alpha_n = \sum_{n=0}^{\infty} \beta_n = \sum_{n=0}^{\infty} \gamma_n = \infty, \quad \sum_{n=0}^{\infty} (\alpha_n^2 + \beta_n^2 + \gamma_n^2) < \infty, \quad \lim_{n\to\infty} \frac{\alpha_n}{\beta_n} = 0, \quad \lim_{n\to\infty} \frac{\beta_n}{\gamma_n} = 0. \tag{75}$$

(ii) The mappings f, g, and h are Lipschitz.

(iii) For each $(\theta, \xi) \in \Theta \times \Xi$, the ODE $\frac{\mathrm{d}}{\mathrm{d}t}\zeta(t) = h(\theta, \xi, \zeta(t))$ has a unique globally asymptotically stable equilibrium $\zeta^\star(\theta, \xi)$, where $\zeta^\star \colon \Theta \times \Xi \to \Xi$ is Lipschitz.

(iv) For each $\theta \in \Theta$, the ODE $\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = g(\theta, \xi(t), \zeta^\star(\theta, \xi(t)))$ has a unique globally asymptotically stable equilibrium $\xi^\star(\theta)$, where $\xi^\star \colon \Theta \to \Xi$ is Lipschitz.

(v) The ODE $\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = f(\theta(t), \xi^\star(\theta(t)), \zeta^\star(\theta(t), \xi^\star(\theta(t))))$ has a unique globally asymptotically stable equilibrium $\theta^\star \in \Theta$.

(vi) $(\theta_n)_{n\in\mathbb{N}}$, $(\xi_n)_{n\in\mathbb{N}}$, and $(\zeta_n)_{n\in\mathbb{N}}$ are uniformly bounded sequences.

Assumption 5.3 (i) says that θ updates slowly compared to ξ and ζ, and that ξ updates slowly compared to ζ, hence the "multiple timescale" analysis. Thus, in discussion we informally call ζ the "fast timescale" variable, ξ the "medium timescale" variable, and θ the "slow timescale" variable. Assumption 5.3 (ii)—(v) say that the variables running at the faster timescales can equilibrate when compared to the slower timescales. Note that Assumption 5.3 (ii) can be replaced with Lipschitz guarantees on the gradients of the value function $\mathcal{V}$. Assumption 5.3 (vi) says that the discrete-time dynamics do not blow up as $m \to \infty$.

Theorem 5.4. *Under Assumption 5.3, Algorithm 1 learns a λ-proximal equilibrium $(\theta^\star, \xi^\star)$ associated with the proximal value function $\mathcal{V}_\lambda$ as $m \to \infty$.*

The proof of this theorem is long and complex, and so we break it up into a few parts. We adapt and generalize the strategy from Borkar (2009) to more than 2 timescales. The main workhorse we use combines a few theorems from Borkar (2009), which we present here.

Proposition 5.5. *Consider a Euclidean space E with associated Euclidean metric ρ_{E}. Let $\phi \colon \mathsf{E} \to \mathsf{E}$ be a function. Let $(\alpha_n)_{n\in\mathbb{N}}$ be a sequence of positive real numbers, and $(\varepsilon_n)_{n\in\mathbb{N}}$ a sequence of vectors in E. Consider the following discrete dynamical system:*

$$\mu_{n+1} = \mu_n + \alpha_n(\phi(\mu_n) + \varepsilon_n), \tag{76}$$

and the analogous continuous dynamical system:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu(t) = \phi(\mu(t)). \tag{77}$$

Assume that:

(i) The step sizes $(\alpha_n)_{n\in\mathbb{N}}$ are strictly positive and such that $\sum_{n=0}^{\infty} \alpha_n = \infty$ and $\sum_{n=0}^{\infty} \alpha_n^2 < \infty$.

(ii) The mapping $\phi \colon \mathsf{E} \to \mathsf{E}$ is Lipschitz.

(iii) The errors ε_n have $\lim_{n\to\infty} \varepsilon_n = 0$.

(iv) The ODE (77) has a globally asymptotically stable equilibrium.

(v) The iterates $(\mu_n)_{n\in\mathbb{N}}$ are uniformly bounded.

Define a continuous mapping $\bar{\mu} \colon \mathbb{R}_{\geq 0} \to \mathsf{E}$ by

$$\bar{\mu}(t_n) = \mu_n, \qquad \forall n \geq 0, \qquad \text{and linearly interpolated elsewhere.} \tag{78}$$

Finally, for $s \geq t$, define $\mu^s \colon s + \mathbb{R}_{\geq 0} \to \mathsf{E}$ as the solution to (77) with initial condition $\mu^s(s) = \bar{\mu}(s)$. Then:

Result 1. *(Borkar, 2009, Lemma 2.1). For any $T > 0$, we have $\lim_{s\to\infty} \sup_{t\in[s,s+T]} \rho_{\mathsf{E}}(\bar{\mu}(t), \mu^s(t)) = 0$.*

Result 2. *(Borkar, 2009, Theorem 2.2). $\lim_{n\to\infty} \mu_n$ converges to a global asymptotically stable equilibrium of (77).*

The careful reader may note that the stated results in Borkar (2009) are weaker and more general than Proposition 5.5; this is because we have added the assumption of a globally asymptotically stable equilibrium and obtained correspondingly significantly stronger results. This extended version of the results in Borkar (2009), which is presented in Proposition 5.5, is alluded to several times within Borkar (2009), but never formally stated. Collecting all the results in the most relevant setting for us into one lemma will allow our proofs to clearly demonstrate the technique of reducing multiple-timescale systems into a single-timescale system, which is useful from a technical and pedagogical perspective.

Now we continue with the proof of Theorem 5.4.

Lemma 5.6. *Under Assumption 5.3, we have $\lim_{n\to\infty} \rho_{\Xi}(\zeta_n, \zeta^\star(\theta_n, \xi_n)) = 0$.*

Proof. We rewrite the update in Equation (74) as:

$$\begin{aligned}
\theta_{n+1} &= \theta_n + \gamma_n \left[\frac{\alpha_n}{\gamma_n} f(\theta_n, \xi_n, \zeta_n)\right] \\
\xi_{n+1} &= \xi_n + \gamma_n \left[\frac{\beta_n}{\gamma_n} g(\theta_n, \xi_n, \zeta_n)\right] \\
\zeta_{n+1} &= \zeta_n + \gamma_n h(\theta_n, \xi_n, \zeta_n).
\end{aligned} \tag{79}$$

This is a discretization with step size γ_n of the coupled differential equation

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) &= 0 \\
\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) &= 0 \\
\frac{\mathrm{d}}{\mathrm{d}t}\zeta(t) &= h(\theta(t), \xi(t), \zeta(t))
\end{aligned} \tag{80}$$

in which case we have

$$\varepsilon_n \doteq \begin{bmatrix} \alpha_n/\gamma_n \cdot f(\theta_n, \xi_n, \zeta_n) \\ \beta_n/\gamma_n \cdot g(\theta_n, \xi_n, \zeta_n) \\ 0 \end{bmatrix}. \tag{81}$$

Because $\alpha = o(\gamma)$, $\beta = o(\gamma)$, the sequences $(\theta_n)_{n\in\mathbb{N}}, (\xi_n)_{n\in\mathbb{N}}, (\zeta_n)_{n\in\mathbb{N}}$ are uniformly bounded, and f, g, and h are Lipschitz, we have that $\varepsilon = o(1)$. Applying Proposition 5.5, we obtain convergence to the globally asymptotically stable equilibria of the ODE (80), which are of the form $(\theta, \xi, \zeta^\star(\theta, \xi))$ for $(\theta, \xi) \in \Theta \times \Xi$. The claim follows. □

Lemma 5.7. *Under Assumption 5.3, we have $\lim_{n\to\infty} \rho_{\Xi}(\xi_n, \xi^\star(\theta_n)) = 0$.*

Proof. We rewrite the update in (76) as

$$\begin{aligned}
\theta_{n+1} &= \theta_n + \beta_n \left[\frac{\alpha_n}{\beta_n} f(\theta_n, \xi_n, \zeta_n)\right] \\
\xi_{n+1} &= \xi_n + \beta_n g(\theta_n, \xi_n, \zeta_n)
\end{aligned} \tag{82}$$

and can rewrite the ξ update further as

$$\xi_{n+1} = \xi_n + \beta_n g(\theta_n, \xi_n, \zeta_n) \tag{83}$$

$$\begin{aligned}
&= \xi_n + \beta_n g(\theta_n, \xi_n, \zeta^\star(\theta_n, \xi_n)) \\
&\quad + \beta_n \left[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi_n, \zeta^\star(\theta_n, \xi_n))\right].
\end{aligned} \tag{84}$$

Thus we may again rewrite Equation (82) using this breakdown to get

$$\begin{aligned}\theta_{n+1} &= \theta_n + \beta_n \left[\frac{\alpha_n}{\beta_n} f(\theta_n, \xi_n, \zeta_n)\right] \\ \xi_{n+1} &= \xi_n + \beta_n g(\theta_n, \xi_n, \zeta^\star(\theta_n, \xi_n)) \\ &\quad + \beta_n \left[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi_n, \zeta^\star(\theta_n, \xi_n))\right].\end{aligned} \tag{85}$$

This is a discretization with step size β_n of the coupled differential equation

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}t}\xi(t) &= g(\theta(t), \xi(t), \zeta^\star(\theta(t), \xi(t)))\end{aligned} \tag{86}$$

in which case we have

$$\varepsilon_n \doteq \begin{bmatrix} \alpha_n/\beta_n \cdot f(\theta_n, \xi_n, \zeta_n) \\ \beta_n[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi_n, \zeta^\star(\theta_n, \xi_n))] \end{bmatrix}. \tag{87}$$

Because $\alpha = o(\beta)$, $\beta = o(1)$, the sequences $(\theta_n)_{n\in\mathbb{N}}, (\xi_n)_{n\in\mathbb{N}}, (\zeta^\star(\theta_n, \xi_n))_{n\in\mathbb{N}}$ are uniformly bounded, and f, g, and $\zeta^\star$ are Lipschitz, we have that $\varepsilon = o(1)$. Applying Proposition 5.5, we obtain convergence to the globally asymptotically stable equilibria of the ODE (86), which are of the form $(\theta, \xi^\star(\theta))$ for $\theta \in \Theta$. The claim follows. □

Lemma 5.8. *Under Assumption 5.3, we have* $\lim_{n\to\infty} \rho_\Theta(\theta_n, \theta^\star) = 0$.

Proof. We rewrite the θ update in (74) as

$$\theta_{n+1} = \theta_n + \alpha_n g(\theta_n, \xi_n, \zeta_n) \tag{88}$$

$$\begin{aligned}&= \theta_n + \alpha_n g(\theta_n, \xi^\star(\theta_n), \zeta^\star(\theta_n, \xi^\star(\theta_n))) \\ &\quad + \alpha_n[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi^\star(\theta_n), \zeta^\star(\theta_n, \xi^\star(\theta_n)))].\end{aligned} \tag{89}$$

Thus the update in Equation (74) may be written as

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_n g(\theta_n, \xi^\star(\theta_n), \zeta^\star(\theta_n, \xi^\star(\theta_n))) \\ &\quad + \alpha_n[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi^\star(\theta_n), \zeta^\star(\theta_n, \xi^\star(\theta_n)))].\end{aligned} \tag{90}$$

This is a discretization with step size α_n of the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = g(\theta(t), \xi^\star(\theta(t)), \zeta^\star(\theta(t), \xi^\star(\theta(t)))) \tag{91}$$

in which case we have

$$\varepsilon_n \doteq \alpha_n[g(\theta_n, \xi_n, \zeta_n) - g(\theta_n, \xi^\star(\theta_n), \zeta^\star(\theta_n, \xi^\star(\theta_n)))]. \tag{92}$$

Because $\alpha = o(1)$, the sequences $(\theta_n)_{n\in\mathbb{N}}, (\xi^\star(\theta_n))_{n\in\mathbb{N}}, (\zeta^\star(\theta_n, \xi^\star(\theta_n)))_{n\in\mathbb{N}}$ are uniformly bounded, and f, $\xi^\star$, and $\zeta^\star$ are Lipschitz, we have that $\varepsilon = o(1)$. Applying Proposition 5.5, we obtain convergence to the globally asymptotically stable equilibria of the ODE (86), which by uniqueness is just $\theta^\star$. The claim follows. □

Proof of Theorem 5.4. First, we use triangle inequality, continuity of $\xi^\star$, and convergence of $\theta_n \to \theta^\star$ to see that

$$\rho_\Xi(\xi_n, \xi^\star(\theta^\star)) \le \rho_\Xi(\xi_n, \xi^\star(\theta_n)) + \rho_\Xi(\xi^\star(\theta_n), \xi^\star(\theta^\star)) \tag{93}$$

$$\implies \lim_{n\to\infty} \rho_\Xi(\xi_n, \xi^\star(\theta^\star)) \le \lim_{n\to\infty} \{\rho_\Xi(\xi_n, \xi^\star(\theta_n)) + \rho_\Xi(\xi^\star(\theta_n), \xi^\star(\theta^\star))\} = 0. \tag{94}$$

Similarly $\rho_\Xi(\zeta_n, \zeta^\star(\theta^\star, \xi^\star(\theta^\star))) = 0$. Thus $(\theta_n, \xi_n, \zeta_n) \to (\theta^\star, \xi^\star(\theta^\star), \zeta^\star(\theta^\star, \xi^\star(\theta^\star)))$. □

Thus our three-timescale algorithm learns proximal equilibria, at least under restrictive settings.

6 Conclusion

In this work, we introduced the closed-loop proximal-games (CTRL-PG) framework for learning injective and discriminative representations for data. We explicitly characterized the proximal equilibria of the associated CTRL-PG game. We applied the CTRL-PG problems to two problems in learning theory and showed that it recovered and generalized classical optimal solutions, crystallizing a connection between modern deep representation learning frameworks and classical signal processing and statistics theory. Finally, we presented a novel algorithm to learn proximal equilibria.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. *Proceedings of the 34th International Conference on Machine Learning*, 70:214–223, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory, 2nd Edition*. Society for Industrial and Applied Mathematics, 1998. doi: 10.1137/1.9781611971132. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611971132.

Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, and Yi Ma. CTRL: Closed-Loop Transcription to an LDR via Minimaxing Rate Reduction. *Entropy*, 24(4), 2022. ISSN 1099-4300. doi: 10.3390/e24040456. URL https://www.mdpi.com/1099-4300/24/4/456.

Xili Dai, Ke Chen, Shengbang Tong, Jingyuan Zhang, Xingjian Gao, Mingyang Li, Druv Pai, Yuexiang Zhai, XIaojun Yuan, Heung-Yeung Shum, et al. Closed-loop transcription via convolutional sparse coding. *arXiv preprint arXiv:2302.09347*, 2023.

John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4): 641–664, 1966.

Farzan Farnia and Asuman Ozdaglar. GANs May Have No Nash Equilibria, 2020. URL https://arxiv.org/abs/2002.09124.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, October 2016. ISSN 0894-0347. doi: 10.1090/jams/852.

Soheil Feizi, Farzan Farnia, Tony Ginart, and David Tse. Understanding GANs in the LQG Setting: Formulation, Generalization and Stability. *IEEE Journal on Selected Areas in Information Theory*, 1(1):304–311, 2020. doi: 10.1109/JSAIT.2020.2991375.

Tanner Fiez, Benjamin Chasnov, and Lillian J. Ratliff. Convergence of Learning Dynamics in Stackelberg Games, 2019. URL https://arxiv.org/abs/1906.01217.

Tanner Fiez, Lillian Ratliff, Eric Mazumdar, Evan Faulkner, and Adhyyan Narang. Global convergence to local minmax equilibrium in classes of nonconvex zero-sum games. *Advances in Neural Information Processing Systems*, 34:29049–29063, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Trevor Hastie and Werner Stuetzle. Principal Curves. *Journal of the American Statistical Association*, 84 (406):502–516, 1989.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Harold Hotelling. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of educational psychology*, 24(6):417, 1933.

He Huang, Philip S. Yu, and Changhu Wang. An Introduction to Image Synthesis with Generative Adversarial Nets. *The Computing Research Repository (CoRR)*, abs/1803.04469, 2018. URL http://arxiv.org/abs/1803.04469.

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization? *ICML*, 2020.

Ian T Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2nd edition, 2002.

Olav Kallenberg. *Foundations of Modern Probability*. Probability Theory and Stochastic Modelling. Springer Cham, 2021. ISBN 9783030618704.

Prasenjit Karmakar and Shalabh Bhatnagar. Two time-scale stochastic approximation with controlled markov noise and off-policy temporal-difference learning. *Mathematics of Operations Research*, 43(1): 130–151, 2018.

T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(12):4217–4228, dec 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2020.2970919.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Yenson Lau, Qing Qu, Han-Wen Kuo, Pengcheng Zhou, Yuqian Zhang, and John Wright. Short and Sparse Deconvolution — A Geometric Approach. *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byg5ZANtvH.

Yanjun Li and Yoram Bresler. Global Geometry of Multichannel Sparse Blind Deconvolution on the Sphere. *Advances in Neural Information Processing Systems*, pages 1132–1143, 2018.

Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9): 1546–1562, 2007. doi: 10.1109/TPAMI.2007.1085.

Chinmay Maheshwari, Manxi Wu, Druv Pai, and Shankar Sastry. Independent and decentralized learning in markov potential games. *arXiv preprint arXiv:2205.14590*, 2022.

Ajkel Mino and Gerasimos Spanakis. LoGAN: Generating Logos with a Generative Adversarial Neural Network Conditioned on Color. *17th IEEE International Conference on Machine Learning and Applications*, pages 965–970, 2018.

Druv Pai, Michael Psenka, Chih-Yuan Chiu, Manxi Wu, Edgar Dobriban, and Yi Ma. Pursuit of a discriminative representation for multiple subspaces via sequential games. *arXiv preprint arXiv:2206.09120*, 2022.

Vignesh Prasad, Dipanjan Das, and Brojeshwar Bhowmick. Variational clustering: Leveraging variational autoencoders for image clustering. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2020.

Qing Qu, Yuexiang Zhai, Xiao Li, Yuqian Zhang, and Zhihui Zhu. Geometric Analysis of Nonconvex Optimization Landscapes for Overcomplete Learning. *International Conference on Learning Representations*, 2019.

Lillian J Ratliff, Samuel A Burden, and S Shankar Sastry. On the characterization of local nash equilibria in continuous games. *IEEE transactions on automatic control*, 61(8):2301–2307, 2016.

Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized q-learning in zero-sum markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

Yifei Shen, Ye Xue, Jun Zhang, Khaled Letaief, and Vincent Lau. Complete Dictionary Learning via ℓ_p-norm Maximization. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, 124: 280–289, 03–06 Aug 2020. URL https://proceedings.mlr.press/v124/shen20a.html.

Michael E Tipping and Christopher M Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality Reduction: a Comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

R. Vidal, Yi Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 1:I–I, 2003. doi: 10.1109/CVPR.2003.1211411.

René Vidal, Yi Ma, and Shankar Sastry. *Generalized Principal Component Analysis*. Springer Verlag, 2016.

John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning Diverse and Discriminative Representations via the Principle of Maximal Coding Rate Reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434, 2020. URL https://proceedings.neurips.cc/paper/2020/file/6ad4174eba19ecb5fed17411a34ff5e6-Paper.pdf.

Yuexiang Zhai, Hermish Mehta, Zhengyuan Zhou, and Yi Ma. Understanding ℓ_4-based Dictionary Learning: Interpretation, Stability, and Robustness. *International Conference on Learning Representations*, 2019.

Yuexiang Zhai, Zitong Yang, Zhenyu Liao, John Wright, and Yi Ma. Complete Dictionary Learning via ℓ_4-Norm Maximization over the Orthogonal Group. *J. Mach. Learn. Res.*, 21(165):1–68, 2020.

Yuqian Zhang, Han-Wen Kuo, and John Wright. Structured Local Optima in Sparse Blind Deconvolution. *IEEE Transactions on Information Theory*, 66(1):419–452, 2019.

Banghua Zhu, Jiantao Jiao, and David Tse. Deconstructing Generative Adversarial Networks. *IEEE Transactions on Information Theory*, 66(11):7155–7179, 2020.