Distribution Agreement

Signature:

_____________________________________ ____________

Eslam Abdelaleem Date

Simultaneous Dimensionality Reduction for Extracting Useful Representations of Large Empirical Multimodal Datasets

By

Eslam Abdelaleem
Doctor of Philosophy

Physics Department

Ilya Nemenman, Ph.D.
Advisor

Christopher J. Rozell, Ph.D.
Committee Member

Daniel M. Sussman, Ph.D.
Committee Member

Gordon J. Berman, Ph.D.
Committee Member

Justin C. Burton, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Simultaneous Dimensionality Reduction for Extracting Useful Representations of Large Empirical Multimodal Datasets

By

Eslam Abdelaleem
B.Sc., University of Science and Technology at Zewail City, Giza, Egypt, 2019
M.Sc., Emory University, Atlanta, Georgia, 2023

Advisor: Ilya Nemenman, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Physics Department
2024

Abstract

Simultaneous Dimensionality Reduction for Extracting Useful Representations of Large Empirical Multimodal Datasets
By Eslam Abdelaleem

The quest for simplification in physics drives the exploration of concise mathematical representations for complex systems. This Dissertation focuses on the concept of dimensionality reduction as a means to obtain low-dimensional descriptions from high-dimensional data, facilitating comprehension and analysis. We address the challenges posed by real-world data that defy conventional assumptions, such as the complex interactions within neural systems or high-dimensional dynamical systems. Leveraging insights from both theoretical physics and machine learning, this work unifies diverse dimensionality reduction methods under a comprehensive framework, the Deep Variational Multivariate Information Bottleneck. This framework enables the design of tailored reduction algorithms based on specific research questions and data characteristics. We explore and assert the efficacy of simultaneous dimensionality reduction approaches over their independent reduction counterparts, demonstrating their superiority in capturing covariation between multiple modalities, while requiring less data. We also introduced novel techniques, such as the Deep Variational Symmetric Information Bottleneck, for general nonlinear simultaneous dimensionality reduction. We show that the same principle of simultaneous reduction is the key to efficient and precise estimation of mutual information, a fundamental measure of statistical dependencies. We show that our new method is able to discover the coordinates of high dimensional observations of dynamical systems. Through analytical investigations and empirical validations, we shed light on the intricacies of dimensionality reduction methods, paving the way for enhanced data analysis across various domains. We underscore the potential of these methodologies to extract meaningful insights from complex datasets, driving advancements in fundamental research and applied sciences. As these methods evolve and find broader applications, they promise to deepen our understanding of complex systems and inform more effective experimental design and data analysis strategies.

Simultaneous Dimensionality Reduction for Extracting Useful Representations of Large Empirical Multimodal Datasets

By

Eslam Abdelaleem
B.Sc., University of Science and Technology at Zewail City, Giza, Egypt, 2019
M.Sc., Emory University, Atlanta, Georgia, 2023

Advisor: Ilya Nemenman, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Physics Department
2024

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ (٧٦)

صَدَقَ اللهُ العظيمُ

سورة يُوسُف

And above every scholar, is the All-Knower (76)

Yusuf

Acknowledgments

Five years is a long time, and it's not only the five years of the PhD, but also the time that came before them and prepared for them. This might seem like a long list of people, but I'm truly grateful and thankful for all of them, and I want to acknowledge their efforts that led to this work, either directly or indirectly.

I'd like to thank my advisor, Dr. Ilya Nemenman. Ilya, you're one of the smartest people I've met in my life. You taught me how to be a better scientist and a better person. Thanks for the lovely journey. I'd also like to thank my committee members (in no particular order): Drs. Gordon Berman, Justin Burton, Daniel Sussman, and Chris Rozell. Each and every one of you helped me in different ways, whether through classes, research, comments, discussions, or teaching. You helped shape this dissertation, so thank you!

I'd like to thank my labmates: Dr. Ahmed Roman, you're the only elder brother figure I have had in my life. Thanks for teaching me and helping me, not just in research, but in life. Dr. K. Michael Martini, I really enjoyed our time working together. It was the most productive time I had, and I learned a lot from you. Drs. Michael Pasek and Sean Ridout, thanks for the time and discussions you contributed to this work, and thanks for your advice in general. Satya, thanks for lending me an ear when needed and all the advice you gave me in navigating infinite logistic obstacles. Ketuna, Zehui, and Arabind, thanks for all the discussions and chats about random things we had. To all the lab members, you made my time more enjoyable, and I learned a lot from every one of you.

I want to thank my cohort as well: Katie, Pablo, Tomi, Charles, Lakshmi, Tharindu, Tao, Xinyu, and Richa. You guys were my family when I came to Emory, and I miss our lounge time together.

I'd like to thank Barbara for being there from day zero, without your help with everything, I wouldn't be writing this today.

I also want to thank my brothers who resided/roomed around the 415 Richards St. residence. You took me in, and you became true brothers to me. Thanks to each and every one of you, and the food was/is/will be amazing!

Sailing across the ocean toward home, I'd like to express my utmost heartfelt gratitude to the late Dr. Ahmed Zewail, the founding father who gave me the chance that led to me writing these words. My whole life, you have always been a symbol and a role model, and I feel great pride when I connect myself to you and call myself Zewailian.

I'd like to thank my physics major friends from Zewail City. Being with you shaped me into the person I am today, and I'm thankful. I'd also like to thank my beloved friends from section 4. Talking to you every day for the last five years (and before) was sometimes the only thing that kept me hanging in there. Thanks to each and every one of you. I'd like to thank the whole community of Zewail City for its immense effect that extended beyond the four years of undergrad and up until now. May your soul, Dr. Zewail, rest in heaven.

I'd like to express my unconditional love and gratitude to my whole family, who have been supportive, caring, and loving for my whole life. I'd like to thank my mom and dad, who have been my number one supporters in all conditions, who loved me, nourished me, and nurtured me, and haven't stopped until today. I'd like to thank my brother Anas for being there; I was able to work here because I believed my family was in good hands because of you. My sisters, Salma and Malak, you always supported me in different ways, and I always had you at my back. Thanks again. I'd like to thank my wife, Salma, for being the reason that this work is concluded, our last year together was the most productive and the hardest in some sense as well, thanks for being with me through the better part and the bitter one. I'd also like to express my gratitude to my two late uncles, Fayez and Hamada. You both shaped me in different ways, and I'm grateful for the time I had with you, sad for the time I didn't, but grateful nonetheless. I want to thank the rest of my extended family as well, who gave me a feeling that I have roots and always have a base of love and encouragement.

To everyone here, I'm truly grateful and duly acknowledging your effect on me, whether directly on this dissertation or directly on the person who worked on it. This is yours as much as it is mine. Thank you!

Contents

List of Figures

Chapter 1

Introduction: From Complexity to Sufficient Simplicity

We, physicists, are in a persistent search for simplification. We aim to describe complex systems using succinct equations or simplified representations. We are drawn to the notion that a simple equation, with a limited set of variables or "*features*", can faithfully replicate the behavior of seemingly complicated physical systems. Consider, for instance, the modeling of spin arrangements in a lattice by their collective magnetization or the characterization of an Avogadro's number of gas particles using coarse-grained variables, such as pressure, volume, and temperature, as used in the ideal gas law. These simplified representations are usually more intuitive, useful (in the sense that they offer predictions, limits for the interesting cases, etc.), and analytically tractable (at least we try), albeit underpinned by a series of implicit assumptions. Assumptions may involve operating within the thermodynamic limit with an infinite number of homogeneous constituents (as in infinite spin lattices or different statistical ensembles), or navigating simpler few-body problems (where 'few' typically refers to 1 or 2, general three-body problems lack closed-form solutions in Newtonian mechanics, for example). We note that the complexity of a description depends on its representation (the choice of the coordinates), and some coordinates are considered ‘natural’ because they lead to particularly simple descriptions (for example solving spherically symmetrical problems in spherical coordinates versus Cartesian coordinates). We often presume the existence of these ‘natural’ coordinates, either based on the physical observables of the problem or established from first principles, such as the natural association of pressure and temperature with measurable phenomena in the ideal gas law. However, the complexity of other real-world problems surpasses these idealizations. Take, for instance, a dataset obtained from recording the activity of a population of neurons (the information processing cells in the brain). When we try to model such systems, we are faced with multiple problems that challenge some of the common assumptions. For instance, the number of recorded state variables in these neurons is neither a few (so that we can model each neuron in biophysical detail), nor do we have an infinite number of uniform features that resemble the thermodynamic limit. Additionally, we do not have an infinite number of observations to statistically quantify all interactions of features among themselves. Moreover, the absence of locality adds more complexity to the situation (in the brain, neurons can be connected by long axons, so that there are no nearest neighbors interactions as in the Ising model, for example). Further, basic symmetries assumed in physical systems are not strictly preserved in the brain — the brain is neither rotationally nor translationally invariant [51]. In addition, dynamics in the brain, and even in each neuron, have multiple interacting temporal scales [63, 106]. For example, the same neuron exhibits ms scale precision during emission neural action potentials [88], but synaptic plasticity takes minutes, hours, or even years [4, 170, 135]). Simply put, there are no well-developed theoretical tools to produce succinct, interpretable mathematical descriptions of systems that violate these assumptions, commonly assumed in theoretical physics approaches. Most methods rely on the intuition of the researcher to propose a description of the system and then verify if it is useful in predicting the outcomes of new experiments. The cycle rarely converges to *quantitative* agreement between models and experiments, as we have grown to expect in more traditional areas of physics. Instead of relying solely on the scientist's intuition, an alternative approach is to leverage advances in statistics and machine learning to discover the optimal low-dimensional descriptions of experimental data for specific scientific questions. This is achieved through various methods of dimensionality reduction (DR), where one finds the optimal low-dimensional representations for a specific objective via optimization over certain classes of possible low-dimensional description space.

Such DR may be interpreted in two different languages, which may seem disparate in some contexts yet perfectly aligned in others. First is the language of theoretical physics, in which the Renormalization Group (RG) is the canonical example of dimensionality reduction [160]. RG is an iterative coarse-graining procedure designed to derive mathematical descriptions of physics problems characterized by multiple length scales. Its primary objective is to extract relevant features—and the laws of their interactions—of a physical system for describing phenomena at large length scales. For this, RG integrates out degrees of freedom at short length scales recursively, while keeping track of the effect these removed degrees of freedom have on the interactions among the ones that remain. Through this process, the relevant operators, the *important features*, gain prominence, while irrelevant operators start to have progressively smaller effects on the system's physical properties at large scales. Note that RG in physics is usually applied to models of complex systems, and applications directly to data are still uncommon [104, 90, 26]. The other language for DR comes from statistics, machine learning (ML), and computer science, where it is formalized as minimizing a loss function that promotes a succinct description of data directly. Here one proceeds as follows. First one postulates a measure of complexity of the description, which could be counting variables, measuring their variance, entropy, or a variety of other approaches. Second, one defines a measure of quality of a description of data, which is typically some measure of the ability to reconstruct or preserve the desired data features. Ultimately, most DR methods then are formulated by combining both measures into a single loss function, with opposite signs, balancing them against each other. Optimizing the loss function entails finding either the best compression with a fixed quality, or the highest quality at a fixed compression level.

The earliest statistical algorithm formulated in this language is probably the Principal Components Analysis (PCA) [75]. It works by finding the rotation matrix that makes the covariance matrix of the system diagonal. In such a rotated basis, the new orthogonal features of the system would be arranged in descending order in terms of their explained variance (that is, their contribution to the quality of reconstruction), with the hope that only a few of those dimensions (strong compression) would be able to capture most of the variance of the original system. These preserved dimensions would then be called the latent features of the system. More recently with the advances in ML, neural network-based methods, such as autoencoders [68] and their variational counterparts [87, 66], have become the state of the art in DR. Because of the neural networks' ability to approximate any continuous functions, such algorithms can search for nonlinear latent features that can minimize the loss function and describe the system better than linear methods, such as PCA. However, the simplicity and interpretability of PCA still make it a go-to method, even with all the recent advances in machine learning. Crucially, within the ML language, interpretability (or the lack of it) of latent features that emerge from nonlinear DR is a significant concern. This is especially true for neural network-based methods, where interpretability is almost nonexistent.

The similarity between the two languages is intriguing. Indeed, what the physicists call relevant operators are in some situations what the statisticians call latent features. Correspondingly, there's a body of literature that explores those mappings [103, 26], which can be exact in some scenarios. The biggest distinctions between the two are:

(i) largely model-based approach of RG methods vs. data-driven structure of ML methods, (ii) explicit use of physical symmetries in RG approaches, which is only now being incorporated into ML methods, and (iii) deep theoretical justification, understanding, and interpretability of RG and its findings as opposed to *ad hoc* design and success and poor interpretability of ML methods. We hope that this dissertation will make a dent in the last point, providing better theoretical underpinnings behind some ML-based DR methods.

By obtaining the latent features of the system and operating within this constructed low-dimensional space, we simplify our description of what initially appears to be a complex problem with many variables. This may make it easier to adequately sample and potentially model the system. The assumption that systems have latent low-dimensional descriptions is pervasive across various natural and social sciences fields, and applying DR methods to the analyzed data is sometimes the standard first step in any analysis in these fields. For example, in the analysis of neural activity [114, 139, 150, 140], behavior [140, 108, 150, 92], complex dynamical systems [40, 35], or even in seemingly very different fields, such as economics [128, 59, 48, 13] and systems management [57, 55], the assumption of utilizing a low-dimensional description for a seemingly high-dimensional system is usually the default choice. Under which conditions such systems have low dimensional representations is a poorly understood — and rarely studied — question. This, however, typically does not stand in the way of DR methods being used, useful, and often successful in these fields. For example, it is standard to study the activity of an animal brain by first performing some form of PCA on it to obtain a low-dimensional description of the neural activity [114], and then to use it to predict some measured behavior (usually, the number of principal components preserved is of order 10). In some cases, latent variables can be interpreted. For example, in economics, it is common to assume the existence of common latent features among the observed prices (certainly an assumption, with weak theoretical justification [83]). However, these latent features can then be interpreted as larger scale market indicators, such as the total market dynamics, inflation, or sector-specific indicators, etc. [49, 13, 59, 48].

Notably, when one talks about DR—including in the examples above—one typically reduces the dimensionality of a single large-dimensional set of variables. For example, in RG approaches, one combines microscopic spins into magnetization or the velocity of particles into the average velocity of a fluid in a mesoscopic volume. Similarly, in ML approaches to scientific data, one typically compresses data modalities one at a time. For example, in experiments on neural control of motor behavior, one separately finds a low-dimensional description of the neural activity and another low dimensional description of the behavior. However, while this might be sufficient to describe the system for some types of questions, it is not always enough. For example, in the neural control of behavior example, recording the neural activity without accounting for the resulting behavior makes it challenging to ensure that all the activity preserved during DR is truly relevant to the observed behavior, or that all the behavioral sequences identified by the DR are controlled neurally. Indeed, even a relatively simple organism like the *Drosophila melanogaster* fruit fly possesses hundreds of thousands of neurons across various connected brain regions [167] and exhibits numerous stereotyped behaviors [19], which change depending on the environment. Consequently, the same neural activity may result in different behaviors, and vice versa. Some neural activity is just internal signal processing and is not behaviorally relevant, while some behavior is purely a mechanical response to the environment and not neurally controlled.

These types of problems, involving dimensionality reduction in more than one set of qualitatively different variables, which in the ML language are termed multiple *views* or *modalities*, present a unique set of additional advantages and challenges in a field known as multiview or multimodal learning. Combining multiple sources of observations in the analysis enables a more detailed and relevant description of the system, in the sense that it disentangles dependencies among the sources. In the example above, one can find patterns of neural activity that predict behavior, as well as stereotypical behaviors that seem to be neurally controlled. However, with the benefits of multimodality, where each modality is often high-dimensional in its own right, the appropriate DR methods must necessarily become more complex, acknowledging the statistical structure of the data. For a single modality, DR is relatively straightforward: we need to preserve a quality of the description, while reducing its size. However, even in the simplest multimodal situation, where we have only two modalities[1], it is not immediately clear how to measure both the quality of the compression and the strength of the compression during DR. That is, one does not know which statistics of the two variables one needs to preserve, and the outcome of DR certainly depends on this choice. One prevalent approach in the literature boils down the multivariate approach to two univariate ones: one independently reduces the dimensionality of both modalities and subsequently seeks correlations between the low-dimensional descriptions. The rationale behind this approach is that, naively, it may require fewer samples[2] and is easier to implement[3], with the hope that the most relevant dimensions

[1] While theoretically (and in some situations, practically [99, 18, 162, 154]) one can consider more than two modalities (as demonstrated in 3.7.2), there are several limitations associated with doing so. For example, in the case of two modalities, one estimates covariance matrices, which have the dimensionality of the first modality times the second modality, and one needs to estimate elements of such matrix with good accuracy. However, for three modalities, one has to deal with a three-dimensional covariance tensor. This increases the number of elements needed to be estimated to guide the DR, and there is typically not enough data to do this well. Additionally, visualizing and interpreting three-way relationships becomes more challenging. Moreover, in many practical scenarios, two modalities are sufficient for addressing many relevant questions (cf. 2.3.1 and 2.6.2). Thus, from now on in this Dissertation, and unless specified otherwise, when we talk about multimodal datasets, we analyze them by considering two modalities at a time.

[2] Naively, estimating the variance within each modality might seem simpler than estimating the covariance between two modalities, suggesting that fewer samples would be required. However, as we will demonstrate in Chapter 2, this assumption does not always hold true, and there are caveats to consider for this argument.

[3] PCA, one of the simplest and most cited (about 6,910,000 papers mention it on Google Scholar at the time of writing) DR methods representing this approach, is essentially the eigenvalue problem 2.4.1.1, with multiple efficient implementations available in almost any programming language and statistical software.

for each modality might also be relevant for the other. In the language of linear models, reducing the multivariate problem to two single variate ones assumes that high variance directions within a modality correspond to high covariance directions across modalities. The validity of this assumption is unclear *a priori*.

An alternative, albeit less commonly used, approach to multivariate DR problems is to jointly and *simultaneously* reduce both modalities to yield joint low-dimensional descriptions, so that the low-dimensional description of one modality is maximally informative about the low-dimensional description of the other, and vice versa. In linear terms, this entails keeping the features that explain the highest covariance between both modalities, rather than their variances. Some sporadic research suggests that this approach is more favorable in terms of sample efficiency (requiring smaller datasets to achieve similar – or even better – accuracy than independently reducing each modality), and that it provides more concise representations (keeping fewer dimensions) [130, 36, 61]. Moreover, as we will demonstrate, in certain scenarios, independent reduction can lead to undesired outcomes by failing to capture the relevant features between modalities. This discrepancy can be intuitively understood, as there is no guarantee that maximum variation within one modality contributes to maximum covariation across them. For example, when recording a moving object of interest against a fast-changing background using two cameras, reducing the frames of each camera independently is likely to preserve details about the background more than preserving details about the object itself.

This Dissertation argues that simultaneous DR (SDR) approaches should almost always be preferred to independent DR (IDR) methods when the goal is to identify co-variation between two statistically related modalities. Such methods require less data and produce more interpretable descriptions than independent single reduction approaches. Further, they are quite easy to implement numerically, certainly not qualitatively more difficult than their single modality counterparts. In the three subsequent Chapters of this Dissertation, we provide quantitative arguments for this assertion by investigating specific important questions about the design, the practice, and the interpretation of SDR methods on a wide variety of multimodal datasets. Specifically, we address the following problems :

1. In multimodal setups, how do the outcomes of the DR process depend on using IDR vs SDR methods? What outcomes (that is, quality and interpretability of latent descriptions) can we anticipate from each approach? How do standard methods employing these approaches function? and How can we leverage the theoretical understanding of the methods to improve the standard of practice?

2. ML research has produced a multitude of DR methods, each differing in assumptions (such as whether the data is linear or not), implementation methods (linear algebra-based versus neural network-based), and more. This diversity has obvious advantages, but it also is overwhelming, limiting the ability to study and select the best of these methods for specific problems. We aspire to unify these methods within an intuitive yet mathematically tractable and rigorous framework. Ultimately, we aim to utilize this framework to develop methods that are not merely black-box solutions, but are tailored to specific research questions.

3. We want to ascertain the usefulness of such an understanding of simultaneous reduction. Among the multitude of ML-based Mutual Information (MI) estimators (MI being a fundamental measure of statistical dependencies between two variables), we study the successful ones – in terms of sample size requirements, consistency, and accuracy. We aim to understand them within our general framework of dimensionality reduction problem. We note that these methods effectively use a simultaneous reduction approach, which provides a low-dimensional description of the data that is maximally informative between

the two variables.

In more detail, in Chapter 2, I address the first of the above problems and investigate the effects of using SDR vs IDR methods in a specific multimodal setup. We explore how these different methods affect our results, under which conditions the true latent features can be discovered by different methods, and how this knowledge can be translated into a better practice. To answer these questions, we focus on an analytically and numerically tractable generative multimodal system. We employ a generative linear model with two modalities—each characterized by shared information between the modalities, self-information relevant for each one independently, and sample noise. The magnitudes of all of these features can be controlled independently. We focus on commonly used, powerful linear DR methods for both SDR and IDR. We explain the specific methods and why they may lead to distinct outcomes. Then we assess the quality of the low-dimensional representations obtained by applying these methods to data generated from the linear model, identifying key parameters relevant to the reduction process. Through the application of different DR methods from both approaches within this controlled setup, we gain essential insights into the DR process, shedding light on previously mentioned but poorly understood observations in the literature. Additionally, we propose a new heuristic to differentiate between self-information features unique to each modality and shared information features within multimodal datasets—a crucial step toward enhancing the interpretability and utility of DR outcomes. This Chapter serves as a motivational step, laying the groundwork for deeper analyses and methodological advancements explored in subsequent Chapters. Our aim here is to provide a comprehensive understanding of DR methods and their implications in tackling realistic, and yet tractable data.

In Chapter 3, I address the second of the above mentioned problems. There are many DR methods, each differing in assumptions and implementation choices. Different methods perform differently in distinct situations, but it is unclear how to choose a good method *a priori*. We systematize them by developing a comprehensive, mathematically rigorous, yet practical framework for unifying different DR methods, particularly the state-of-the-art deep variational ones. These methods utilize deep neural networks and variational approximations to learn robust and precise data representations, often acting as generative models for synthesizing samples from learned distributions. Additionally, using the framework, we can design new DR methods based on the needs of a practitioner: they need to specify what they want to preserve and how they believe the latent descriptions may depend on the observed data, and we automatically derive a corresponding DR algorithm and generate its neural network implementation. The framework is based on an interpretation of the information bottleneck (IB) principle. IB sets an explicit trade-off between the strength of compression and the quality of the latent description, both measured using information-theoretic quantities. More precisely, if we have a variable X that has some relationship to another variable Y, IB works by compressing X to a new low-dimensional variable Z that shares the most relation with Y. The quality of the low-dimensional latent space is measured using the mutual information (MI)[4] between Z and Y, while the strength of the compression is measured by how much information the latent state preserves about the compressed variable, $I(X;Z)$. Overall, the IB loss function is $\mathcal{L} = I(X;Z) - \beta I(Y;Z)$. Here β controls the quality of the compression, so that the information between X and Y is squeezed through a bottleneck of Z, giving the name to the method. In our more general framework, we trade off the information in an encoder graph, representing statistical structures used to derive the compressed variables (how the data should be encoded in a low-dimensional space), against that in a decoder graph, representing a generative model, which specifies how we want to reconstruct the variables of interest the low-dimensional compressed space. We then

[4] Mutual information between two continuous variables X and Y is a measure of all statistical dependencies between these two variables. It is defined formally as $I(X;Y) = \int\int p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\, dy$, where x and y are particular values of the variables X and Y, respectively. We discuss MI in depth in Chapters 3, 4.

approximate probability distributions in these graphs via variational approximations, which can be learned with the help of neural networks. We named this framework the *Deep Variational Multivariate Information Bottleneck (DVMIB)*. Crucially, we were able to retrieve multiple DR methods previously introduced in the literature as special cases of the framework. Additionally, we were able to improve and generalize some of these existing methods. Moreover, within this framework, we introduced a novel method, the *Deep Variational Symmetric Information Bottleneck (DVSIB)*, which is a specific variant of SDR that allows for simultaneous compression of the two modalities into distinct latent spaces that are maximally informative about one another. These latent representations are defined in two distinct spaces, potentially with different physical units, a quality highly sought in different fields. For example, going back to the previously mentioned example of neural activity and behavior, DVSIB would reduce the original high-dimensional neural activity into a low-dimensional one, and similarly for the behavior, such that the new low-dimensional descriptions are the ones that are maximally informative, and yet the neural and the behavioral latent spaces only use the neural activity and the behavioral recordings, respectively, in their definition. Another example that has been used extensively recently is within the multiview learning framework, where we learn a joint low-dimensional space from two (or more) modalities, such as images and their corresponding descriptive text. The goal is to have two encoders (one for each modality) that separately map the image and text into a shared latent space, allowing it to perform various tasks such as classification within that space. Additionally, two decoders are trained and used for tasks such as image retrieval and text-to-image generation. In such a scenario, the state-of-the-art architecture—Contrastive Language-Image Pre-Training (CLIP) [125]—is indeed a simultaneous reduction approach. This can be encapsulated (and potentially improved) within our new method, DVSIB. In our tests, our new method achieves better results in terms of classification accuracy and succinct low-dimensional representations on some test problems. Although real-world applications are mostly beyond the scope of this dissertation, a discussion of different applications is included in the chapters, with an application to a real physical system in the Discussion (Chapter 5). More interestingly, DVSIB verifies the previously developed intuition that the SDR paradigm is, indeed, more data-efficient than IDR, providing higher classification accuracy with a lower number of samples.

In Chapter 4, I study the problem of estimating Mutual Information (MI) as an SDR problem, aiming to provide efficient and precise guidelines for MI estimation in certain scenarios. MI naturally arises from fundamental principles in various fields like communication and probability theories to quantify statistical dependencies between variables. Despite its significance, accurately estimating MI from empirical data poses severe challenges, leading to the development of multiple estimation approaches over time, none of which work universally well. Traditional methods, such as parametric, nearest neighbor, and kernel-based methods, often struggle in high-dimensional scenarios. Conversely, recent neural network-based approaches have gained traction for their practicality and ability to handle high-dimensional data. However, these methods have mostly been tested in non-realistic, toy scenarios with effectively infinite amounts of data, which is not true of *any* realistic situation. The ability of these methods to work at finite sample sizes (i.e., the sample size dependencies of their biases and variances) is still poorly understood. These neural network methods, although versatile, lack inherent reporting criteria to warn users when their output should not be trusted. Since they are neural network-based methods, they are optimized via a training algorithm to minimize a certain loss function. However, the criteria for when to stop the optimization have not been established previously. Our analysis addresses these challenges by studying sample efficiency and providing accurate heuristics for determining when to terminate the training and whether to trust the output of the estimators.

In the last Chapter 5, I discuss promising directions for further inquiry. For example, we can use a DVSIB-like architecture to extract compact representations of dynamical systems observed via large-dimensional measurements. In this context, the two modalities may be the past and the future recordings of the dynamical system observations, and the low-dimensional descriptions then represent the dynamics of the coarse-grained description. For instance, consider deriving the laws of motion of a simple physical pendulum, where observations are movie frames recording its motion over time. The past then consists of a time window of frames, and the future could entail subsequent time windows. Then ideally a DR algorithm would extract the low-dimensional variables that are directly related to the angle and the angular velocity of the pendulum. Indeed, I show that our method can extract low-dimensional descriptions of these movies that are these physical quantities! Crucially, the method works essentially out-of-the-box, with no need to impute a lot of domain-specific physical knowledge to result in an accurate inference. I discuss the different relevant parameters for this problem, and its potential effects on the results. Another avenue worth exploring is uncovering the shared low-dimensional structure between neural recordings and resultant behavior. We anticipate that these low-dimensional spaces, to be discovered by our method, would offer improved accuracy in decoding corresponding behavior, coupled with interpretability and modeling potential owing to their generative and low-dimensional nature. The body of literature in the neuroscience domain that considers behavior while reducing neural activity typically compresses only the recorded neural activity conditioned on the behavior [130, 81, 131], with a few exceptions that consider both [54]. However, generally, this is not performing simultaneous reduction. As a result, we end up with a single low-dimensional space for the neural activity, while the behavior remains untouched (or processed separately), which is a limitation if the behavior is high-dimensional (such as videos or detailed motion). In the few exceptions where the behavior is also reduced, the neural activity and behavior are mixed together in a shared space, making it unclear how to interpret such a latent space. Having two distinct latent spaces for the two different domains as provided by DVSIB opens many interesting avenues worth studying.

Overall, this dissertation presents the work by me and my collaborators in advancing our understanding of DR methods through the lens of physics and information theory. By elucidating the differences between independent and simultaneous reduction techniques, I hope to have provided valuable insights into the underlying principles that can help in designing better physical models of complex systems. Moreover, the development of a unified framework has already facilitated the generalization of existing methods, while paving the way for the creation of novel approaches tailored to the specific challenges encountered in, but not limited to physics, machine learning, and life sciences research. Through the application of our methodologies, particularly in the estimation of mutual information, we have demonstrated their potential efficacy in extracting meaningful insights from data. This has implications not only for understanding fundamental physical phenomena, but also for optimizing experimental design and data analysis techniques in various fields. As we continue to refine and apply these methods in different research areas, we can expect further advancements in our understanding of the methods, and also more demonstrations of their utility for analysis of complex natural and man-made systems.

Chapter 2

On the Difference Between Independent and Simultaneous Dimensionality Reduction

2.1 Summary

[1]Current experiments frequently produce high-dimensional, multimodal datasets—such as those combining neural activity and animal behavior or gene expression and phenotypic profiling—with the goal of extracting useful correlations between the modalities. Often, the first step in analyzing such datasets is dimensionality reduction. We explore two primary classes of approaches to dimensionality reduction (DR): Independent Dimensionality Reduction (IDR) and Simultaneous Dimensionality Reduction (SDR). In IDR methods, of which Principal Components Analysis is a paradigmatic example, each modality is compressed independently, striving to

[1] This chapter presents the paper [3] with the title *Simultaneous Dimensionality Reduction: A Data Efficient Approach for Multimodal Representations Learning*. The work was conducted in collaboration with Ahmed Roman, K. Michael Martini, and Ilya Nemenman. I performed all simulations, conducted all analyses, and wrote the manuscript. Ilya conceived the model, while Ahmed and Michael contributed to discussions regarding the procedures and analyses. All authors reviewed the manuscript.

retain as much variation within each modality as possible. In contrast, in SDR, one simultaneously compresses the modalities to maximize the covariation between the reduced descriptions while paying less attention to how much individual variation is preserved. Paradigmatic examples include Partial Least Squares and Canonical Correlations Analysis. Even though these DR methods are a staple of statistics, their relative accuracy and data set size requirements are poorly understood. We use a generative linear model to synthesize multimodal data with known variance and covariance structures to examine these questions. We assess the accuracy of the reconstruction of the covariance structures as a function of the number of samples, signal-to-noise ratio, and the number of varying and covarying signals in the data. Using numerical experiments, we demonstrate that linear SDR methods consistently outperform linear IDR methods and yield higher-quality, more succinct reduced-dimensional representations with smaller datasets. Remarkably, regularized CCA can identify low-dimensional weak covarying structures even when the number of samples is much smaller than the dimensionality of the data, which is a regime challenging for all dimensionality reduction methods. Our work corroborates and explains previous observations in the literature that SDR can be more effective in detecting covariation patterns in data. These findings strengthen the intuition that SDR should be preferred to IDR in real-world data analysis when detecting covariation is more important than preserving variation.

2.2 Introduction

Many modern experiments across various fields generate massive multimodal data sets. For instance, in neuroscience, it is common to record the activity of a large number of neurons while simultaneously recording the resulting animal behavior [140, 139, 150, 92]. Other examples include measuring gene expressions of thousands of cells and their corresponding phenotypic profiles, or integrating gene expression data from different experimental platforms, such as RNA-Seq and microarray data [38, 166, 144, 80, 98]. In economics, important variables such as inflation are often measured using combinations of macroeconomic indicators as well as indicators belonging to different economic sectors [59, 13, 49, 128]. In all of these examples, an important goal is to estimate statistical correlations among the different modalities.

Analyses usually begin with dimensionality reduction (DR) into a smaller and more interpretable representation of the data. We distinguish two types of DR: *independent* (IDR) and *simultaneous* (SDR) [101]. In the former, each modality is reduced independently, while aiming to preserve its variation, which we call *self* signal. In the latter, the modalities are compressed simultaneously, while maximizing the covariation (or the *shared* signal) between the reduced descriptions and paying less attention to preserving the individual variation. It is not clear if IDR techniques, such as the Principal Components Analysis (PCA) [75], are well-suited for extracting shared signals since they may overlook features of the data that happen to be of low variance, but of high covariance [39, 23]. In particular, poorly sampled weak shared signals, common in high-dimensional datasets, can exacerbate this issue. SDR techniques, such as Partial Least Squares (PLS) [161] and Canonical Correlations Analysis (CCA) [76], are sometimes mentioned as more accurate in detecting weak shared signal [36, 61, 111]. However, the relative accuracy and data set size requirements for detecting the shared signals in the presence of self signals and noise remain poorly understood for both classes of methods.

In this study, we aim to assess the strengths and limitations of linear IDR, represented by PCA, and linear SDR, exemplified by PLS and CCA, in detecting weak shared signals. For this, we use a generative linear model that captures key features of relevant examples, including noise, the self signal, and the shared signal components. Using this model, we analyze the performance of the methods in different conditions.

Our goal is to assess how well these techniques can (i) extract the relevant shared signal and (ii) identify the dimensionality of the shared and the self signals from noisy, undersampled data. We investigate how the signal-to-noise ratios, the dimensionality of the reduced variables, and the method of computing correlations combine with the sample size to determine the quality of the DR. We propose best practices for achieving high-quality reduced representations with small sample sizes using these linear methods.

2.3 Model

2.3.1 Relations to Previous Work

The extraction of signals from large-dimensional data sets is a challenging task when the number of observations is comparable to or smaller than the dimensionality of the data. The undersampling problem introduces spurious correlations that may appear as signals, but are, in fact, just statistical fluctuations. This poses a challenge for DR techniques, as they may retain unnecessary dimensions or identify noise dimensions as true signals. Here, we focus exclusively on linear DR methods. For these, the Marchenko-Pastur (MP) distribution of eigenvalues of the covariance matrix of pure noise derived using the Random Matrix Theory (RMT) methods [100] has been used to introduce a cutoff between noise and true signal in real datasets. However, recent work [47] has shown that, when observations are a linear combination of uncorrelated noise and latent low-dimensional self signals, then the self signals alter the distribution of eigenvalues of the sampling noise, questioning the validity of this naive approach.

Moving beyond a single modality, [25] calculated the singular value spectrum of cross-correlations between two nominally uncorrelated random signals. However, it remains unknown whether the linear mixing of self signals and shared signals affects the spectra of noise, and how all of these components combine to limit the ability to detect shared signals between two modalities from data sets of realistic sizes. Filling in this gap using numerical simulations is the main goal of this work, and analytical treatment of this problem will be left for the future.

The linear model and linear DR approaches studied here do not capture the full complexity of real-world data sets and state-of-the-art algorithms. However, if sampling issues and self signals limit the ability of linear DR methods to extract shared signals, it would be surprising for nonlinear methods to succeed in similar scaling regimes on real data. Thus extending the previous work to explicitly study the effects of linear mixtures of self signals, shared signals, and noise on limitations of DR methods is likely to endow us with intuition that is useful in more complex scenarios routinely encountered in different domains of science.

Examples of scenarios with shared and self signals include inference of dynamics of a system through a latent space [40, 35], where shared signals correspond to latent factors that are relevant for predicting the future of the system from its past, while self signals correspond to nonpredictive variation [21]. In economics, shared and self signals correspond to diverse macroeconomic indicators that are grouped into correlated distinct categories in structural factor models [48, 59, 128, 13]. In neuroscience, shared signals can correspond to the latent space, by which neural activity affects behavior, while self signals encode neural activity that does not manifest in behavior and behavior that is not controlled by the part of the brain being recorded from [137, 140, 108, 130, 114, 150, 92].

Interestingly, in the context of the neural control of behavior, it was noticed that SDR reconstructs the shared neuro-behavioral latent space more efficiently and using a smaller number of samples than IDR [130]. Similar observations have been made in more general statistical contexts [36, 61, 111, 153], though the agreement is not uniform [55, 56, 57]. Because of this, most practical recommendations for detecting shared signals are heuristic [62], with widely acknowledged, but poorly understood limitations and possible resolutions [91]. Our goal is to ground such rules in numerical simulations and scaling arguments.

2.3.2 Linear Model with Self and Shared Signals

We consider a linear model with noise, $m_{\text{self,X}}, m_{\text{self,Y}}$ self signals that are relevant to each modality independently, as well as m_{shared} shared signals that capture the interrelationships between modalities.[2] It results in T observations of two high-dimensional standardized observables, X and Y:

$$\begin{aligned}
\left[\tilde{X} \in \mathbb{R}^{N_X}\right] &= \underbrace{R_X}_{\text{Independent white noise}} + \underbrace{U_X V_X}_{\text{Self-Signal for X}} + \underbrace{PQ_X}_{\text{Shared-Signal}} , \\
\left[\tilde{Y} \in \mathbb{R}^{N_Y}\right] &= \underbrace{R_Y}_{\text{Independent white noise}} + \underbrace{U_Y V_Y}_{\text{Self-Signal for Y}} + \underbrace{PQ_Y}_{\text{Shared-Signal}} , \qquad (2.1)
\end{aligned}$$

$$X = \tilde{X}/\sigma_{\tilde{X}}, Y = \tilde{Y}/\sigma_{\tilde{Y}}. \qquad (2.2)$$

The observations of X and Y are linear combinations of the following: (a) Independent white noise components R_X and R_Y with variances $\sigma^2_{R_X}$ and $\sigma^2_{R_Y}$. (b) Self-signal components U_X and U_Y residing in lower-dimensional subspaces $\mathbb{R}^{m_{\text{self},X}}$ and $\mathbb{R}^{m_{\text{self},Y}}$ with variances $\sigma^2_{U_X}$ and $\sigma^2_{U_Y}$. (c) Shared-signal components P in a shared lower-dimensional subspace $\mathbb{R}^{m_{\text{shared}}}$ with variance σ^2_P. These components are projected into their respective high-dimensional spaces $\mathbb{R}^{N_X}$ and $\mathbb{R}^{N_Y}$ using fixed quenched projection matrices V_X, V_Y, Q_X, and Q_Y with specified variances $\sigma^2_{V_X}$, $\sigma^2_{V_Y}$, $\sigma^2_{Q_X}$, and $\sigma^2_{Q_Y}$, all respectively. Entries in these matrices are drawn from a Gaussian distribution with a zero mean and the corresponding variances. Further, division by $\sigma_{\tilde{X}}$ and $\sigma_{\tilde{Y}}$ standardizes each column of the data matrices by their empirical standard deviations.

[2] This model is an extension of the model introduced by [47], and its probabilistic form has been studied by [107]. In its turn, the latter is an extension of work by [89], and [11]. However, within this model, we focus on the intensive limit, common in RMT [123], where the number of observations scales as the number of observed variables. This scenario is common in many real-world applications, and, to our knowledge, a similar extensive treatment to assess different DR methods as a function of various parameters of the system does not exist.

The total variance in the matrix $\tilde{X}$ can be calculated as the sum of the variances of its individual components: $\sigma^2_{\tilde{X}} = \sigma^2_{R_X} + m_{\text{self},X} \times \sigma^2_{U_X}\sigma^2_{V_X} + m_{\text{shared}} \times \sigma^2_P \sigma^2_{Q_X}$. A similar calculation can be done for the total variance in $\tilde{Y}$.

We define self and shared signal-to-noise ratios $\gamma_{\text{self},X/Y}, \gamma_{\text{shared},X/Y}$ as the relative strength of signals compared to background noise per component in each modality. These definitions allow us to examine how easily self or shared signals in each dimension can be distinguished from the noise.

$$\gamma_{\text{self},X/Y} = \frac{\sigma^2_{U_{X/Y}}\sigma^2_{V_{X/Y}}}{\sigma^2_{R_{X/Y}}}, \quad \gamma_{\text{shared},X/Y} = \frac{\sigma^2_P \sigma^2_{Q_{X/Y}}}{\sigma^2_{R_{X/Y}}} \tag{2.3}$$

Our main goal is to evaluate the ability of linear SDR and IDR methods to reconstruct the shared signal P, while overlooking the effects of the self signals $U_{X/Y}$ on the statistics of the shared ones.

2.4 Methods

We apply DR techniques to X and Y to obtain their reduced dimensional forms Z_X and Z_Y, respectively. Z_X, Z_Y are of sizes that can range from $T \times 1$ to $T \times N_X$ and $T \times N_Y$, respectively. As an IDR method, we use PCA [75]. As SDR methods, we apply PLS [161] and CCA [76, 152, 171], including both normal and regularized versions of the latter. Each of these methods focuses on specific parts of the overall covariance matrix

$$C_{X,Y} = \begin{bmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{bmatrix} = \begin{bmatrix} \frac{1}{T}X^\top X & \frac{1}{T}X^\top Y \\ \frac{1}{T}Y^\top X & \frac{1}{T}Y^\top Y \end{bmatrix}. \tag{2.4}$$

PCA aims to identify the most significant features that explain the majority of the *variance* in C_{XX} and C_{YY}, independently. PLS, on the other hand, focuses on singular values and vectors that explain the *covariance* component C_{XY}. Along the same lines, CCA aims to find linear combinations of X and Y that are responsible for the *correlation* $(C_{XY}/\sqrt{C_{XX}C_{YY}})$ between X and Y. A detailed description of each method is in the next section 2.4.1.

For every numerical experiment, we generate training and test data sets $(X_{\text{train}}, Y_{\text{train}})$ and $(X_{\text{test}}, Y_{\text{test}})$ according to Eqs. (2.1-2.2)[3]. We apply PCA, PLS, CCA, and regularized CCA (rCCA) to the training to obtain the singular directions $W_{X_{\text{train}}}$ and $W_{Y_{\text{train}}}$ for each method (see Appendix 2.4.1). We then obtain the projections of the test data on these singular directions

$$Z_X = X_{\text{test}} W_{X_{\text{train}}},$$
$$Z_Y = Y_{\text{test}} W_{Y_{\text{train}}}. \tag{2.5}$$

Finally, we evaluate the *reconstructed correlations* metric $\mathcal{RC}'$, which measures how well these singular directions recover the shared signals in the data, corrected by the expected positive bias due to the sampling noise, see Appendix 2.4.2 for details. $\mathcal{RC}' = 0$ corresponds to no overlap between the true and the recovered shared directions, and $\mathcal{RC}' = 1$ corresponds to perfect recovery.

2.4.1 Linear Dimensionality Reduction Methods

2.4.1.1 Principal Component Analysis - PCA

PCA is a widely used linear IDR method that aims to find the orthogonal principal directions, such that a few of them explain the largest possible fraction of the variance within the data. PCA decomposes the covariance matrix of the data matrix X, $C_{XX} = \frac{1}{T} X^\top X$, into its eigenvectors and eigenvalues through singular value decomposition (SVD). The SVD yields orthogonal directions, represented by the vectors $w_X^{(i)}$, that

[3] We fix $\sigma^2_{R_{X/Y}}, \sigma^2_{V_{X/Y}}, \sigma^2_{Q_{X/Y}}$ and allow $\sigma^2_{U_{X/Y}}, \sigma^2_P$ to vary when we choose $\gamma_{\text{self},X/Y}, \gamma_{\text{shared},X/Y}$. We first generate the fixed projection matrices $V_{X/Y}, Q_{X/Y}$, and we vary $R_{X/Y}, U_{X/Y}, P$ for each trial.

capture the most significant variability in the data. In most numerical implementations [118], these directions are obtained consecutively, one by one, such that the dot product between any two directions is zero $w_X^{(i)} \cdot w_X^{(j)} = \delta_{ij}$. The eigenvectors $w_X^{(i)}$ are obtained as the best solution to the optimization problem:

$$w_X^{*(i)} = \arg\max_{w_X^{(i)}} \frac{{w_X^{(i)}}^\top {X^{(i)}}^\top X^{(i)} w_X^{(i)}}{{w_X^{(i)}}^\top w_X^{(i)}}. \tag{2.6}$$

Here $X^{(i)}$ is the ith deflated matrix where $X^{(1)}$ is the original matrix, and for every subsequent $i+1$, the matrix is deflated by subtracting the projection of X on the obtained weights: $X^{(i+1)} = X - \Sigma_{s=1}^{i} X w_{(s)} w_{(s)}^\top$. The eigenvectors are sorted in decreasing order according to their corresponding eigenvalues, and the first k eigenvectors $w_X^{(i=1:k)}$ are selected to form the projection matrix W_X. The obtained vectors determine the size of the reduced form Z_X, where $|Z_X| = k$ is the number of vectors retained from the decomposition of X. The vectors $w_X^{(i)}$ are then stacked together to form the projection matrix W_X. The low-dimensional representation Z_X is then obtained by multiplying the original data matrix X with this projection matrix, resulting in the reduced data matrix $Z_X = XW_X$. Similar treatment is done for Y in order to obtain $Z_Y = YW_Y$

One of the main advantages of PCA is its simplicity and efficiency. However, one of the drawbacks of this method is that it performs DR for X and Y independently, and one then searches for relations between Z_X and Z_Y by regressing one on the other. Thus obtained low-dimensional descriptions may capture variance but not the covariance between the two datasets.

2.4.1.2 Partial Least Squares - PLS

PLS, or Partial Least Squares, performs SDR by finding the shared signals that explain the maximum covariance between two sets of data [161]. PLS performs the SVD of the covariance matrix $C_{XY} = \frac{1}{T}X^\top Y$ (or equivalently $C_{YX} = \frac{1}{T}Y^\top X$). The left and right singular vectors $(w_X^{*(i)}, w_Y^{*(i)})$ are obtained consecutively pair by pair such that $w_X^{(i)} \cdot w_Y^{(j)} = \delta_{ij}$. They are solutions of the optimization problem:

$$(w_X^{*(i)}, w_Y^{*(i)}) = \operatorname*{arg\,max}_{w_X^{(i)}, w_Y^{(i)}} \frac{{w_X^{(i)}}^\top {X^{(i)}}^\top Y^{(i)} w_Y^{(i)}}{\sqrt{({w_X^{(i)}}^\top w_X^{(i)})({w_Y^{(i)}}^\top w_Y^{(i)})}} \tag{2.7}$$

The matrices $X^{(i)}, Y^{(i)}$ are deflated in a similar manner to PCA 2.4.1.1. The singular vectors are sorted in the decreasing order of their corresponding singular values, and the first k vectors are selected to form the projection matrices (W_X, W_Y). The obtained vectors determine the size of the reduced form (Z_X, Z_Y), where $|Z_X| = |Z_Y| = k$ is the number of vectors retained. The vectors $(w_X^{(i)}, w_Y^{(i)})$ are then stacked together to form the projection matrices (W_X, W_Y) respectively. The low-dimensional representations (Z_X, Z_Y) are obtained by projecting the original data matrices (X, Y) onto these projection matrices: $Z_X = XW_X$, and $Z_Y = YW_Y$.

In summary, PLS performs simultaneous reduction on both datasets, maximizing the covariance between the reduced representations Z_X and Z_Y. This property makes PLS a powerful tool for studying the relationships between two datasets and identifying the underlying factors that explain their joint variability.

2.4.1.3 Canonical Correlation Ananlysis - CCA

2.4.1.3.1 Normal CCA

CCA is another SDR method, which aims to find the directions that explain the maximum correlation between two datasets [76]. However, unlike PLS, CCA obtains the shared signals by performing SVD on the correlation matrix $\frac{C_{XY}}{\sqrt{C_{XX}}\sqrt{C_{YY}}}$. The singular vectors $(w_X^{*(i)}, w_Y^{*(i)})$ are obtained consecutively pair by pair such that $w_X^{(i)} \cdot w_Y^{(j)} = \delta_{ij}$. CCA enforces the orthogonality of $w_X^{(i)}, w_Y^{(i)}$ independently as well, such that $w_X^{(i)} \cdot w_X^{(j)} = w_Y^{(i)} \cdot w_Y^{(j)} = \delta_{ij}$. The singular vectors are obtained by solving the optimization problem:

$$(w_X^{*(i)}, w_Y^{*(i)}) = \underset{w_X^{(i)}, w_Y^{(i)}}{\arg\max} \frac{{w_X^{(i)}}^\top {X^{(i)}}^\top Y^{(i)} w_Y^{(i)}}{\sqrt{({w_X^{(i)}}^\top {X^{(i)}}^\top X^{(i)} w_X^{(i)})({w_Y^{(i)}}^\top {Y^{(i)}}^\top Y^{(i)} w_Y^{(i)})}}. \tag{2.8}$$

Like in PLS 2.4.1.2, the matrices $X^{(i)}, Y^{(i)}$ are deflated in a similar manner. In addition, the first k singular vectors $(w_X^{*(i)}, w_Y^{*(i)})$ are stacked together to form the projection matrices (W_X, W_Y), which then are used to obtain the reduced data matrices $Z_X = XW_X$, and $Z_Y = YW_Y$.

One of the key differences between PLS and CCA is that while both perform SDR, CCA also simultaneously performs IDR implicitly. Indeed, it involves multiplication of C_{XY} by $C_{XX}^{-1/2}$ on the left and $C_{YY}^{-1/2}$ on the right, which, in turn, requires finding singular values of the X and the Y data matrices independently.

2.4.1.3.2 Regularized CCA - rCCA

While CCA is a useful method for finding the maximum correlating features between two sets of data, it does have some limitations. Specfically, in the undersampled regime, where $T \leq \max(N_X, N_Y)$, the matrices C_{XX} and C_{YY} are singular and their inverses do not exist. Using the pseudoinverse to solve the problem can lead to numerical instability and sensitivity to noise. Regularized CCA (rCCA) [152, 171] overcomes this problem by adding a small regularization term to the covariance matrices, allowing them to be invertible. Specifically, one tales

$$\tilde{C}_{XX} = C_{XX} + c_X I_X, \tag{2.9}$$

$$\tilde{C}_{YY} = C_{YY} + c_Y I_Y, \tag{2.10}$$

where $\tilde{C}_{XX}, \tilde{C}_{YY}$ are the new regularized matrices, $c_X, c_Y > 0$ are small regularization parameters and I_X, I_Y are identity matrices with sizes $N_X \times N_X, N_Y \times N_Y$ respectively.

This original implementation of rCCA resulted in correlation matrices with diagonals not equal to one. Thus, a better implementation uses a different form of regularization [171] by adding the regularization parameters c_X and c_Y individually to the equations as an affine combination (i. e., $\sum_i^n c_i = 1$) as the following:

$$\tilde{C}_{XX} = \frac{1}{T}(c_{X_1} w_X^\top X^\top X w_X + c_{X_2} w_X^\top w_X) \tag{2.11}$$

$$\tilde{C}_{YY} = \frac{1}{T}(c_{Y_1} w_Y^\top Y^\top Y w_Y + c_{Y_2} w_Y^\top w_Y). \tag{2.12}$$

This results in the regularized equations for X and Y to be:

$$\tilde{C}_{XX} \quad = \tfrac{1}{T}\big((1 - c_X) w_X^\top X^\top X w_X + c_X w_X^\top w_X\big) \tag{2.13}$$

$$\tilde{C}_{YY} \quad = \tfrac{1}{T}\big((1 - c_Y) w_Y^\top Y^\top Y w_Y + c_Y w_Y^\top w_Y\big), \tag{2.14}$$

where c_X and c_Y are the regularization parameters, with values between 0 and 1, resulting in solving the optimization problem:

$$\begin{aligned}(w_X^{*(i)}, w_Y^{*(i)}) = \underset{w_X^{(i)}, w_Y^{(i)}}{\arg\max} \quad & {w_X^{(i)}}^\top {X^{(i)}}^\top Y^{(i)} w_Y^{(i)} \\ \Big/ \Bigg(& \sqrt{(1 - c_X)({w_X^{(i)}}^\top {X^{(i)}}^\top X^{(i)} w_X^{(i)}) + c_X({w_X^{(i)}}^\top w_X^{(i)})} \\ & \sqrt{(1 - c_Y)({w_Y^{(i)}}^\top {Y^{(i)}}^\top Y^{(i)} w_Y^{(i)}) + c_Y({w_Y^{(i)}}^\top w_Y^{(i)})}\Bigg) .\end{aligned} \tag{2.15}$$

Writing the regularization conditions in this form is in fact a convex interpolation problem between PLS and CCA, which is a more robust solution and does not suffer from shortening the length of correlations due to the added regularization. As a result, this implementation of rCCA achieves the best accuracy among all other methods.

2.4.2 Assessing Success and Sampling Noise Treatment

To assess the success of DR, we calculated the ratio between the total correlation between $Z_{X_{\text{test}}}$ and $Z_{Y_{\text{test}}}$, defined as in Eq. (2.5), and the total correlation between X

and Y, which we input into the model. Specifically, we take the total correlation as the Frobenius norm of the correlation matrix, $||A||_F = \sqrt{\sum_i \sigma_i^2(A)}$, where $\sigma(A)$ are the singular values of the matrix A. Therefore, the metric of the quality of the DR is

$$\mathcal{RC} = \frac{||\mathrm{Corr}(Z_{X_{\mathrm{test}}}, Z_{Y_{\mathrm{test}}})||_F}{||\mathrm{Corr}(P,P)||_F} = \frac{||\mathrm{Corr}(Z_{X_{\mathrm{test}}}, Z_{Y_{\mathrm{test}}})||_F}{m_{\mathrm{shared}}}, \tag{2.16}$$

where Corr stands for the correlation matrix between its arguments, and we use $||\mathrm{Corr}(P,P)||_F = m_{\mathrm{shared}}$ as the total shared correlation that one needs to recover. Statistical fluctuations aside, $\mathcal{RC}$ should vary between zero (bad reconstruction of the shared variables) and one (perfect reconstruction).

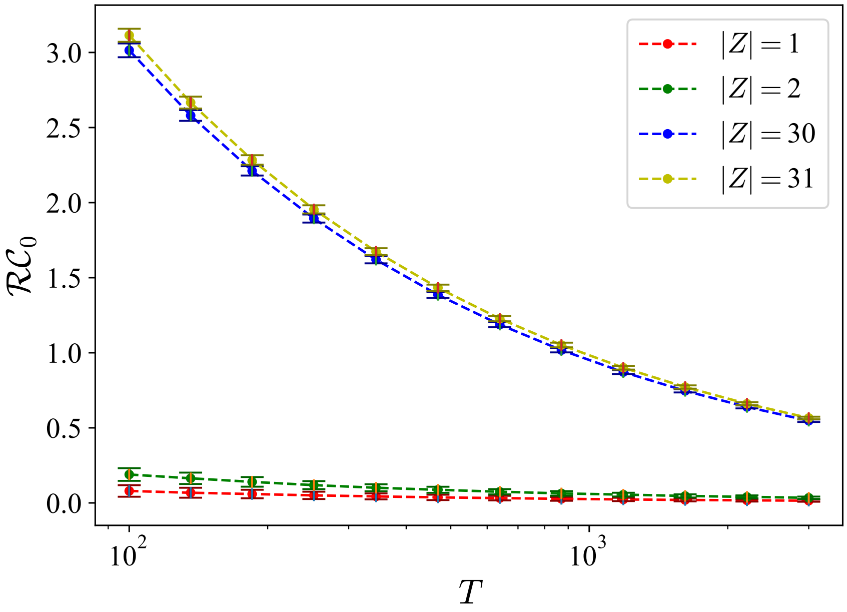


Figure 2.1: The resulting correlations are averages of all the points in the phasespace, then averaged over 10 different realizations of the matrices. The error bars are for two standard deviations around the mean

In many real-world applications, the number of available samples, T, is often limited compared to the dimensionality of the data, N_X and N_Y. This undersampling can introduce spurious correlations. We are not aware of analytical results to calculate the effects of the sampling noise on estimating singular values in the model in Eq. (2.1)

[28]. Thus, to estimate the effect of the sampling noise, we adopt an empirical approach. Specifically, we generate two random matrices, $Z_{X_{\text{random}}}$ and $Z_{Y_{\text{random}}}$, of sizes $T \times |Z_X|$ and $T \times |Z_Y|$, respectively. We then calculate the correlation between these matrices, denoted as $\mathcal{RC}_0$, for multiple such trials using the metric in Eq. (2.16). For random $Z_{X_{\text{random}}}$ and $Z_{Y_{\text{random}}}$, $\mathcal{RC}$ should be zero. However, Fig. 2.1 shows that, especially for large dimensionalities of the compressed variables and small T, the sampling noise results in a significant spurious $\mathcal{RC}_0 > 0$, which may even be larger than 1! Crucially, $\mathcal{RC}_0$ does not fluctuate around its mean across trials, so that the sampling bias is narrowly distributed.

To compensate for this sampling bias, we subtract it from the reconstruction quality metric,

$$\mathcal{RC}' = \mathcal{RC} - \mathcal{RC}_0. \tag{2.17}$$

It is this $\mathcal{RC}'$ that we plot in all Figures in this chapter as the ultimate metric of the reconstruction quality. While subtracting the bias is not the most rigorous mathematically, it provides a practical approach for reducing the effects of the sampling noise.

2.4.3 Implementation

We used Python and the `scikit-learn` [118] library for performing PCA, PLS, and CCA, while the `cca-zoo` [33] library was used for rCCA. For PCA, SVD was performed with default parameters. For PLS, the PLS Canonical method was used with the NIPALS algorithm. For both PLS and CCA, the tolerance was set to 10^{-4} with a maximum convergence limit of 5000 iterations. For rCCA, regularization parameters were set as $c_1 = c_2 = 0.1$. All other parameters not explicitly here were set to their default values.

All figures shown in this chapter were averaged over 10 independent realizations of R_X, R_Y, U_X, U_Y, P, while fixing the projection matrices V_X, V_Y, Q_X, Q_Y. We then performed an additional round of averaging everything over 10 realizations of the projection matrices themselves. The simulations were parallelized and run on Amazon Web Services (AWS) servers of instance types `ml.c5.2xlarge`.

2.5 Results

2.5.1 Results of the Linear Model

We perform numerical experiments to explore the undersampled regime, $T \lesssim N_X, N_Y$. We use $T = \{100, 300, 1000, 3000\}$ samples, $N_X = N_Y = 1000$. We explore the case of one shared signal only, $m_{\text{shared}} = 1$ and we mask this shared signal by a varying number of self signals and noise. We vary the number of retained dimensions, $(|Z_X|, |Z_Y|)$, and explore how many of them are needed to recover the shared signal in the noise and the self signal background with different SNR.

For brevity, we explore two cases: (1) One self-signal in X and Y in addition to the shared signal ($m_{\text{self}} = 1$); (2) many self-signals in X and Y. For both cases, we calculate the quality of reconstruction as the function of the shared and the self SNR, γ_{shared} and γ_{self}. In all figures, we show $\mathcal{RC}'$ for severely undersampled (first row, $T = 300$) and relatively well sampled (second row, $T = 3000$) regimes. We also show the value of $\mathcal{RC}_0$, the bias that we removed from our reconstruction quality metric, for completeness, see section 2.4.2 for details.

2.5.2 One self-signal in Xnd Yn addition to the shared signal ($m_{\textbf{self}} = 1$)

2.5.2.1 Keeping 1 dimension after reduction ($|Z_{X/Y}| = 1$)

Figure 2.2 shows that, in Case 1, when one dimension is retained in DR of X and Y, PCA populates the compressed variable with the largest variance signals and hence struggles to retain the shared signal when $\gamma_{\text{self}} > \gamma_{\text{shared}}$, regardless of the number of samples. However, both PLS and rCCA excel in achieving nearly perfect reconstructions. When $T \ll N_X$, straightforward CCA cannot be applied (see 2.4.1.3-2.4.1.3.2), but it too achieves a perfect reconstruction when $T > N_X$.

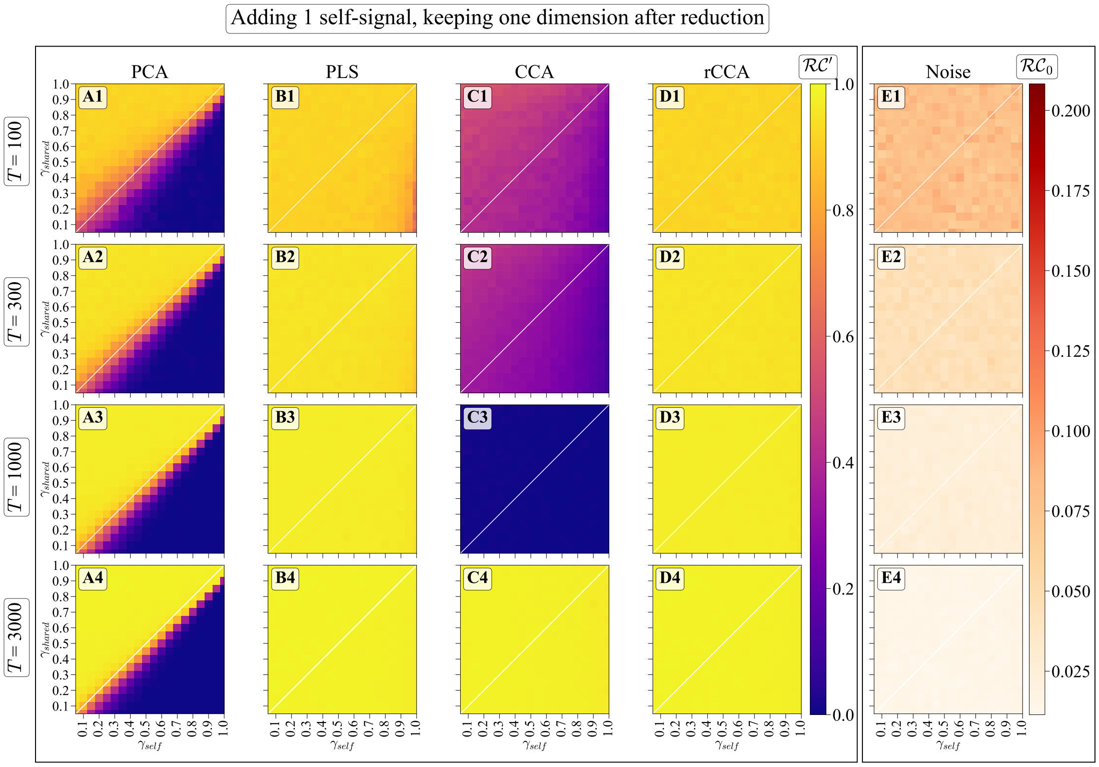


Figure 2.2: Performance of PCA, PLS, CCA, rCCA, and noise in recovery of the shared signal for $|Z_X| = |Z_Y| = 1 = m_{\text{self}}$. PCA struggles to detect shared signals when they are weaker than the self signals. PLS and rCCA demonstrate nearly perfect reconstruction. CCA displays no reconstruction in the undersampled regime $T \ll N_X$, and it is nearly perfect for large T.

2.5.2.2 Keeping 2 dimensions after reduction ($|Z_{X/Y}| = 2$)

In Fig. 2.3, we allow two dimensions in the reduced variables. For PCA, we expect this to be sufficient to preserve both the self and the shared signals. Indeed, PCA now works for all γs and Ts, although with a slightly reduced accuracy for large shared signals compared to Fig. 2.2. PLS and rCCA continue to deliver highly accurate reconstructions. So does the CCA for $T > N_X$. Spurious correlations, as measured by $\mathcal{RC}_0$ grow slightly with the increasing dimensionality of Z_X, Z_Y compared to Fig. 2.2. This is expected since more projections must now be inferred from the same amount of data.

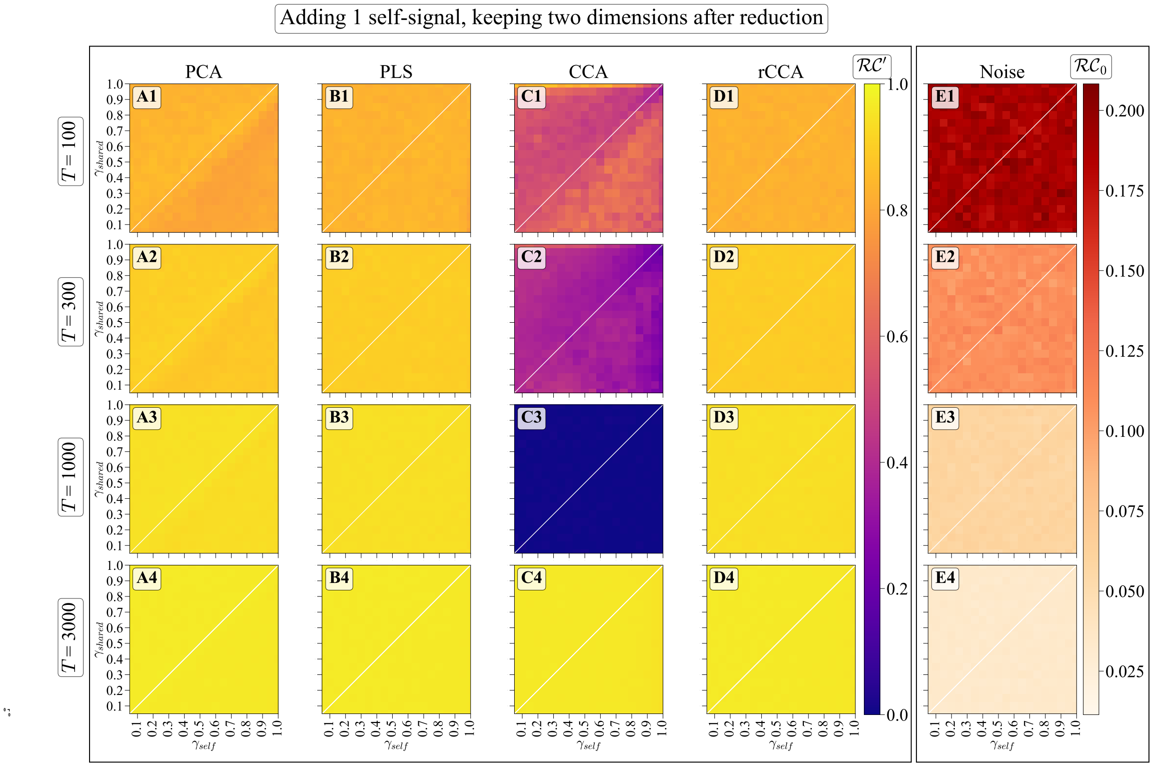


Figure 2.3: Same as Fig. 2.2, but for $|Z_X| = |Z_Y| = 2 = m_{\text{self}} + m_{\text{shared}}$. Now there are enough compressed variables for PCA to detect the shared signal. Other methods perform similarly to Fig. 2.2, albeit the noise is larger.

2.5.3 Many self-signal in Xnd Yn addition to the shared signal ($m_{\textbf{self}} = 30$)

2.5.3.1 Keeping 1 dimension after reduction ($|Z_{X/Y}| = 1$)

We now turn to $m_{\text{self}} \gg m_{\text{shared}}$. We use $m_{\text{shared}} = 1$, $m_{\text{self}} = 30$ for concreteness. We expect that the performance of SDR methods will degrade weakly, as they are designed to be less sensitive to the masking effects of the self signals. In contrast, we expect IDR to be more easily confused by the many strong self-signals, degrading the performance. Indeed, Fig. 2.4 shows that PCA now faces challenges in detecting shared signals, even when the self signals are weaker than in Fig. 2.2. Increasing T improves its performance only slightly. Somewhat surprisingly, PLS performance also degrades, with improvements at $T \gg N_X$. CCA again displays no reconstruction when $T \ll N_X$, switching to near perfect reconstruction at large T. Crucially, rCCA again shines, maintaining its strong performance, consistently demonstrating nearly perfect reconstruction.

2.5.3.2 Keeping 30 dimensions after reduction ($|Z_{X/Y}| = 30$)

Since one retained dimension is not sufficient for PCA to represent the shared signal when $\gamma_{\text{shared}} \lesssim \gamma_{\text{self}}$, we increase the dimensionality of reduced variables $|Z_X| = |Z_Y| = m_{\text{self}} \gg m_{\text{shared}})$, cf. Fig. 2.5. PCA now detects shared signals even when they are weaker than the self-signals, $\gamma_{\text{shared}} < \gamma_{\text{self}}$, but at a cost of the reconstruction accuracy plateauing significantly below 1. In other words, when self and shared signals are comparable, they mix, allowing for partial reconstruction. However, even at $T \gg N_X$, PCA cannot break into the phase diagram's lower right corner. Other methods perform similarly, reconstructing shared signals over the same or wider ranges of sampling and the SNR ratios than in Fig. 2.4. For all of them, the improvement comes at the cost of the decreased asymptotic performance. The most distinct feature of this regime is the

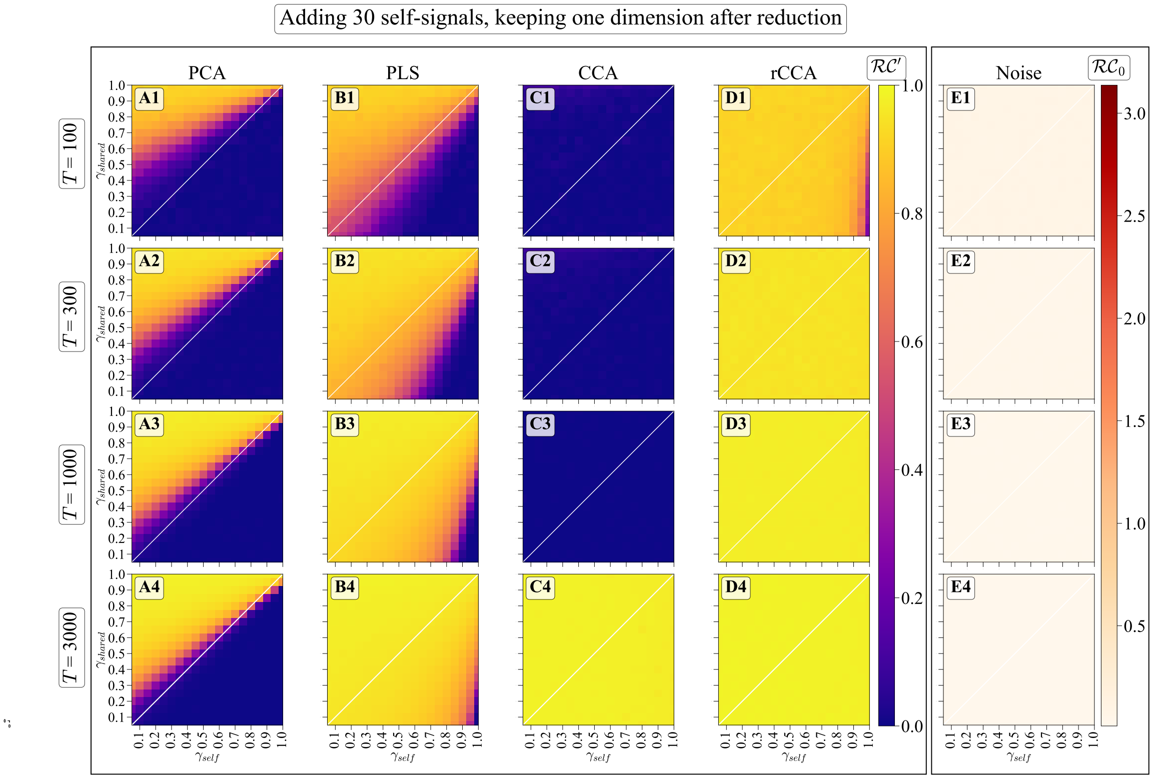


Figure 2.4: Reconstruction results for $m_{\text{self}} = 30$, $m_{\text{shared}} = 1$, and $|Z_X| = |Z_Y| = 1$. PCA struggles to detect any shared signals when they are even comparable to the self ones. PLS performance also degrades. CCA displays its usual impotence at small T. Finally, rCCA demonstrates nearly perfect reconstruction for all parameter values.

dramatic effect of noise, where 30-dimensional compressed variables can accumulate enough sampling fluctuations to recover correlations that are supposedly nearly twice as high as the data actually has.

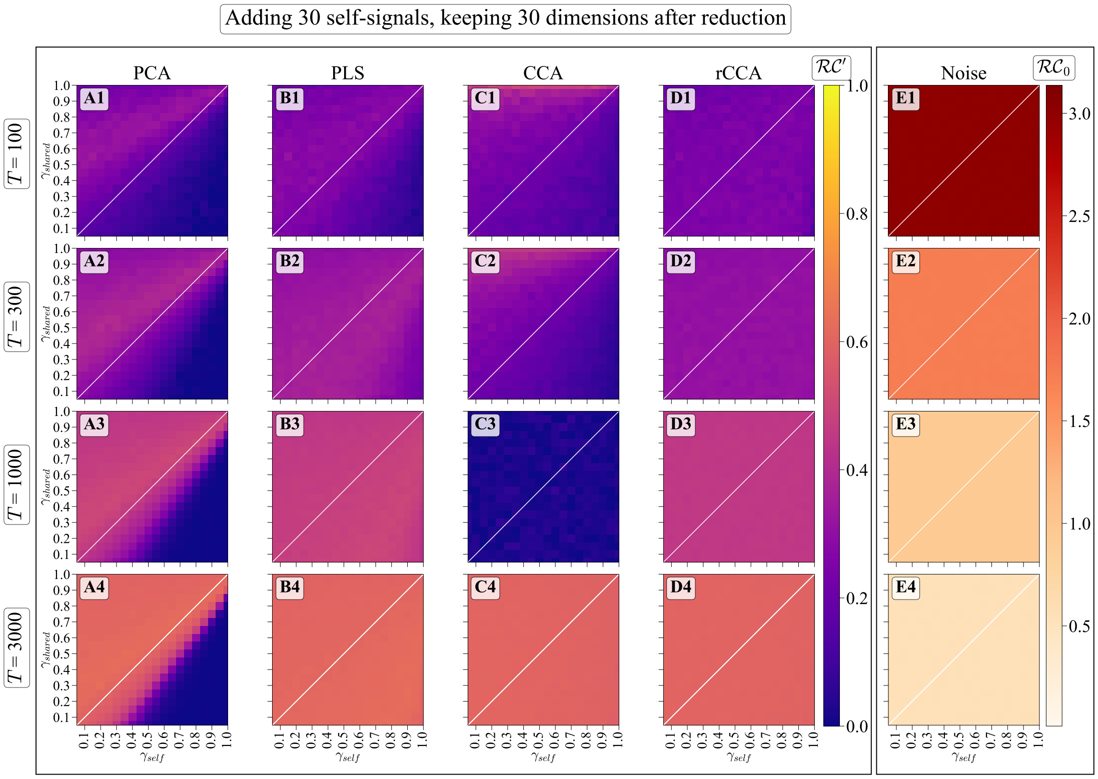


Figure 2.5: DR performance for $|Z_X| = |Z_Y| = m_{\rm self} > m_{\rm shared}$). PCA now detects shared signals even when they are weaker than the self signals. However, the quality of reconstruction is significantly lower than in Fig. 2.3. PLS detects signals in a larger part of the phase space, but also with a significant reduction in quality, which improves with sampling. CCA has its usual problem for $T \ll N_X$, and, like PLS, it has a significantly lower reconstruction quality than in the regime in Fig. 2.4. rCCA is able to detect the signal in the whole phase space, but again with worse quality. Finally, spurious correlations are high, though they decrease with better sampling.

2.5.3.3 Keeping 31 dimensions after reduction ($|Z_{X/Y}| = 31$)

Figure 2.6 now explores a regime when the dimensionality of the compressed variables is enough to store both the self and the shared interactions at the same time, $|Z_X| = |Z_Y| = m_{\rm self} + m_{\rm shared} = 31$. With just one more dimension than Fig. 2.5, PCA abruptly transitions to being able to recover shared signals for all SNRs, albeit still saturating at a far from perfect performance at large T. PLS, CCA, rCCA, and noise show behavior remain similar to Fig. 2.5.

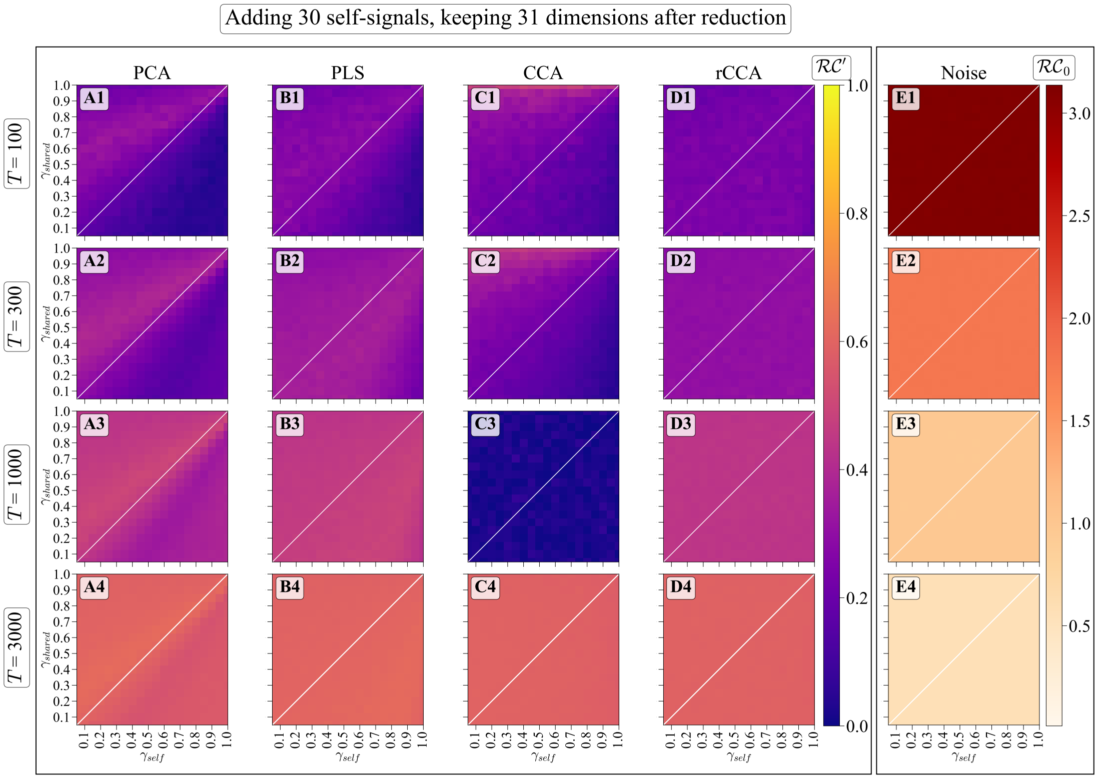


Figure 2.6: PCA, PLS, CCA, rCCA, and noise results when 31 dimensions are kept after reduction ($|Z_X| = |Z_Y| = m_{\text{self}} + m_{\text{shared}}$). PCA now can detect more shared signals when they are weaker than the self signals (A1), however, with a significantly lower quality compared to figure 2.3, but suddenly explores the whole phase space, still with lower accuracy than Case 1. PLS, CCA, rCCA, and noise show similar behavior to figure 2.5.

2.5.4 Key Parameters and Testing Technique for Dimensionality of Self and Shared Signals

Our analysis suggests that there are three relevant factors that determine the ability of DR to reconstruct shared signals. The first is the strength of the shared and the self signals compared to each other and to noise. For brevity, in the following analysis, we fix γ_{self} and define the ratio $\tilde{\gamma} = \gamma_{\text{shared}}/\gamma_{\text{self}}$ to represent this effect. The second factor affecting the performance is the ratio between the number of shared and self signals, denoted by $\tilde{m} = m_{\text{shared}}/m_{\text{self}}$. The third factor is the number of samples per dimension of the reduced variable, denoted by $\tilde{q} = T/|Z|$.

In Fig. 2.7, we illustrate how these parameters influence the performance of DR, $\mathcal{RC}'$. Each subplot varies $\tilde{q}$, while holding T constant and changing $|Z_X|$. We compare the results of PCA (representing IDR) and rCCA (representing SDR). Each curve is averaged over 10 trials, with error bars indicating 1 standard deviation around the mean, using algorithmic parameters as described in section 2.4.3.

We see that the relative strength of signals, as represented by $\tilde{\gamma}$, plays a significant role in determining which method performs better. If the shared signals are larger (bottom) both approaches work. However, for weak shared signals (top), SDR is generally more effective. Further, the ratio between the number of shared and self signals, $\tilde{m}$, also plays an important role. When $\tilde{m}$ is large (left), IDR is more likely to detect the shared signal before the self signals, and it approaches the performance of SDR. However, when $\tilde{m}$ is small, IDR is more likely to capture the self signals before moving on to the shared signals, degrading performance (right). Finally, not surprisingly, the number of samples per dimension of the compressed variables, $\tilde{q}$, is also critical to the success. If $\tilde{q}$ is small, the signal is drowned in the sampling noise, and adding more retained dimensions hurts the DR process. This expresses itself as a peak for SDR performance around $|Z_X| = m_{\text{shared}}$. For IDR, the peak is around $|Z_X| = m_{\text{self}} + m_{\text{shared}}$, thus requiring more data to achieve performance similar to SDR.

We observe that the performance of rCCA (SDR) is almost independent of changing $\tilde{m}$ or $\tilde{\gamma}$, indicating that it focuses on shared dimensions even if the latter is masked by self signals. The algorithm crucially depends on $\tilde{q}$, where adding more dimensions (decreasing $\tilde{q}$) than needed hurts the reduction. This is because, for a fixed number of samples, the reconstruction of each dimension then gets worse. In contrast, for PCA (IDR), the performance depends on all three relevant parameters, $\tilde{q}$, $\tilde{m}$, and $\tilde{\gamma}$. At some parameter combinations, the performance of IDR in reconstructing shared signals approaches SDR. However, in all cases, SDR never performs worse than IDR on this task.

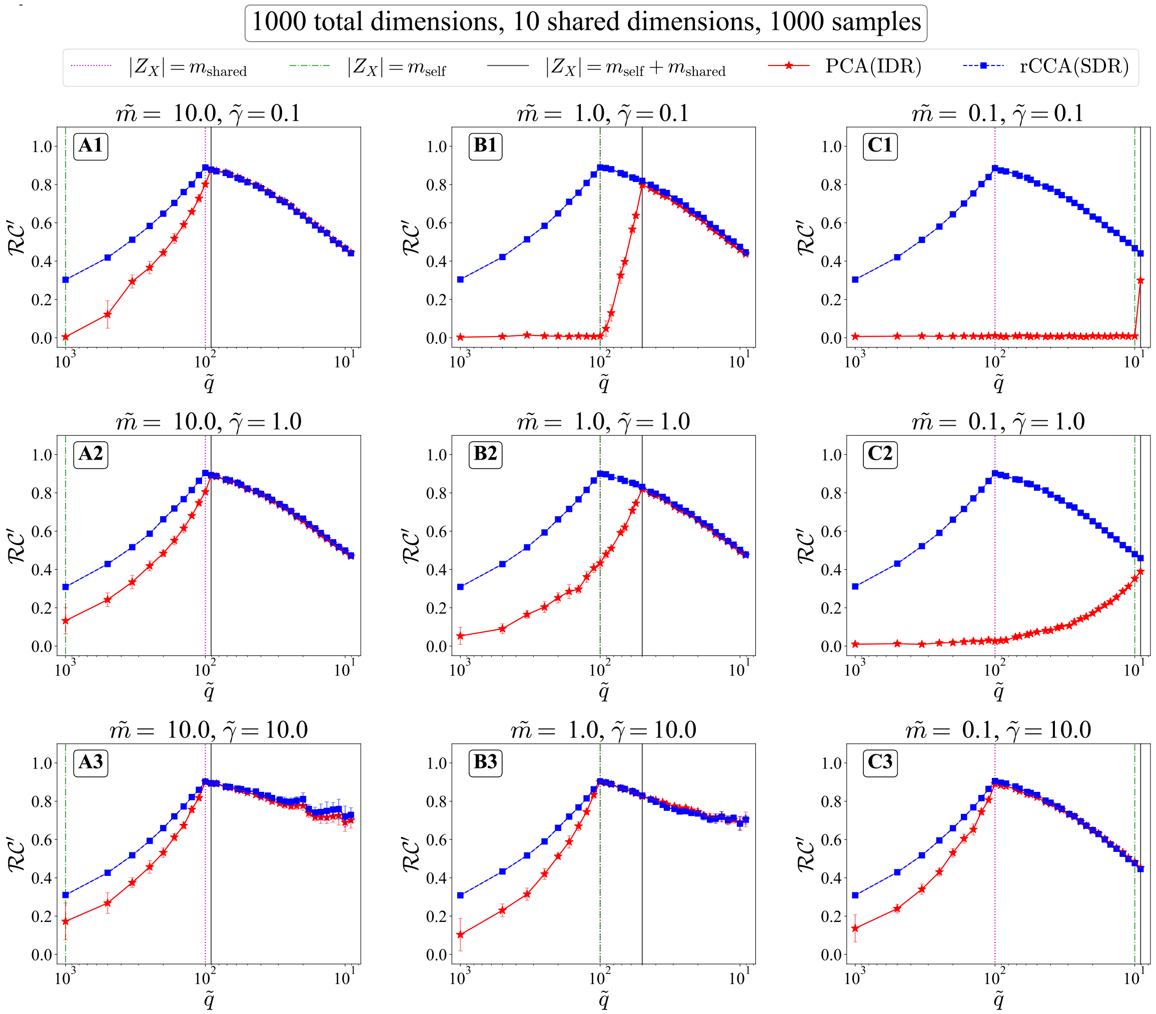


Figure 2.7: Performance of PCA (IDR) and rCCA (SDR) for different values of the relevant parameters of the model: the number of samples per dimension of the compressed variable ($\tilde{q}$), the strength of shared signals relative to the self ones ($\tilde{\gamma}$), and the ratio of the number of shared to self signal components ($\tilde{m}$), while fixing the number of samples ($T = 1000$) and the number of shared dimensions ($m_{\text{shared}} = 10$). Note that decreasing $\tilde{q}$ (left to right) corresponds to increasing the dimension of the latent space $|Z_X|$ at a fixed number of samples T.

2.5.5 Beyond Linear Models - Noisy MNIST

2.5.5.1 The Dataset

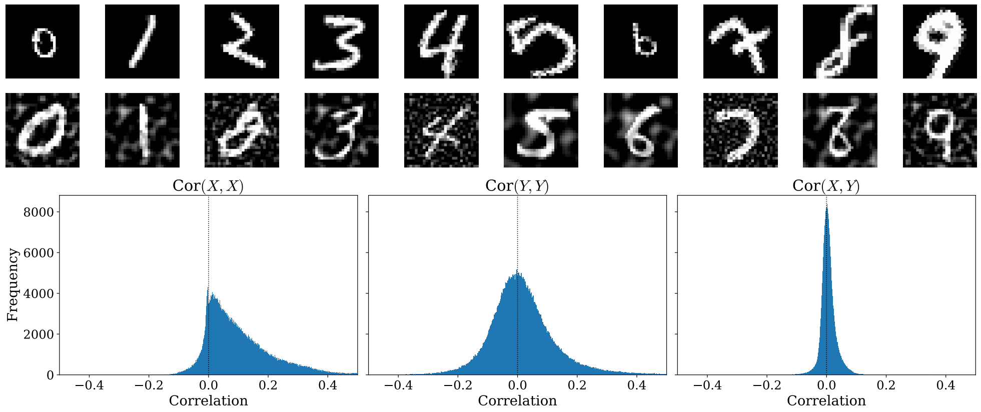


Figure 2.8: Dataset containing paired MNIST digit samples sharing only the same identity *(shared signal)*. The first row (X) shows MNIST digits randomly subjected to scaling, $(0.5 - 1.5)$, and rotation with an angle of $(0 - \pi/2)$, while the second row (Y) shows MNIST digits with an added background Perlin noise *(self signals)*. In the bottom row, histograms of self correlations for the X and Y datasets (left and middle, respectively) illustrate a wide range of correlations, while the histogram of the cross correlation between X and Y (right) demonstrates a smaller range.

To analyze linear DR methods on nonlinear data, we followed the same procedure as in Fig. 2.7 for a dataset inspired by the noisy MNIST dataset [94, 157, 158, 2]. This dataset has two distinct views of data, each of dimensionality 28×28 pixels, examples of which are shown in Fig. 2.8. The first view is an image of the digit subjected to a random rotation within an angle uniformly sampled between 0 and $\frac{\pi}{2}$, along with scaling by a factor uniformly distributed between 0.5 and 1.5. The second view consists of another image with the same digit identity with an additional background layer of Perlin noise [120], with the noise factor uniformly distributed between 0 and 1. Both views are normalized to an intensity range of $[0, 1)$, then flattened to form an array of 784 dimensions.

To cast this dataset into our language, we shuffled the images within labels, retaining the shared label identity (that is the shared signal), but we still have the view-specific details (which is the self signal). This resulted in a total dataset size of $\sim 56k$ images for training and $\sim 7k$ images for testing. The correlation histogram of X (or Y) with itself shows a relatively wide spectrum when compared to the cross correlation between X and Y, highlighting that the self signal is stronger, and can lead to different DR methods overseeing the shared one. The complexity of the tasks makes it sufficiently challenging, serving as a good benchmark for evaluating the performance of the different DR techniques.

2.5.5.2 Results

Figure 2.9 shows the performance of PCA, PLS, CCA, and rCCA applied to the modified Noisy MNIST dataset for varying sampling scenarios. The three panels are evaluated for different sample sizes (1000, $10,000$, and $\sim 56,000$ samples), from undersampled to the full dataset.

In each scenario, the training samples are used for the DR methods. Subsequently, the learned projection matrices onto the singular directions are used to transform a separate test dataset of around $7,000$ samples into low-dimensional spaces, yielding

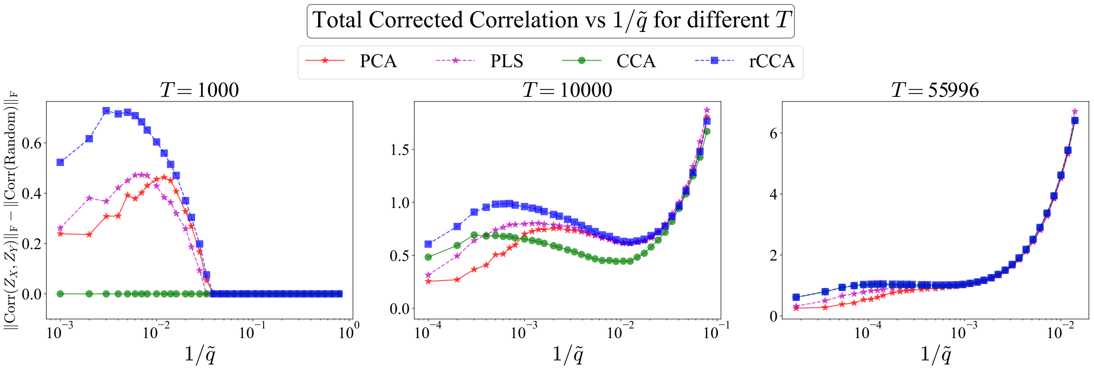


Figure 2.9: Performance of PCA, PLS, CCA, rCCA applied to the modified Noisy MNIST dataset across varying sampling scenarios. Each panel represents different sample sizes (1000, $10,000$, and approximately $56,000$ samples). The x-axis denotes the inverse of the number of samples per retained dimensions ($1/\tilde{q}$), while the y-axis represents the total corrected correlation between the obtained low-dimensional representations Z_X and Z_Y.

Z_X and Z_Y. The correlation between these transformed spaces is computed using the Frobenius norm of the correlation matrix. As before, we then subtracted from it the correlation value obtained from a random matrix of the same size. This difference is then plotted against $1/\tilde{q}$, which is the measure of how many dimensions are retained at each sampling ratio.

In the undersampled scenario (1000 samples), rCCA and PLS demonstrate an early detection (in terms of the number of kept dimensions after reduction) of shared signals, whereas PCA initially lags behind. As the number of dimensions increases, all methods exhibit a decline in correlation due to increased noise as we have fewer samples per dimension. CCA does not work in this scenario, since covariance matrices are degenerate.

Upon increasing the sample size ($10,000$ samples), a similar pattern emerges initially, where all methods experience an increase in total correlation till a certain number of kept dimensions is reached, then a decline when adding more dimensions. The decline is because one needs to estimate more singular vectors from the same number of samples. However, beyond a certain number of singular vectors, an increase in correlation is observed. This is because the number of vectors is now sufficient to learn both the shared and the self signals. We observe that rCCA maintains superior performance, while PCA reaches peak correlation at a higher number of kept dimensions, providing a rough estimation of the number of true self and shared signals. With the full dataset (approximately 56,000 samples), a similar trend is seen. Yet CCA's performance approaches that of rCCA.

Notably, the consistent superiority of Simultaneous Dimensionality Reduction (SDR) over Independent Dimensionality Reduction (IDR) is reaffirmed, emphasizing its effectiveness in detecting shared signals even in nonlinear datasets.

2.6 Discussions

2.6.1 Extensions and Generalizations

We used a generative linear model which captures multiple desired features of multimodal data with shared and non-shared signals. The model focused only on data with two measured modalities. However, while not a part of this study, the model can be readily extended to accommodate more than two modalities (e. g., $X_i = R_i + U_i V_i + P Q_i$ for $i = 1, ..., n$, where n represents the number of modalities). Then, methods such as Tensor CCA, which can handle more than two modalities [99], can be used to get insight into DR on such data.

2.6.2 Explaining Observations in the Literature

We analyzed different DR methods on data from this model in different parameter regimes. Linear SDR methods were clearly superior to their IDR counterparts for detecting shared signals. We observed similar results on a nonlinear dataset as well. We thus make a strong practical suggestion that, whenever the goal is to reconstruct a low dimensional representation of covariation between two components of the data, IDR methods (PCA) should always be avoided in favor of SDR. Of the examined SDR approaches, rCCA is a clear winner in all parameter regimes and should always be preferred. These findings explain the results of, for example, [130] and others that SDR can recover joint neuro-behavioral latent spaces with fewer latent dimensions and using fewer samples than IDR methods. Further, our observation that SDR is always superior to IDR in the context of our model corroborates the theoretical findings of [101], who proved a similar result in the context of discrete data and a different SDR algorithm, namely the Symmetric Information Bottleneck [50]. [153] made similar conclusions using conditional covariance matrices for the reduction in the context of classification. More recent work of [2] showed similar results using deep variational methods. Collectively, these diverse investigations, linear and nonlinear, theoretical, computational, and empirical, provide strong evidence that generic (not just linear) SDR methods are likely to be more efficient in extracting covariation than their IDR analogs.

2.6.3 Is SDR strictly effective in low sampling situations?

Our study answers an open question in the literature surrounding the effectiveness of SDR techniques. Specifically, there has been debate about whether PLS, an SDR method, is effective at low sampling [36, 61, 55, 56]. Our results show that SDR is not necessarily effective in the undersampled regime. It works well when the number of samples per retained dimension is high (even if the number of samples per observed dimension is low), but only when the dimensionality of the reduced description is matched to the actual dimensionality of the shared signals.

2.6.4 Diagnostic Test for number of latent signals

In addition to the previous, our results can be used as a diagnostic test to determine the number of shared versus self signals in data. As demonstrated in Fig. 2.7, total correlations between Z_X and Z_Y obtained by applying PCA and rCCA increase monotonically as the dimensionality of Zs increases, until this dimensionality becomes larger than the signal dimensionality. For PCA, the signal dimensionality is equal to the sum of the number of the shared and the self signals, $m_{\text{shared}} + m_{\text{self}}$. For rCCA, it is only the number of the shared signal. Thus increasing the dimensionality of the compressed variables and tracking the performance of rCCA and PCA until they diverge can be used to identify the number of self signals in the data, provided that the data, indeed, has a low-dimensional latent structure. This approach can be a valuable tool in various applications, where the characterization of shared and self signals in complex systems can provide insights into their structure and function.

2.6.5 Limitations, and Future Work

2.6.5.1 Linearity of the model

While this work has provided useful insight, the assumptions made here may not fully capture the complexity of real-world data. Specifically, our data is generated by a linear model with random Gaussian features. It is unlikely that real data have this exact structure. Therefore, there is a need for further exploration of the advantages and limitations of linear DR methods on data that have a low-dimensional, but nonlinear shared structure. This can be done using more complex nonlinear generative models, such as nonlinearly transforming the data generated by Eq. (2.1-2.2), or random feature two-layered neural network models [126]. Alternatively, analyzing the model, Eq. (2.1) using various theoretical techniques [23, 153, 123] is likely to offer even more insights into its properties. Collectively, these diverse approaches would aid our understanding of different DR methods under diverse conditions.

2.6.5.2 Linearity of the methods

A different possible future research direction is to explore the performance of nonlinear DR methods on data from generative models with a latent low-dimensional nonlinear structure. Autoencoders and their variational extensions are a natural extension of IDR to learn nonlinear reduced dimensional representations [68, 87, 66]. Meanwhile, Deep CCA and its variational extensions [7, 157, 32, 158] should be explored as a nonlinear version of SDR. Both of these types of methods can potentially capture more complex relationships between the modalities and improve the quality of the reduced representations, and while recent work suggests that [2], it is not clear if the SDR class of methods is always more efficient than the IDR one.

2.6.5.3 Linearity of the metric

Our analysis also depends on the choice of metric used to quantify the performance of DR, and different choices should also be explored. For example, to capture nonlinear correlations, mutual information can be utilized to quantify the relationships between the reduced representations.

2.6.6 Conclusion

In conclusion, we highlight a general principle that, when searching for a shared signal between different modalities of data, SDR methods are preferable to IDR methods. Additionally, the differences in performance between the two classes of methods can tell us a lot about the underlying structure of the data. Finally, for a limited number of samples, naive approaches, such as increasing the number of compressed dimensions indefinitely to overcome the masking of shared signals by self signals are infeasible. Thus, the use of SDR methods becomes even more essential in such cases, and despite the aforementioned limitations, we believe that our work provides a compelling addition to the body of knowledge that SDR outperforms IDR in detecting shared signals quite generally.

2.7 Limitations, and Future Work

While this work has provided useful insight, the assumptions made here may not fully capture the complexity of real-world data. Specifically, our data is generated by a linear model with random Gaussian features. It is unlikely that real data have this exact structure. Therefore, there is a need for further exploration of the advantages and limitations of linear DR methods on data that have a low-dimensional, but nonlinear shared structure. This can be done using more complex nonlinear generative models, such as nonlinearly transforming the data generated by Eq. (2.1-2.2), or random feature two-layered neural network models [126].

A different possible future research direction is to explore the performance of nonlinear DR methods on data from generative models with a latent low-dimensional nonlinear structure. Autoencoders and their variational extensions are a natural extension of IDR to learn nonlinear reduced dimensional representations [68, 87, 66]. Meanwhile, Deep CCA and its variational extensions [7, 157, 32, 158] should be explored as a nonlinear version of SDR. Both of these types of methods can potentially capture more complex relationships between the modalities and improve the quality of the reduced representations, and it is not clear if the SDR class of methods is always more efficient than the IDR one.

Further, our analysis depends on the choice of metric used to quantify the performance of DR, and different choices should also be explored. For example, to capture nonlinear correlations, mutual information can be utilized to quantify the relationships between the reduced representations.

Despite the aforementioned limitations, we believe that our work provides a compelling addition to the body of knowledge that SDR outperforms IDR in detecting shared signals quite generally.

Chapter 3

Deep Variational Multivariate Information Bottleneck Framework

3.1 Summary

[1]Variational dimensionality reduction methods are known for their high accuracy, generative abilities, and robustness. We introduce a framework to unify many existing variational methods and design new ones. The framework is based on an interpretation of the multivariate information bottleneck, in which an encoder graph, specifying what information to compress, is traded-off against a decoder graph, specifying a generative model. Using this framework, we rederive existing dimensionality reduction methods including the deep variational information bottleneck and variational auto-encoders. The framework naturally introduces a trade-off parameter extending the deep variational CCA (DVCCA) family of algorithms to beta-DVCCA. We derive a new method, the deep variational symmetric informational bottleneck (DVSIB),

[1] This chapter presents the paper [2] with the title *Deep Variational Multivariate Information Bottleneck – A Framework for Variational Losses*. This work was conducted in collaboration with K. Michael Martini and Ilya Nemenman. Michael and I contributed equally to all the analytics, coding, result production, and manuscript writing. All authors contributed to conceiving the framework and reviewed the manuscript.

which simultaneously compresses two variables to preserve information between their compressed representations. We implement these algorithms and evaluate their ability to produce shared low dimensional latent spaces on Noisy MNIST dataset. We show that algorithms that are better matched to the structure of the data (in our case, beta-DVCCA and DVSIB) produce better latent spaces as measured by classification accuracy, dimensionality of the latent variables, and sample efficiency. We believe that this framework can be used to unify other multi-view representation learning algorithms and to derive and implement novel problem-specific loss functions.

3.2 Introduction

Large dimensional multi-modal datasets are abundant in multimedia systems utilized for language modeling [163, 168, 127, 129, 60, 155, 65], neural control of behavior studies [139, 150, 92, 114], multi-omics approaches in systems biology [37, 166, 144, 80, 97], and many other domains. Such data come with the curse of dimensionality, making it hard to learn the relevant statistical correlations from samples. The problem is made even harder by the data often containing information that is irrelevant to the specific questions one asks. To tackle these challenges, a myriad of dimensionality reduction (DR) methods have emerged. By preserving certain aspects of the data while discarding the remainder, DR can decrease the complexity of the problem, yield clearer insights, and provide a foundation for more refined modeling approaches.

DR techniques span linear methods like Principal Component Analysis (PCA) [75], Partial Least Squares (PLS) [161], Canonical Correlations Analysis (CCA) [76], and regularized CCA [152, 171], as well as nonlinear approaches, including Autoencoders (AE) [68], Deep CCA [7], Deep Canonical Correlated AE [157], Correlational Neural Networks [32], Deep Generalized CCA [18], and Deep Tensor CCA [162]. Of particular interest to us are variational methods, such as Variational Autoencoders (VAE)

[87], beta-VAE [66], Joint Multimodal VAE (JMVAE) [143], Deep Variational CCA (DVCCA) [158], Deep Variational Information Bottleneck (DVIB) [5], Variational Mixture-of-experts AE [134], and Multiview Information Bottleneck [46]. These DR methods use deep neural networks and variational approximations to learn robust and accurate representations of the data, while, at the same time, often serving as generative models for creating samples from the learned distributions.

There are many theoretical derivations and justifications for variational DR methods [87, 66, 143, 158, 84, 124, 5, 14, 96, 156, 154, 45, 79, 78]. This diversity of derivations, while enabling adaptability, often leaves researchers with no principled ways for choosing a method for a particular application, for designing new methods with distinct assumptions, or for comparing methods to each other.

Here, we introduce the Deep Variational Multivariate Information Bottleneck (DVMIB) framework, offering a unified mathematical foundation for many variational DR methods. Our framework is grounded in the multivariate information bottleneck loss function [147, 50]. This loss, amenable to approximation through upper and lower variational bounds, provides a system for implementing diverse DR variants using deep neural networks. We demonstrate the framework's efficacy by deriving the loss functions of many existing variational DR methods starting from the same principles. Furthermore, our framework naturally allows the adjustment of trade-off parameters, leading to generalizations of these existing methods. For instance, we generalize DVCCA to β-DVCCA. The framework further allows us to introduce and implement in software novel DR methods. We view the DVMIB framework, with its uniform information bottleneck language, conceptual clarity of translating statistical dependencies in data via graphical models of encoder and decoder structures into variational losses, the ability to unify existing approaches, and easy adaptability to new scenarios as one of the main contributions of our work.

Beyond its unifying role, our framework offers a principled approach for deriving problem-specific loss functions using domain-specific knowledge. Thus, we anticipate its application for multi-view representation learning across diverse fields. To illustrate this, we use the framework to derive a novel dimensionality reduction method, the Deep Variational Symmetric Information Bottleneck (DVSIB), which compresses two random variables into two distinct latent variables that are maximally informative about one another. This new method produces better representations of classic datasets than previous approaches. The introduction of DVSIB is another major contribution of our work.

In summary, this chapter makes the following contributions to the field:

1. **Introduction of the Variational Multivariate Information Bottleneck Framework:** We provide both intuitive and mathematical insights into this framework, establishing a robust foundation for further exploration.

2. **Rederivation and Generalization of Existing Methods within a Common Framework:** We demonstrate the versatility of our framework by systematically rederiving and generalizing various existing methods from the literature, showcasing the framework's ability to unify diverse approaches.

3. **Design of a Novel Method — Deep Variational Symmetric Information Bottleneck (DVSIB):** Employing our framework, we introduce DVSIB as a new method, contributing to the growing repertoire of techniques in variational dimensionality reduction. The method constructs high-accuracy latent spaces from substantially fewer samples than comparable approaches.

The chapter is structured as follows. First, we introduce the underlying mathematics and the implementation of the DVMIB framework. We then explain how to use the framework to generate new DR methods. In Tbl. 3.1, we present several known and newly derived variational methods, illustrating how easily they can be derived within the framework. As a proof of concept, we then benchmark *simple* computational implementations of methods in Tbl. 3.1 against the Noisy MNIST dataset. Appendices present detailed treatment of all terms in variational losses in our framework, discussion of multi-view generalizations, and more details —including visualizations— of the performance of many methods on the Noisy MNIST.

3.3 Multivariate Information Bottleneck Framework

We represent DR problems similar to the Multivariate Information Bottleneck (MIB) of Friedman et al. [50], which is a generalization of the more traditional Information Bottleneck algorithm [147] to multiple variables. The reduced representation is achieved as a trade-off between two Bayesian networks. Bayesian networks are directed acyclic graphs that provide a factorization of the joint probability distribution, $P(X_1, X_2, X_3, .., X_N) = \prod_{i=1}^{N} P(X_i|Pa^G_{X_i})$, where $Pa^G_{X_i}$ is the set of parents of X_i in graph G. The multiinformation [142] of a Bayesian network is defined as the Kullback-Leibler divergence between the joint probability distribution and the product of the marginals, and it serves as a measure of the total correlations among the variables, $I(X_1, X_2, X_3, ..., X_N) = D_{KL}(P(X_1, X_2, X_3, ..., X_N)\|P(X_1)P(X_2)P(X_3)...P(X_N))$. For a Bayesian network, the multiinformation reduces to the sum of all the local informations $I(X_1, X_2, ..X_N) = \sum_{i=1}^{N} I(X_i; Pa^G_{X_i})$ [50].

The first of the Bayesian networks is an encoder (compression) graph, which models how compressed (reduced, latent) variables are obtained from the observations. The second network is a decoder graph, which specifies a generative model for the data from the compressed variables, i.e., it is an alternate factorization of the distribution. In MIB, the information of the encoder graph is minimized, ensuring strong compression (corresponding to the approximate posterior). The information of the decoder graph is maximized, promoting the most accurate model of the data (corresponding to maximizing the log-likelihood). As in IB [147], the trade-off between the compression and reconstruction is controlled by a trade-off parameter β:

$$L = I_{\text{encoder}} - \beta I_{\text{decoder}}. \tag{3.1}$$

In this work, our key contribution is in writing an explicit variational loss for typical information terms found in both the encoder and the decoder graphs. All terms in the decoder graph use samples of the compressed variables as determined from the encoder graph. If there are two terms that correspond to the same information in Eq. (3.1), one from each of the graphs, they do not cancel each other since they correspond to two different variational expressions. For pedagogical clarity, we do this by first analyzing the Symmetric Information Bottleneck (SIB), a *special case* of MIB. We derive the bounds for three types of information terms in SIB, which we then use as building blocks for all other variational MIB methods in subsequent Sections.

3.3.1 Deep Variational Symmetric Information Bottleneck

The Deep Variational Symmetric Information Bottleneck (DVSIB) simultaneously reduces a pair of datasets X and Y into two separate lower dimensional compressed versions Z_X and Z_Y. These compressions are done at the same time to ensure that the latent spaces are maximally informative about each other. The joint compression is known to decrease dataset size requirements compared to individual ones [101]. Having distinct latent spaces for each modality usually helps with interpretability. For example, X could be the neural activity of thousands of neurons, and Y could be the recordings of joint angles of the animal. Rather than one latent space representing both, separate latent spaces for the neural activity and the joint angles are sought. By maximizing compression as well as $I(Z_X, Z_Y)$, one constructs the latent spaces that capture only the neural activity pertinent to joint movement and only the movement that is correlated with the neural activity (cf. [114]). Many other applications could benefit from a similar DR approach.

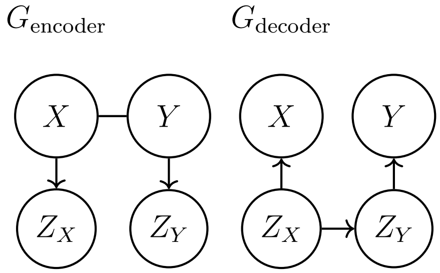


Figure 3.1: The encoder and decoder graphs for DVSIB.

In Fig. 3.1, we define two Bayesian networks for DVSIB, G_{encoder} and G_{decoder}. G_{encoder} encodes the compression of X to Z_X and Y to Z_Y. It corresponds to the factorization $p(x, y, z_x, z_y) = p(x, y)p(z_x|x)p(z_y|y)$ and the resultant $I_{\text{encoder}} = I^E(X;Y)+I^E(X;Z_X)+I^E(Y;Z_Y)$. The $I^E(X,Y)$ term does not depend on the compressed variables, does not affect the optimization problem, and hence is discarded in what follows. G_{decoder} represents a generative model for X and Y given the compressed latent variables Z_X and Z_Y. It corresponds to the factorization $p(x, y, z_x, z_y) = p(z_x)p(z_y|z_x)p(x|z_x)p(y|z_y)$ and the resultant $I_{\text{decoder}} = I^D(Z_X;Z_Y) + I^D(X;Z_X) + I^D(Y;Z_Y)$. Combing the informations from both graphs and using Eq. (3.1), we find the SIB loss:

$$L_{\text{SIB}} = I^E(X;Z_X) + I^E(Y;Z_Y) - \beta \left(I^D(Z_X;Z_Y) + I^D(X;Z_X) + I^D(Y;Z_Y) \right). \quad (3.2)$$

Note that information in the encoder terms is minimized, and information in the decoder terms is maximized. Thus, while it is tempting to simplify Eq. (3.2) by canceling $I^E(X;Z_X)$ and $I^D(X;Z_X)$, this would be a mistake. Indeed, these terms come from different factorizations: the encoder corresponds to learning $p(z_x|x)$, and the decoder to $p(x|z_x)$.

While the DVSIB loss may appear similar to previous models, such as MultiView Information Bottleneck (MVIB) [46] and Barlow Twins [164], it is distinct both conceptually and in practice. For example, MVIB aims to generate latent variables that are as similar to each other as possible, sharing the same domain. DVSIB, however, endeavors to produce distinct latent representations, which could potentially have different units or dimensions, while maximizing mutual information between them. Barlow Twins architecture on the other hand appears to have two latent subspaces while in fact they are one latent subspace that is being optimized by a regular information bottleneck.

We now follow a procedure and notation similar to Alemi et al. [5] and construct variational bounds on all I^E and I^D terms. Terms without leaf nodes, i. e., $I^D(Z_X, Z_Y)$, require new approaches.

3.3.2 Variational Bounds on DVSIB Encoder Terms

The information $I^E(Z_X; X)$ corresponds to compressing the random variable X to Z_X. Since this is an encoder term, it needs to be minimized in Eq. (3.2). Thus, we seek a variational bound $I^E(Z_X; X) \leq \tilde{I}^E(Z_X; X)$, where $\tilde{I}^E$ is the variational version of I^E, which can be implemented using a deep neural network. We find $\tilde{I}^E$ by using the positivity of the Kullback–Leibler divergence. We make $r(z_x)$ be a variational approximation to $p(z_x)$. Then $D_{\mathrm{KL}}(p(z_x)\|r(z_x)) \geq 0$, so that $-\int dz_x p(z_x)\ln(p(z_x)) \leq -\int dz_x p(z_x)\ln(r(z_x))$. Thus, $-\int dx dz_x p(z_x, x)\ln(p(z_x)) \leq -\int dx dz_x p(z_x, x)\ln(r(z_x))$. We then add $\int dx dz_x p(z_x, x)\ln(p(z_x|x))$ to both sides and find:

$$\begin{aligned} I^E(Z_X; X) &= \int dx dz_x p(z_x, x) \ln\left(\frac{p(z_x|x)}{p(z_x)}\right) \\ &\leq \int dx dz_x p(z_x, x) \ln\left(\frac{p(z_x|x)}{r(z_x)}\right) \equiv \tilde{I}^E(Z_X; X). \end{aligned} \tag{3.3}$$

We further simplify the variational loss by approximating $p(x) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(x-x_i)$, so that:

$$\tilde{I}^E(Z_X;X) \approx \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln\left(\frac{p(z_x|x_i)}{r(z_x)}\right) = \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(z_x|x_i)\|r(z_x)). \tag{3.4}$$

The term $I^E(Y;Z_Y)$ can be treated in an analogous manner, resulting in:

$$\tilde{I}^E(Z_Y;Y) \approx \frac{1}{N}\sum_{i=1}^{N} D_{KL}(p(z_y|y_i)\|r(z_y)). \tag{3.5}$$

3.3.3 Variational Bounds on DVSIB Decoder Terms

The term $I^D(X;Z)$ corresponds to a decoder of X from the compressed variable Z_X. It is maximized in Eq. (3.2). Thus, we seek its variational version $\tilde{I}^D$, such that $I^D \geq \tilde{I}^D$. Here, $q(x|z_x)$ will serve as a variational approximation to $p(x|z_x)$. We use the positivity of the Kullback-Leibler divergence, $D_{\mathrm{KL}}(p(x|z_x)\|q(x|z_x)) \geq 0$, to find $\int dx\, p(x|z_x)\ln(p(x|z_x)) \geq \int dx\, p(x|z_x)\ln(q(x|z_x))$. This gives $\int dz_x dx\, p(x,z_x)\ln(p(x|z_x))$ $\geq \int dz_x dx\, p(x,z_x)\ln(q(x|z_x))$. We add the entropy of X to both sides to arrive at the variational bound:

$$\begin{aligned} I^D(X;Z_X) &= \int dz_x dx\, p(x,z_x)\ln\frac{p(x|z_x)}{p(x)} \\ &\geq \int dz_x dx p(x,z_x)\ln\frac{q(x|z_x)}{p(x)} \equiv \tilde{I}^D(X;Z_X). \end{aligned} \tag{3.6}$$

We further simplify $\tilde{I}^D$ by replacing $p(x)$ by samples, $p(x) \approx \frac{1}{N}\sum_i^N \delta(x-x_i)$ and using the $p(z_x|x)$ that we learned previously from the encoder:

$$\tilde{I}^D(X;Z_X) \approx H(X) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(x_i|z_x)). \tag{3.7}$$

Here $H(X)$ does not depend on $p(z_x|x)$ and, therefore, can be dropped from the loss.

The variational version of $I^D(Y;Z_Y)$ is obtained analogously:

$$\tilde{I}^D(Y;Z_Y) \approx H(Y) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_y|y_i)\ln(q(y_i|z_y)). \tag{3.8}$$

3.3.4 Variational Bounds on Decoder Terms not on a Leaf - MINE

The variational bound above cannot be applied to the information terms that do not contain leaves in G_{decoder}. For SIB, this corresponds to the $I^D(Z_X, Z_Y)$ term. This information is maximized. To find a variational bound such that $I^D(Z_X, Z_Y) \geq \tilde{I}^D(Z_X, Z_Y)$, we use the MINE mutual information estimator [17], which samples both Z_X and Z_Y from their respective variational encoders. Other mutual information estimators, such as I_{InfoNCE} [121], can be used as long as they are differentiable[2]. Other estimators might be better suited for different problems, but for our current application, I_{MINE} was sufficient. We variationally approximate $p(z_x, z_y)$ as $p(z_x)p(z_y)e^{T(z_x,z_y)}/\mathcal{Z}_{\text{norm}}$, where $\mathcal{Z}_{\text{norm}} = \int dz_x dz_y p(z_x)p(z_y)e^{T(z_x,z_y)}$ is the normalization factor. Here $T(z_x, z_y)$ is parameterized by a neural network that takes in samples of the latent spaces z_x and z_y and returns a single number. We again use the positivity of the Kullback-Leibler divergence, $D_{\text{KL}}(p(z_x, z_y)\|p(z_x)p(z_y)e^{T(z_x,z_y)}/\mathcal{Z}_{\text{norm}}) \geq 0$, which implies $\int dz_x dz_y p(z_x, z_y)\ln(p(z_x, z_y)) \geq \int dz_x dz_y p(z_x, z_y)\ln\frac{p(z_x)p(z_y)e^{T(z_x,z_y)}}{\mathcal{Z}_{\text{norm}}}$. Subtracting $\int dz_x dz_y p(z_x, z_y)\ln(p(z_x)p(z_y))$ from both sides, we find:

$$I^D(Z_X;Z_Y) \geq \int dz_x dz_y p(z_x, z_y)\ln\frac{e^{T(z_x,z_y)}}{\mathcal{Z}_{\text{norm}}} \equiv \tilde{I}^D_{\text{MINE}}(Z_X;Z_Y). \tag{3.9}$$

[2] Further details and discussions for different estimators are presented in Chapter 4.

3.3.5 Parameterizing the Distributions and the Reparameterization Trick

$H(X)$, $H(Y)$, and $I(X, Y)$ do not depend on $p(z_x|x)$ and $p(z_y|y)$ and are dropped from the loss. Further, we can use any ansatz for the variational distributions we introduced. We choose parametric probability distribution families and learn the nearest distribution in these families consistent with the data. We assume $p(z_x|x)$ is a normal distribution with mean $\mu_{Z_X}(x)$ and a diagonal variance $\Sigma_{Z_X}(x)$. We learn the mean and the log variance as neural networks. We also assume that $q(x|z_x)$ is normal with a mean $\mu_X(z_x)$ and a unit variance. In principle, we could also learn the variance for this distribution, but practically we did not find the need for that, and the approach works well as is. Finally, we assume that $r(z_x)$ is a standard normal distribution. We use the reparameterization trick to produce samples of $z_{xi,j} = z_{x_j}(x_i) = \mu(x_i) + \sqrt{\Sigma_{Z_X}(x_i)}\eta_j$ from $p(z_x|x_i)$, where η_j is drawn from a standard normal distribution [87]. We choose the same types of distributions for the corresponding z_y terms.

To sample from $p(z_x, z_y)$ we use $p(z_x, z_y) = \int dxdy\, p(z_x, z_y, x, y) = \int dxdy\, p(z_x|x)p(z_y|y) \times p(x, y) \approx \frac{1}{N}\sum_{i=1}^{N} p(z_x|x_i)p(z_y|y_i) = \frac{1}{NM^2}\sum_{i=1}^{N}(\sum_{j=1}^{M} \delta(z_x - z_{xi,j}))(\sum_{j=1}^{M} \delta(z_y - z_{y_{i,j}}))$, where $z_{xi,j} \in p(z_x|x_i)$ and $z_{y_{i,j}} \in p(z_y|y_i)$, and M is the number of new samples being generated. To sample from $p(z_x)p(z_y)$, we generate samples from $p(z_x, z_y)$ and scramble the generated entries z_x and z_y, destroying all correlations. With this, the components of the loss function become

$$\tilde{I}^E(X;Z_X) \approx \frac{1}{2N}\sum_{i=1}^{N}\left[\mathrm{Tr}(\Sigma_{Z_X}(x_i)) + ||\vec{\mu}_{Z_X}(x_i)||^2 - k_{Z_X} - \ln\det(\Sigma_{Z_X}(x_i))\right], \tag{3.10}$$

$$\tilde{I}^D(X;Z_X) \approx \frac{1}{MN}\sum_{i,j=1}^{N,M} -\frac{1}{2}||(x_i - \mu_X(z_{xi,j}))||^2, \tag{3.11}$$

$$\tilde{I}^D_{\mathrm{MINE}}(Z_X;Z_Y) \approx \frac{1}{M^2N}\sum_{i,j_x,j_y=1}^{N,M,M}\left[T(z_{xi,j_x}, z_{y_{i,j_y}}) - \ln \mathcal{Z}_{\mathrm{norm}}\right], \tag{3.12}$$

where $\mathcal{Z}_{\mathrm{norm}} = \mathbb{E}_{z_x\sim p(z_x), z_y\sim p(z_y)}[e^{T(z_x,z_y)}]$, k_{Z_X} is the dimension of Z_X, and the corresponding terms for Y are similar. Combining these terms results in the variational loss for DVSIB:

$$L_{\mathrm{DVSIB}} = \tilde{I}^E(X;Z_X) + \tilde{I}^E(Y;Z_Y) - \beta\left(\tilde{I}^D_{\mathrm{MINE}}(Z_X;Z_Y) + \tilde{I}^D(X;Z_X) + \tilde{I}^D(Y;Z_Y)\right). \tag{3.13}$$

3.4 Deriving Other DR Methods

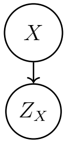
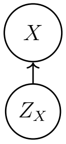
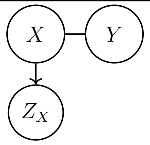
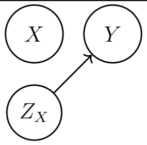
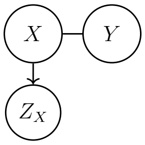
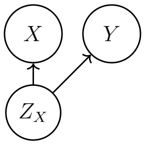
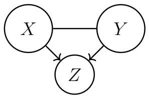
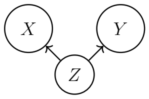
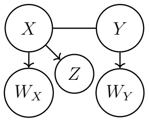
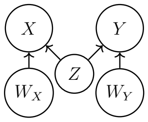
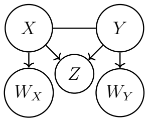
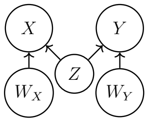
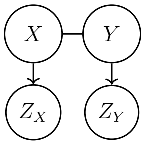
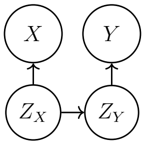
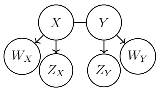
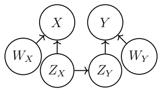
The variational bounds used in DVSIB can be used to implement loss functions that correspond to other encoder-decoder graph pairs and hence to other DR algorithms. The simplest is the beta variational auto-encoder. Here G_{encoder} consists of one term: X compressed into Z_X. Similarly G_{decoder} consists of one term: X decoded from Z_X (see Table 3.1). Using this simple set of Bayesian networks, we find the variational loss:

$$L_{\text{beta-VAE}} = \tilde{I}^E(X;Z_X) - \beta\tilde{I}^D(X;Z_X). \tag{3.14}$$

Both terms in Eq. (3.14) are the same as Eqs. (3.10, 3.11) and can be approximated and implemented by neural networks.

Similarly, we can re-derive the DVCCA family of losses [158]. Here G_{encoder} is X

Table 3.1: Method descriptions, variational losses, and the Bayesian Network graphs for each DR method derived in our framework. See Appendix 3.7.1 for details. For methods where we can reduce either X or Y, only X graphs/loss are shown

Method Description	$G_{\textbf{encoder}}$	$G_{\textbf{decoder}}$
beta-VAE [87, 66]: Two independent Variational Autoencoder (VAE) models trained, one for each view, X and Y. $L_{\text{VAE}} = \tilde{I}^E(X; Z_X) - \beta \tilde{I}^D(X; Z_X)$		
DVIB [5]: Two bottleneck models trained, one for each view, X and Y, using the other view as the supervising signal. $L_{\text{DVIB}} = \tilde{I}^E(X; Z_X) - \beta \tilde{I}^D(Y; Z_X)$		
beta-DVCCA: Similar to DVIB [5], but with reconstruction of both views. Two models trained, compressing either X or Y, while reconstructing both X and Y. $L_{\text{DVCCA}} = \tilde{I}^E(X; Z_X) - \beta(\tilde{I}^D(Y; Z_X) + \tilde{I}^D(X; Z_X))$ **DVCCA** [158]: β-DVCCA with $\beta = 1$.		
beta-joint-DVCCA: A single model trained using a concatenated variable $[X, Y]$, learning one latent representation Z. $L_{\text{jDVCCA}} = \tilde{I}^E((X, Y); Z) - \beta(\tilde{I}^D(Y; Z) + \tilde{I}^D(X; Z))$ **joint-DVCCA** [158]: β-jDVCCA with $\beta = 1$.		
beta-DVCCA-private: Two models trained, compressing either X or Y, while reconstructing both X and Y, and simultaneously learning private information W_X and W_Y $L_{\text{DVCCA-p}} = \tilde{I}^E(X; Z) + \tilde{I}^E(X; W_X) + \tilde{I}^E(Y; W_Y) - \beta(\tilde{I}^D(X; (W_X, Z)) + \tilde{I}^D(Y; (W_Y, Z)))$ **DVCCA-private** [158]: β-DVCCA-p with $\beta = 1$.		
beta-joint-DVCCA-private: A single model trained using a concatenated variable $[X, Y]$, learning one latent representation Z, and simultaneously learning private information W_X and W_Y. $L_{\text{jDVCCA-p}} = \tilde{I}^E((X, Y); Z) + \tilde{I}^E(X; W_X) + \tilde{I}^E(Y; W_Y) - \beta(\tilde{I}^D(X; (W_X, Z)) + \tilde{I}^D(Y; (W_Y, Z)))$ **joint-DVCCA-private**[158]: β-jDVCCA-p with $\beta = 1$.		
DVSIB: A symmetric model trained, producing Z_X and Z_Y. $L_{\text{DVSIB}} = \tilde{I}^E(X; Z_X) + \tilde{I}^E(Y; Z_Y)$ $-\beta \left(\tilde{I}^D_{\text{MINE}}(Z_X; Z_Y) + \tilde{I}^D(X; Z_X) + \tilde{I}^D(Y; Z_Y)\right)$		
DVSIB-private: A symmetric model trained, producing Z_X and Z_Y, while simultaneously learning private information W_X and W_Y. $L_{\text{DVSIBp}} = \tilde{I}^E(X; W_X) + \tilde{I}^E(X; Z_X) +$ $\tilde{I}^E(Y; Z_Y) + \tilde{I}^E(Y; W_Y) -$ $\beta \left(\tilde{I}^D_{\text{MINE}}(Z_X; Z_Y) + \tilde{I}^D(X; (Z_X, W_X)) + \tilde{I}^D(Y; (Z_Y, W_Y))\right)$		

compressed into Z_X. G_{decoder} reconstructs both X and Y from the same compressed latent space Z_X. In fact, our loss function is more general than the DVCCA loss and has an additional compression-reconstruction trade-off parameter β. We call this more general loss β-DVCCA, and the original DVCCA emerges when $\beta = 1$:

$$L_{\text{DVCCA}} = \tilde{I}^E(X; Z_X) - \beta(\tilde{I}^D(Y; Z_X) + \tilde{I}^D(X; Z_X)). \tag{3.15}$$

Using the same library of terms as we found in DVSIB, Eqs. (3.10, 3.11), we find:

$$\begin{aligned} L_{\text{DVCCA}} \approx & \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_x|x_i)\|r(z_x)) \\ - \beta & \left(\frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(y_i|z_x)) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(x_i|z_x)) \right). \end{aligned} \tag{3.16}$$

This is similar to the loss function of the deep variational CCA [158], but now it has a trade-off parameter β. It trades off the compression into Z_X against the reconstruction of X and Y from the compressed variable Z_X.

Table 3.1 shows how our framework reproduces and generalizes other DR losses (see Appendix 3.7.1). Our framework naturally extends beyond two variables as well (see Appendix 3.7.2).

3.5 Results

To test our methods, we created a dataset inspired by the noisy MNIST dataset [94, 157, 158], consisting of two distinct views of data, both with dimensions of 28×28 pixels, cf. Fig. 3.2. The first view comprises the original image randomly rotated by an angle uniformly sampled between 0 and $\frac{\pi}{2}$ and scaled by a factor uniformly distributed between 0.5 and 1.5. The second view consists of the original image with an added background Perlin noise [120] with the noise factor uniformly

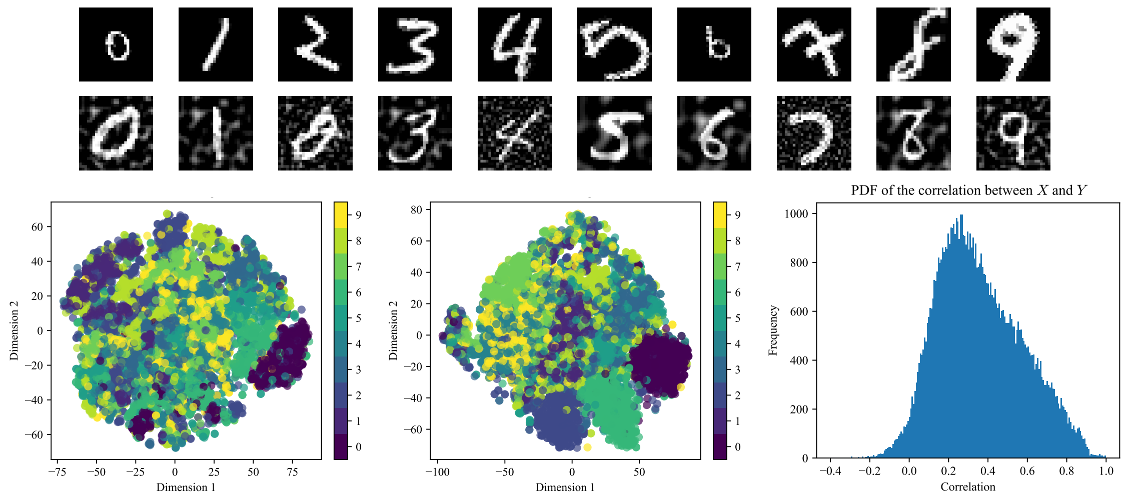


Figure 3.2: Dataset consisting of pairs of digits drawn from MNIST that share an identity. Top row, X: MNIST digits randomly scaled $(0.5 - 1.5)$ and rotated $(0 - \pi/2)$. Bottom row, Y: MNIST digits with a background Perlin noise. t-SNE of X and Y datasets (left and middle) shows poor separation by digit, and there is a wide range of correlation between X and Y (right).

distributed between 0 and 1. Both image intensities are scaled to the range of $[0, 1)$. The dataset was shuffled within labels, retaining only the shared label identity between two images, while disregarding the view-specific details, i.e., the random rotation and scaling for X, and the correlated background noise for Y. The dataset, totaling $70,000$ images, was partitioned into training (80%), testing (10%), and validation (10%) subsets. Visualization via t-SNE [67] plots of the original dataset suggest poor separation by digit, and the two digit views have diverse correlations, making this a sufficiently hard problem.

The DR methods we evaluated include all methods from Tbl. 3.1. PCA and CCA [75, 76] served as a baseline for linear dimensionality reduction. Multi-view Information Bottleneck [46] was included for a specific comparison with DVSIB (see Appendix 3.7.3). We emphasize that none of the algorithms were given labeled data. They had to infer compressed latent representations that presumably should cluster into ten different digits based simply on the fact that images come in pairs, and the (unknown) digit label is the only information that relates the two images.

Each method was trained for 100 epochs using fully connected neural networks with layer sizes $(\text{input\_dim}, 1024, 1024, (k_Z, k_Z))$, where k_Z is the latent dimension size, employing ReLU activations for the hidden layers. The input dimension (input_dim) was either the size of X (784) or the size of the concatenated $[X, Y]$ (1568). The last two layers of size k_Z represented the means and log(variance) learned. For the decoders, we employed regular decoders, fully connected neural networks with layer sizes $(k_Z, 1024, 1024, \text{output\_dim})$, using ReLU activations for the hidden layers and sigmoid activation for the output layer. Again, the output dimension (output_dim) could either be the size of X (784) or the size of the concatenated $[X, Y]$ (1568). The latent dimension (k_Z) could be k_{Z_X} or k_{Z_Y} for regular decoders, or $k_{Z_X} + k_{W_X}$ or $k_{Z_Y} + k_{W_Y}$ for decoders with private information. Additionally, another decoder denoted as decoder_MINE, based on the MINE estimator for estimating $I(Z_X, Z_Y)$, was used in DVSIB and DVSIB with private information. The decoder_MINE is a fully connected neural network with layer sizes $(k_{Z_X} + k_{Z_Y}, 1024, 1024, 1)$ and ReLU activations for the hidden layers. Optimization was conducted using the ADAM optimizer with default parameters.

To evaluate the methods, we trained them on the training portions of X and Y without exposure to the true labels. Subsequently, we utilized the trained encoders to compute Z_{train}, Z_{test}, and $Z_{\text{validation}}$ on the respective datasets. To assess the quality of the learned representations, we revealed the labels of Z_{train} and trained a linear SVM classifier with Z_{train} and $\text{labels}_{\text{train}}$. Fine-tuning of the classifier was performed to identify the optimal SVM slack parameter (C value), maximizing accuracy on Z_{test}. This best classifier was then used to predict $Z_{\text{validation}}$, yielding the reported accuracy. We also conducted classification experiments using fully connected neural networks, with detailed results available in the Appendix 3.7.4. For both SVM and the fully connected network, we find the baseline accuracy on the original training data and

Table 3.2: Maximum accuracy from a linear SVM and the optimal k_Z and β for variational DR methods reported on the Y (above the line) and the joint $[X, Y]$ (below the line) datasets. († fixed values)

Method	**Acc. %**	$\boldsymbol{k_{Z\mathbf{best}}}$	95% $\boldsymbol{k_{Z}}_{\text{range}}$	$\boldsymbol{\beta}_{\text{best}}$	95% $\beta_{\mathbf{range}}$	$\boldsymbol{C}_{\text{best}}$
Baseline	90.8	784†	-	-	-	0.1
PCA	90.5	256	[64,256*]	-	-	1
CCA	85.7	256	[32,256*]	-	-	10
β-VAE	96.3	256	[64,256*]	32	[2,1024*]	10
DVIB	90.4	256	[16,256*]	512	[8,1024*]	0.003
DVCCA	89.6	128	[16,256*]	1†	-	31.623
β-DVCCA	95.4	256	[64,256*]	16	[2,1024*]	10
DVCCA-p	92.1	16	[16,256*]	1†	-	0.316
β-DVCCA-p	95.5	16	[**4**,256*]	1024	[1,1024*]	0.316
MVIB	**97.7**	8	[**4**,64]	1024	[128,1024*]	0.01
DVSIB	**97.8**	256	[**8**,256*]	128	[2,1024*]	3.162
DVSIB-p	**97.8**	256	[**8**,256*]	32	[2,1024*]	10
jBaseline	91.9	1568†	-	-	-	0.003
jDVCCA	92.5	256	[64,265*]	1†	-	10
β-jDVCCA	96.7	256	[16,265*]	256	[1,1024*]	1
jDVCCA-p	92.5	64	[32,265*]	1†	-	10
β-jDVCCA-p	92.7	256	[**4**,265*]	2	[1,1024*]	10

labels $(X_{\text{train}}, \text{labels}_{\text{train}})$ and $(Y_{\text{train}}, \text{labels}_{\text{train}})$, fine-tuning with the test datasets, and reporting the results of the validation datasets. Using Linear SVM enables us to assess the linear separability of the clusters of Z_X and Z_Y obtained through the DR methods. While neural networks excel at uncovering nonlinear relationships that could result in higher classification accuracy, the comparison with a linear SVM establishes a level playing field. It ensures a fair comparison among different methods and is independent of the success of the classifier used for comparison in detecting nonlinear features in the data, which might have been missed by the DR methods. Here, we focus on the results of the Y datasets (MNIST with correlated noise background); results for X are in the Appendix 3.7.4. A parameter sweep was performed to identify optimal k_Z values, ranging from 2^1 to 2^8 dimensions on $\log_2$ scale, as well as optimal β values, ranging from 2^{-5} to 2^{10}. For methods with private information, k_{W_X} and k_{W_Y} were

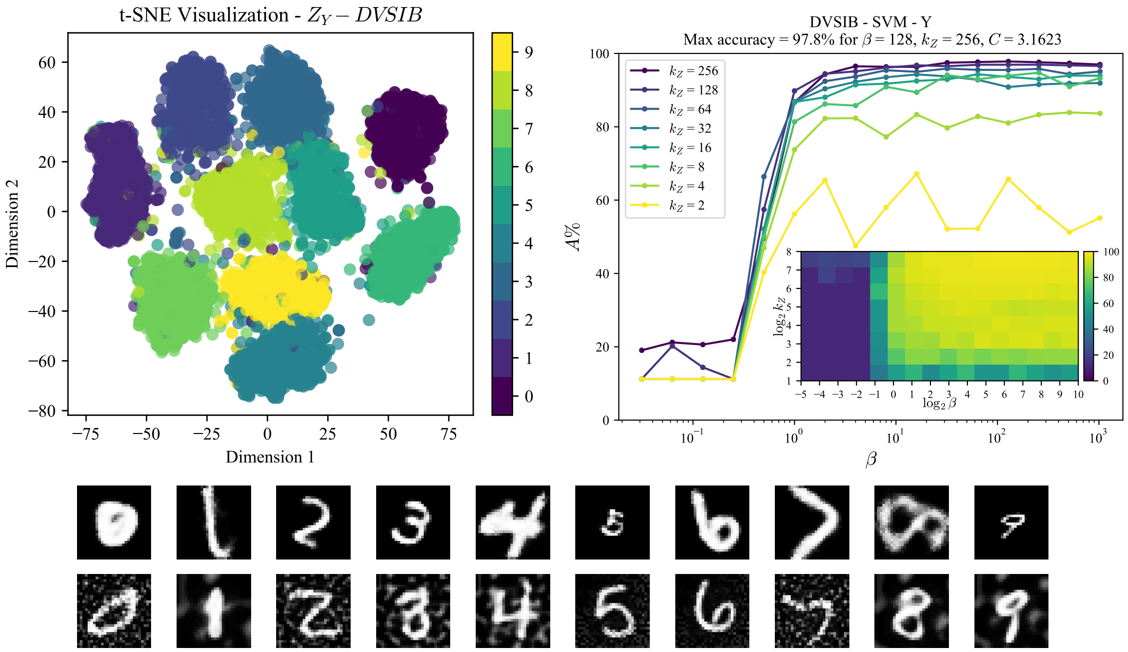


Figure 3.3: Top: t-SNE plot of the latent space Z_Y of DVSIB colored by the identity of digits. Top Right: Classification accuracy of an SVM trained on DVSIB's Z_Y latent space. The accuracy was evaluated for DVSIB with a parameter sweep of the trade-off parameter $\beta = 2^{-5}, ..., 2^{10}$ and the latent dimension $k_Z = 2^1, ..., 2^8$. The max accuracy was 97.8% for $\beta = 128$ and $k_Z = 256$. Bottom: Example digits generated by sampling from the DVSIB decoder, X and Y branches.

varied from 2^1 to 2^6. The highest accuracy is reported in Tbl. 3.2, along with the optimal parameters used to obtain this accuracy. Additionally, for every method we find the range of β and the dimensionality k_Z of the latent variable Z_Y that gives 95% of the method's maximum accuracy. If the range includes the limits of the parameter, this is indicated by an asterisk.

Figure 3.3 shows a t-SNE plot of DVSIB's latent space, Z_Y, colored by the identity of digits. The resulting latent space has 10 clusters, each corresponding to one digit. The clusters are well separated and interpretable. Further, DVSIB's Z_Y latent space provides the best classification of digits using a linear method such as an SVM showing the latent space is linearly separable. DVSIB maximum classification accuracy obtained for the linear SVM is 97.8%. Crucially, DVSIB maintains accuracy of at least 92.9% (95% of 97.8%) for $\beta \in [2, 1024^*]$ and $k_Z \in [8, 256^*]$. This accuracy is high compared to other methods and has a large range of hyperparameters that maintain its ability to correctly capture information about the identity of the shared digit. DVSIB is a generative method, we have provided sample generated digits from the decoders that were trained from the model graph.

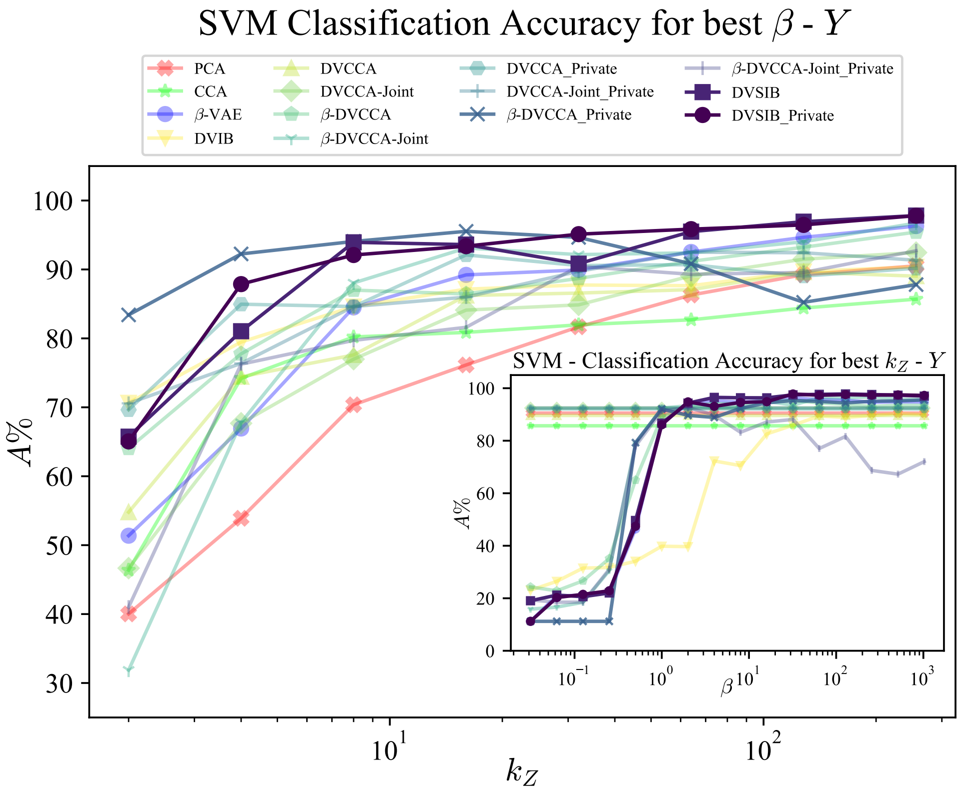


Figure 3.4: The best SVM classification accuracy curves for each method. Here DVSIB and DVSIB-private obtained the best accuracy and, together with β-DVCCA-private, they had the best accuracy for low latent dimensional spaces.

In Fig. 3.4, we show the highest SVM classification accuracy curves for each method. DVSIB and DVSIB-private tie for the best classification accuracy for Y. Together with β-DVCCA-private they have the highest accuracy for all dimensions of the latent space, k_Z. In theory, only one dimension should be needed to capture the identity of a digit, but our datasets also contain information about the rotation and scale for X and the strength of the background noise for Y. Y should then need at least two latent dimensions to be reconstructed and X should need at least three. Since DVSIB,

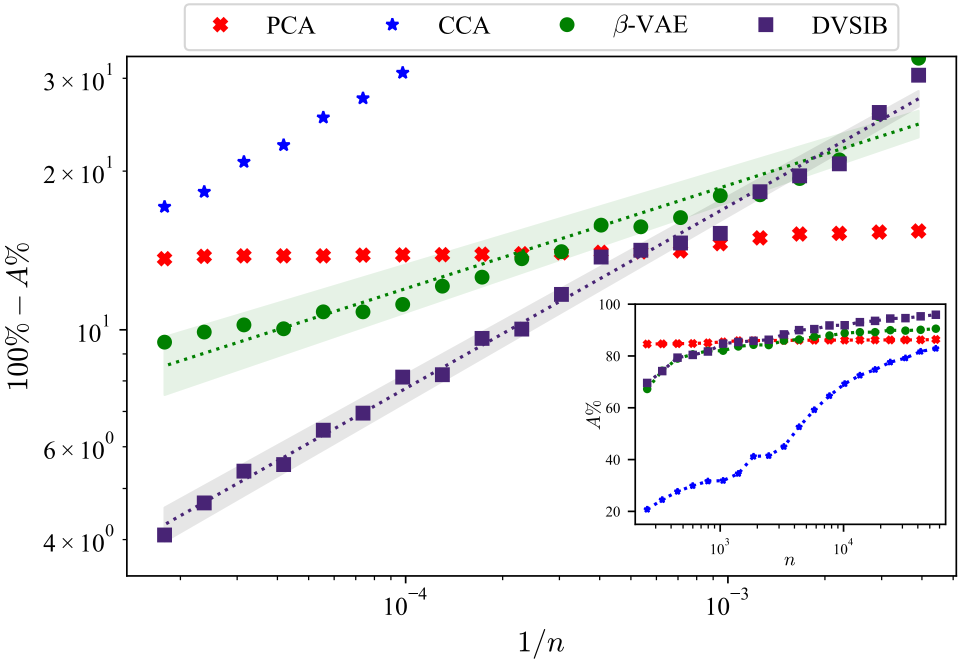


Figure 3.5: Classification accuracy (A) of DVSIB has a better sample size (n) dependent scaling. Main: a log-log plot of $100\% - A$ vs $1/n$. Slope for fitted lines are 0.345 ± 0.007 for DVSIB, and 0.196 ± 0.013 for β-VAE, corresponding to a faster increase of accuracy of DVSIB with n. Inset: same data, but plotted as A vs n.

DVSIB-private, and β-DVCCA-private performed with the best accuracy starting with the smallest k_Z, we conclude that methods with the encoder-decoder graphs that more closely match the structure of the data produce higher accuracy with lower dimensional latent spaces.

Next, in Fig. 3.5, we compare the sample training efficiency of DVSIB and β-VAE by training new instances of these methods on a geometrically increasing number of samples $n = [256, 339, 451, \ldots, \sim 42k, \sim 56k]$, consisting of 20 subsamples of the full training data (X, Y) to get $(X_{\text{train}_n}, Y_{\text{train}_n})$, where each larger subsample includes the previous one. Each method was trained for 60 epochs, and we used $\beta = 1024$ (as defined by the DVMIB framework). Further, all reported results are with the latent space size $k_Z = 64$. We explored other numbers of training epochs and latent space dimensions (see Appendix 3.7.5.1), but did not observe qualitative differences. We follow the same procedure as outlined earlier, using the 20 trained encoders for each method to compute Z_{train_n}, Z_{test}, and $Z_{\text{validation}}$ for the training, test, and validation datasets. As before, we then train and evaluate the classification accuracy of SVMs for the Z_Y representation learned by each method. Fig. 3.5, inset, shows the classification accuracy of each method as a function of the number of samples used in training. Again, CCA and PCA serve as linear methods baselines. PCA is able to capture the linear correlations in the dataset consistently, even at low sample sizes. However, it is unable to capture the nonlinearities of the data, and its accuracy does not improve with the sample size. Because of the iterative nature of the implementation of the PCA algorithm [117], it is able to capture some linear correlations in a relatively low number of dimensions, which are sufficiently sampled even with small-sized datasets. Thus the accuracy of PCA barely depends on the training set size. CCA, on the other hand, does not work in the under-sampled regime (see [3] for discussion of this). DVSIB performs uniformly better, at all training set sizes, than the β-VAE. Furthermore, DVSIB improves its quality faster, with a different sample size scaling. Specifically, DVSIB and β-VAE accuracy (A, measured in percent) appears to follow the scaling form $A = 100 - c/n^m$, where c is a constant, and the scaling exponent $m = 0.345 \pm 0.007$ for DVSIB, and 0.196 ± 0.013 for β-VAE. We illustrate this scaling in Fig. 3.5 by plotting a log-log plot of $100 - A$ vs $1/n$ and observing a linear relationship.

3.6 Conclusion

We developed an MIB-based framework for deriving variational loss functions for DR applications. We demonstrated the use of this framework by developing a novel variational method, DVSIB. DVSIB compresses the variables X and Y into latent variables Z_X and Z_Y respectively, while maximizing the information between Z_X and Z_Y. The method generates two distinct latent spaces—a feature highly sought after in various applications—but it accomplishes this with superior data efficiency, compared to other methods. The example of DVSIB demonstrates the process of deriving variational bounds for terms present in all examined DR methods. A comprehensive library of typical terms is included in Appendix 3.7.1 for reference, which can be used to derive additional DR methods. Further, we (re)-derive several DR methods, as outlined in Table 3.1. These include well-known techniques such as β-VAE, DVIB, DVCCA, and DVCCA-private. MIB naturally introduces a trade-off parameter into the DVCCA family of methods, resulting in what we term the β-DVCCA DR methods, of which DVCCA is a special case. We implement this new family of methods and show that it produces better latent spaces than DVCCA at $\beta = 1$, cf. Tbl. 3.2.

We observe that methods that more closely match the structure of dependencies in the data can give better latent spaces as measured by the dimensionality of the latent space and the accuracy of reconstruction (see Figure 3.4). This makes DVSIB, DVSIB-private, and β-DVCCA-private perform the best. DVSIB and DVSIB-private both have separate latent spaces for X and Y. The private methods allow us to learn additional aspects about X and Y that are not important for the shared digit label, but allow reconstruction of the rotation and scale for X and the background noise of Y. We also found that DVSIB can make more efficient use of data when producing latent spaces as compared to β-VAEs and linear methods.

Our framework may be extended beyond variational approaches. For instance, in the deterministic limit of VAE, autoencoders can be retrieved by defining the encoder/decoder graphs as nonlinear neural networks $z = f(x)$ and $x = g(z)$. Additionally, linear methods like CCA can be viewed as special cases of the information bottleneck [34] and hence must follow from our approach. Similarly, by using specialized encoder and decoder neural networks, e.g., convolutional ones, our framework can implement symmetries and other constraints into the DR process. Overall, the framework serves as a versatile and customizable toolkit, capable of encompassing a wide spectrum of dimensionality reduction methods. With the provided tools and code [1] , we aim to facilitate the adaptation of the approach to diverse problems.

3.7 Supplementary Information

3.7.1 Deriving and Designing Variational Losses

In the next two subsections, we provide a library of typical terms found in encoder graphs, Appendix 3.7.1.1, and decoder graphs, Appendix3.7.1.2. In Appendix 3.7.1.3, we provide examples of combining these terms to produce variational losses corresponding to beta-VAE, DVIB, beta-DVCCA, beta-DVCCA-joint, beta-DVCCA-private, DVSIB, and DVSIB-private.

3.7.1.1 Encoder Graph Components

We expand Sec. 3.3.2 and present a range of common components found in encoder graphs across various DR methods, cf. Fig. (3.6).

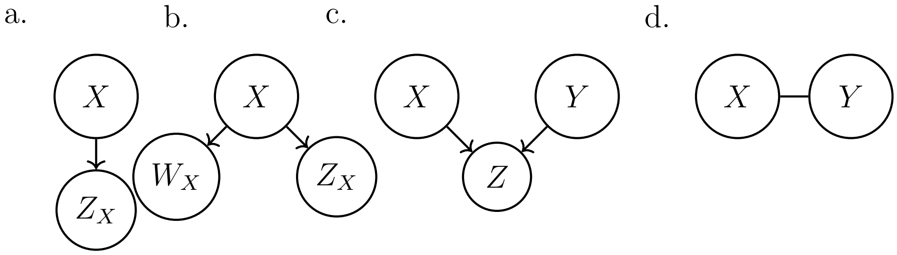


Figure 3.6: Encoder graph components.

a. This graph corresponds to compressing the random variable X to Z_X. Variational bounds for encoders of this type were derived in the main text in Sec. 3.3.2 and correspond to the loss:

$$\begin{aligned}\tilde{I}^E(X;Z_X) &= \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(z_x|x_i)\|r(z_x))\\ &\approx \frac{1}{2N}\sum_{i=1}^{N}\left[\mathrm{Tr}(\Sigma_{Z_X}(x_i)) + ||\vec{\mu}_{Z_X}(x_i)||^2 - k_{Z_X} - \ln\det(\Sigma_{Z_X}(x_i))\right].\end{aligned} \tag{3.17}$$

b. This type of encoder graph is similar to the first, but now with two outputs, Z_X and W_X. This corresponds to making two encoders, one for Z_X and one for W_X, $\tilde{I}^E(Z_X;X) + \tilde{I}^E(W_X;X)$, where

$$\tilde{I}^E(Z_X;X) \approx \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(z_x|x_i)\|r(z_x)), \tag{3.18}$$

$$\tilde{I}^E(W_X;X) \approx \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(w_x|x_i)\|r(w_x)). \tag{3.19}$$

c. This type of encoder consists of compressing X and Y into a single variable Z. It corresponds to the information loss $I^E(Z;(X,Y))$. This again has a similar encoder structure to type (a), but X is replaced by a joint variable (X,Y). For this loss, we find a variational version:

$$\tilde{I}^E(Z;(X,Y)) \approx \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(z|x_i,y_i)\|r(x_i,y_i)). \tag{3.20}$$

d. This final type of an encoder term corresponds to information $I^E(X,Y)$, which is constant with respect to our minimization. In practice, we drop terms of this type.

3.7.1.2 Decoder Graph Components

In this section, we elaborate on the decoder graphs that happen in our considered DR methods, cf. Fig. (3.7).

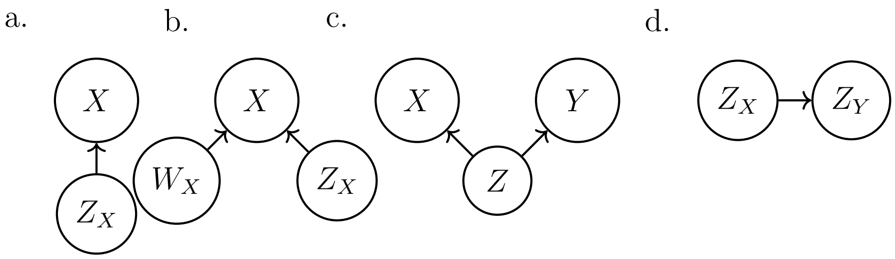


Figure 3.7: Decoder graph components.

All decoder graphs sample from their methods' corresponding encoder graph.

a. In this decoder graph, we decode X from the compressed variable Z_X. Variational bounds for decoders of this type were derived in the main text, Sec. 3.3.3, and they correspond to the loss:

$$\begin{aligned}\tilde{I}^D(X;Z_X) &= H(X) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln q(x_i|z_x)\\ &\approx H(X) + \frac{1}{MN}\sum_{i,j=1}^{N,M} -\frac{1}{2}||(x_i - \mu_X(z_{xi,j}))||^2,\end{aligned} \tag{3.21}$$

where $H(X)$ can be dropped from the loss since it doesn't change in optimization.

b. This type of decoder term is similar to that in part (a), but X is decoded from two variables simultaneously. The corresponding loss term is $I^D(X;(Z_X,W_X))$. We find a variational loss by replacing Z_X in part (a) by (Z_X,W_X):

$$\tilde{I}^D(X;(Z_X,W_X)) \approx H(X) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x dw_x p(z_x,w_x|x_i)\ln(q(x_i|z_x,w_x)), \tag{3.22}$$

where, again, the entropy of X can be dropped.

c. This decoder term can be obtained by adding two decoders of type (a) together.

In this case, the loss term is $I^D(X;Z) + I^D(Y;Z)$:

$$\tilde{I}^D(X;Z) + \tilde{I}^D(Y;Z) \approx H(X) + H(Y) \\ + \frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_i)\ln(q(x_i|z)) + \frac{1}{N}\sum_{i=1}^{N}\int dzp(z|y_i)\ln(q(y_i|z)), \quad (3.23)$$

and the entropy terms can be dropped, again.

d. Decoders of this type were discussed in the main text in Sec. 3.3.4. They correspond to the information between latent variables Z_X and Z_Y. We use the MINE estimator to find variational bounds for such terms:

$$\tilde{I}^D_{\text{MINE}}(Z_X;Z_Y) = \int dz_x dz_y p(z_x,z_y)\ln\frac{e^{T(z_x,z_y)}}{\mathcal{Z}_{\text{norm}}} \approx \frac{1}{NM^2}\sum_{i,j_x,j_y=1}^{N,M,M}\left[T(z_{xi,j_x}, z_{y_{i,j_y}}) - \ln\mathcal{Z}_{\text{norm}}\right]. \quad (3.24)$$

3.7.1.3 Detailed Method Implementations

For completeness, we provide detailed implementations of methods outlined in Tbl. 3.1.

3.7.1.3.1 Beta Variational Auto-Encoder

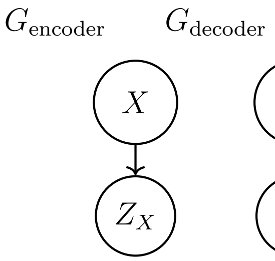


Figure 3.8: Encoder and decoder graphs for the beta-variational auto-encoder method

A variational autoencoder [87, 66] compresses X into a latent variable Z_X and then reconstructs X from the latent variable, cf. Fig. (3.9). The overall loss is a trade-off between the compression $I^E(X;Z_X)$ and the reconstruction $I^D(X;Z_X)$:

$$\begin{aligned} &I^E(X;Z_X) - \beta I^D(X;Z_X) \leq \tilde{I}^E(X;Z_X) - \beta \tilde{I}^D(X;Z_X) \\ &\lesssim \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(z_x|x_i)\|r(z_x)) - \beta\left(H(X) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(x_i|z_x))\right). \end{aligned} \tag{3.25}$$

$H(X)$ is a constant with respect to the minimization, and it can be omitted from the loss. Similar to the main text, DVSIB case, we make ansatzes for forms of each of the variational distributions. We choose parametric distribution families and learn the nearest distribution in these families consistent with the data. Specifically, we assume $p(z_x|x)$ is a normal distribution with mean $\mu_{Z_X}(X)$ and variance $\Sigma_{Z_X}(X)$. We learn the mean and the log-variance as neural networks. We also assume that $q(x|z_x)$ is normal with a mean $\mu_X(z_x)$ and a unit variance. Finally, we assume that $r(z_x)$ is drawn from a standard normal distribution. We then use the re-parameterization trick to produce samples of $z_{x_j}(x) = \mu(x) + \sqrt{\Sigma_{Z_X}(x)}\eta_j$ from $p(z_x|x)$, where η is drawn from a standard normal distribution. Overall, this gives:

$$\begin{aligned} L_{\mathrm{VAE}} = &\frac{1}{2N}\sum_{i=1}^{N}\left[\mathrm{Tr}(\Sigma_{Z_X}(x_i)) + \vec{\mu}_{Z_X}(x_i)^T\vec{\mu}_{Z_X}(x_i) - k_{Z_X} - \ln\det(\Sigma_{Z_X}(x_i))\right] \\ &- \beta\left(\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M} -\frac{1}{2}(x_i - \mu_X(z_{x_j}))^T(x_i - \mu_X(z_{x_j}))\right). \end{aligned} \tag{3.26}$$

This is the same loss as for a beta auto-encoder. However, following the convention in the Information Bottleneck literature [147, 50], our β is the inverse of the one typically used for beta auto-encoders. A small β in our case results in a stronger compression, while a large β results in a better reconstruction.

3.7.1.3.2 Deep Variational Information Bottleneck

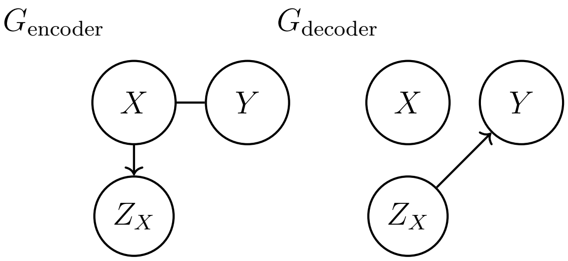


Figure 3.9: Encoder and decoder graphs for the Deep Variational Information Bottleneck.

Just as in the beta auto-encoder, we immediately write down the loss function for the information bottleneck. Here, the encoder graph compresses X into Z_X, while the decoder tries to maximize the information between the compressed variable and the relevant variable Y, cf. Fig. (3.9). The resulting loss function is:

$$L_{\text{IB}} = I^E(X;Y) + I^E(X;Z_X) - \beta I^D(Y;Z_X). \tag{3.27}$$

Here the information between X and Y does not depend on $p(z_x|x)$ and can dropped in the optimization.

Thus the Deep Variational Information Bottleneck [5] becomes :

$$L_{\text{DVIB}} \approx \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_x|x_i)\|r(z_x)) - \beta\left(\frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(y_i|z_x))\right), \tag{3.28}$$

where we dropped $H(Y)$ since it doesn't change in the optimization.

As we have been doing before, we choose to parameterize all these distributions by Gaussians and their means and their log variances are learned by neural networks. Specifically, we parameterize $p(z_x|x) = N(\mu_{z_x}(x), \Sigma_{z_x})$, $r(z_x) = N(0, I)$, and $q(y|z_x) = N(\mu_Y, I)$. Again we can use the reparameterization trick and sample from $p(z_x|x_i)$ by $z_{x_j}(x) = \mu(x) + \sqrt{\Sigma_{z_x}(x)}\eta_j$ where η is drawn from a standard normal distribution.

3.7.1.3.3 Beta Deep Variational CCA

beta-DVCCA, cf. Fig. 3.10, is similar to the traditional information bottleneck, but now X and Y are both used as relevance variables:

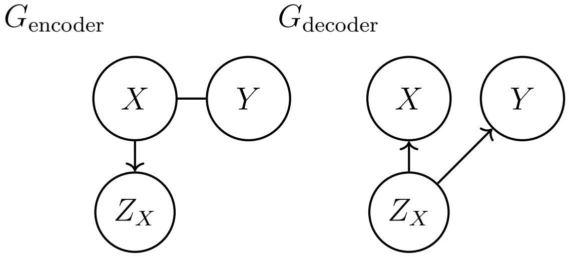


Figure 3.10: Encoder and decoder graphs for beta Deep Variational CCA.

$$L_{\text{DVCCA}} = \tilde{I}^E(X;Y) + \tilde{I}^E(X;Z_X) - \beta(\tilde{I}^D(Y;Z_X) + \tilde{I}^D(X;Z_X)) \tag{3.29}$$

Using the same library of terms as before, we find:

$$\begin{aligned} L_{\text{DVCCA}} \approx & \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_x|x_i)\|r(z_x)) \\ & - \beta\left(\frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(y_i|z_x)) + \frac{1}{N}\sum_{i=1}^{N}\int dz_x p(z_x|x_i)\ln(q(x_i|z_x))\right). \end{aligned} \tag{3.30}$$

This is similar to the loss function of the deep variational CCA [158], but now it has a trade-off parameter β. It trades off the compression into Z against the reconstruction of X and Y from the compressed variable Z.

3.7.1.3.4 Beta joint-Deep Variational CCA

Joint deep variational CCA [158], cf. Fig. 3.11, compresses (X,Y) into one Z and then reconstructs the individual terms X and Y,

$$L_{\text{DVCCA}} = I^E(X;Y) + I^E((X,Y);Z) - \beta(I^D(Y;Z) + I^D(X;Z)). \tag{3.31}$$

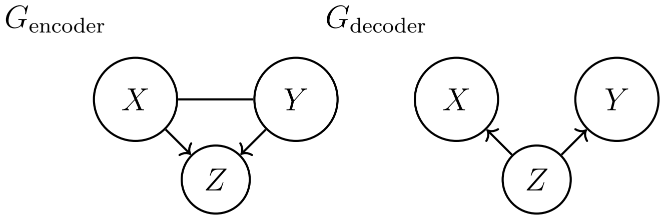


Figure 3.11: Encoder and decoder graphs for beta joint-Deep Variational CCA.

Using the terms we derived, the loss function is:

$$L_{\text{DVCCA}} \approx \frac{1}{N}\sum_{i=1}^{N} D_{KL}(p(z|x_i, y_i)\|r(z)) \\ - \beta\left(\frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_i)\ln(q(y_i|z)) + \frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_i)\ln(q(x_i|z))\right). \quad (3.32)$$

The information between X and Y does not change under the minimization and can be dropped.

3.7.1.3.5 Beta joint-Deep Variational CCA-private

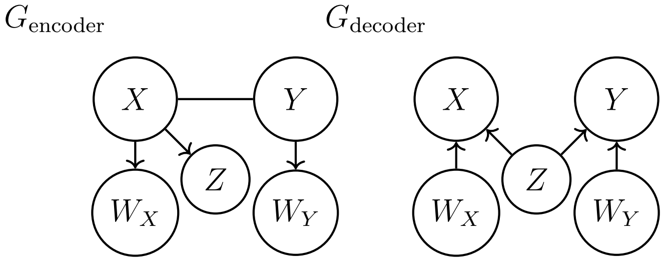


Figure 3.12: Encoder and decoder graphs for beta Deep Variational CCA-private

This is a generalization of the Deep Variational CCA [158] to include private information, cf. Fig. 3.12. Here X is encoded into a shared latent variable Z and a private latent variable W_X. Similarly Y is encoded into the same shared variable and a different private latent variable W_Y. X is reconstructed from Z and W_X, and Y is reconstructed from Z and W_Y. In the joint version (X, Y) are compressed jointly in

Z similar to the previous joint methods. What follows is the loss X version of beta Deep Variational CCA-private.

$$\begin{aligned} L_{\text{DVCCAp}} = I^E(X;Y) + I^E((X,Y);Z) + I^E(X;W_X) + I^E(Y;W_Y) \\ - \beta(I^D(X;(W_X,Z)) + I^D(Y;(W_Y,Z))). \end{aligned} \tag{3.33}$$

After the usual variational manipulations, this becomes:

$$\begin{aligned} L_{\text{DVCCAp}} \approx \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z|x_i)\|r(z)) + \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(w_x|x_i)\|r(w_x)) \\ + \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(w_y|y_i)\|r(w_y)) - \beta\left(\frac{1}{N}\sum_{i=1}^{N}\int dz dw_x p(w_x|x_i)p(z|x_i)\ln(q(y_i|z,w_x))\right. \\ \left. + \frac{1}{N}\sum_{i=1}^{N}\int dz dw_y p(w_y|y_i)p(z|x_i)\ln(q(x_i|z,w_y))\right). \end{aligned} \tag{3.34}$$

3.7.1.3.6 Deep Variational Symmetric Information Bottleneck

This has been analyzed in detail in the main text, Sec. 3.3.1, and will not be repeated here.

3.7.1.3.7 Deep Variational Symmetric Information Bottleneck-private

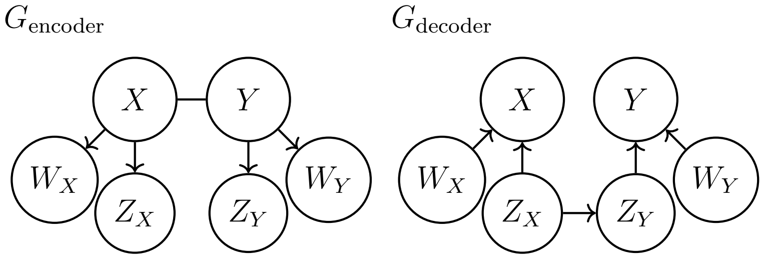


Figure 3.13: Encoder and decoder graphs for DVSIB-private.

This is a generalization of the Deep Variational Symmetric Information Bottleneck to include private information. Here X is encoded into a shared latent variable Z_X

and a private latent variable W_X. Similarly, Y is encoded into its own shared Z_Y variable and a private latent variable W_Y. X is reconstructed from Z_X and W_X, and Y is reconstructed from Z_Y and W_Y. Z_X and Z_Y are constructed to be maximally informative about each another. This results in

$$\begin{aligned} L_{\text{DVSIBp}} &= I^E(X;W_X) + I^E(X;Z_X) + I^E(Y;Z_Y) + I^E(Y;W_Y) \\ &\quad - \beta\left(I^D(Z_X;Z_Y) + I^D(X;(Z_X,W_X)) + I^D(Y;(Z_Y,W_Y))\right). \end{aligned} \tag{3.35}$$

After the usual variational manipulations, this becomes (see also main text):

$$\begin{aligned} L_{\text{DVSIBp}} &\approx \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_x|x_i)\|r(z_x)) + \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_y|x_i)\|r(z_y)) \\ &\quad + \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(w_x|x_i)\|r(w_x)) + \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(w_y|y_i)\|r(w_y)) \\ - \beta &\Bigg(\int dz_x dz_y p(z_x,z_y) \ln\frac{e^{T(z_x,z_y)}}{\mathcal{Z}_{\text{norm}}} + \frac{1}{N}\sum_{i=1}^{N}\int dz_y dw_y p(w_y|y_i)p(z_y|y_i)\ln(q(y_i|z_y,w_y)) \\ &\quad + \frac{1}{N}\sum_{i=1}^{N}\int dz_x dw_x p(w_x|x_i)p(z_x|x_i)\ln(q(x_i|z_x,w_x))\Bigg), \end{aligned} \tag{3.36}$$

where

$$\mathcal{Z}_{\text{norm}} = \int dz_x dz_y p(z_x)p(z_y)e^{T(z_x,z_y)}. \tag{3.37}$$

3.7.2 Multi-variable Losses (More than 2 Views / Variables)

It is possible to rederive several multi-variable losses that have appeared in the literature within our framework.

3.7.2.1 Multi-view Total Correlation Auto-encoder

Here we demonstrate several graphs for multi-variable losses. This first example consists of a structure, where all the views X_1, X_2, and X_3 are compressed into the

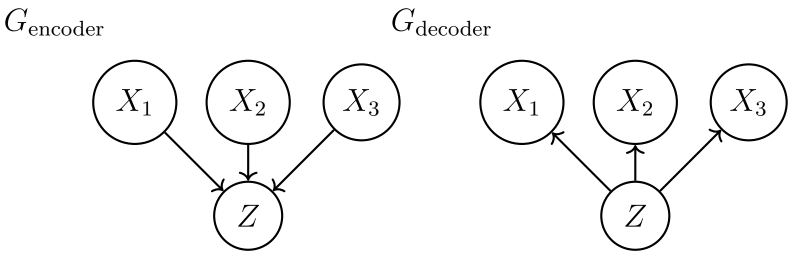


Figure 3.14: Encoder and decoder graphs for a multi-view auto-encoder.

same latent variable Z. The corresponding decoder produces reconstructed views from the same latent variable Z. This is known in the literature as a multi-view auto-encoder.

$$L_{\text{MVAE}} = \tilde{I}^E((X_1, X_2, X_3); Z) - \beta(\tilde{I}^D(X_1; Z) + \tilde{I}^D(X_2; Z) + \tilde{I}^D(X_3; Z)). \tag{3.38}$$

Using the same library of terms as before, we find:

$$\begin{aligned} L_{\text{MVAE}} &\approx \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z|x_{1i}, x_{2i}, x_{3i})\|r(z)) \\ -\beta &\left(\frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_{1i}, x_{2i}, x_{3i})\ln(q(x_{1i}|z)) + \frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_{1i}, x_{2i}, x_{3i})\ln(q(x_{2i}|z))\right. \\ &\left. +\frac{1}{N}\sum_{i=1}^{N}\int dzp(z|x_{1i}, x_{2i}, x_{3i})\ln(q(x_{3i}|z))\right). \end{aligned} \tag{3.39}$$

3.7.2.2 Deep Variational Multimodal Information Bottlenecks

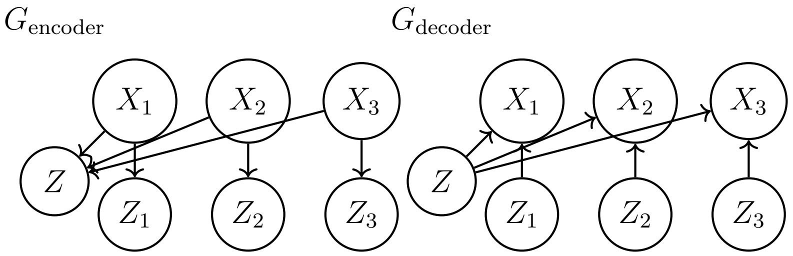


Figure 3.15: Encoder and decoder graphs for Multimodal Information Bottleneck.

This example consists of a structure where all the views X_1, X_2, and X_3 are compressed into separate latent views Z_1, Z_2, and Z_3 and one global shared latent variable Z. This structure is analogous to DVCCA-private, but it extends to three variables rather than two. It appears in the literature with slightly different variations. In the decoder graph, X_1 is reconstructed from both Z and Z_1, X_2 is reconstructed from both Z and Z_2, and X_3 is reconstructed from both Z and Z_3.

$$\begin{aligned} L_{\text{DVAE}} = \tilde{I}^E((X_1, X_2, X_3); Z) + \tilde{I}^E(X_1; Z_1) + \tilde{I}^E(X_2; Z_2) + \tilde{I}^E(X_3; Z_3) \\ - \beta(\tilde{I}^D(X_1; (Z, Z_1)) + \tilde{I}^D(X_2; (Z, Z_2)) + \tilde{I}^D(X_3; (Z, Z_3))) \quad (3.40) \end{aligned}$$

Using the same library of terms as before, we find:

$$\begin{aligned} L_{\text{DVAE}} \approx \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z|x_{1i}, x_{2i}, x_{3i})\|r(z)) + \sum_{j=1}^{3}\frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(p(z_j|x_{j_i})\|r_j(z)) \\ - \beta\left(\frac{1}{N}\sum_{i=1}^{N}\int dzdz_1dz_2dz_3p(z|x_{1i}, x_{2i}, x_{3i})p(z_1|x_{1i})p(z_2|x_{2i})p(z_3|x_{3i})\ln(q(x_{1i}|z, z_1))\right. \\ +\frac{1}{N}\sum_{i=1}^{N}\int dzdz_1dz_2dz_3p(z|x_{1i}, x_{2i}, x_{3i})p(z_1|x_{1i})p(z_2|x_{2i})p(z_3|x_{3i})\ln(q(x_{2i}|z, z_2)) \\ \left.+\frac{1}{N}\sum_{i=1}^{N}\int dzdz_1dz_2dz_3p(z|x_{1i}, x_{2i}, x_{3i})p(z_1|x_{1i})p(z_2|x_{2i})p(z_3|x_{3i})\ln(q(x_{3i}|z, z_3))\right). \\ (3.41) \end{aligned}$$

3.7.2.3 Discussion

There exist many other structures that have been explored in the multi-view representation learning literature, including conditional VIB [134, 82], which is formulated in terms of conditional information. These types of structures are beyond the current scope of our framework. However, they could be represented by an encoder mapping from all independent views X_ν to Z, subtracted from another encoder mapping from the joint view $\vec{X}$ to Z. Coupled with this would be a decoder mapping from Z to the independent views X_ν (or the joint view $\vec{X}$, analogous to the Joint-DVCCA). Similarly, one can use our framework to represent other multi-view approaches, or their approximations [96, 154, 82]. This underscores the breadth of methods seeking to address specific questions by exploring known or assumed statistical dependencies within data, and also the generality of our approach, which can re-derive these methods.

3.7.3 Multi-view Information Bottleneck

The multiview information bottleneck (MVIB) [46] attempts to remove redundant information between views (v_1, v_2). This is achieved with the following losses:

$$L_1 = I(z_1; v_1|v_2) - \lambda_1 I(v_2; z_1), \tag{3.42}$$

$$L_2 = I(z_2; v_2|v_1) - \lambda_1 I(v_1; z_2). \tag{3.43}$$

These losses are equivalent to two deep variational information bottlenecks performed in parallel. Within our framework, the same algorithm emerges with the encoder graph that compresses v_1 into z_1 and v_2 into z_2, while the decoder graph would reconstruct v_2 from z_1 and v_1 from z_2.

[46] combines these two losses while enforcing the condition that z_1 and z_2 are the same. They bounded the combined loss function to obtain:

$$L_{\text{MVIB}} = D_{SKL}(P(z_1|v_1)||P(z_2|v_2)) - \beta I(z_1, z_2), \tag{3.44}$$

with z_1 and z_2 being the same latent space in this approximation. Here D_{SKL} is the symmetrized KL divergence, v_i corresponds to the two different views, and z_i corresponds to their two latent, compressed representation. (Here we changed the parameter β to be in front of $I(z_1, z_2)$, to be consistent with the definition of β we use elsewhere in this work.) While this loss looks similar to the DVSIB loss, it is conceptually different. It attempts to produce latent variables that are as similar to one another as possible (ideally, $z_1 = z_2$). In contrast, DVSIB attempts to produce different latent variables that could, in theory, have different units, dimensionalities, and domains, while still being as informative about each other as possible. For example, in the noisy MNIST, Z_X contains information about the labels, the angles, and the scale of images (all needed for reconstructing X) and no information about the noise structure. At the same time, Z_Y contains information about the labels and the noise factor only (both needed to reconstruct Y). See Appendix 3.7.4.4 for 2-d latent spaces colored by these variables, illustrating the difference between Z_X and Z_Y in DVSIB. Further, in practice, the implementation of MVIB uses the same encoder for both views of the data; this is equivalent to encoding different views using the same function and then trying to force the output to be as close as possible to each other, in contrast to DVSIB.

We evaluate MVIB on the noisy MNIST dataset and include it in Table 3.2. The performance is similar to that of DVSIB, but slightly worse.

Moreover, MVIB appears to be highly sensitive to parameters and training conditions. Despite employing identical initial conditions and parameters used for training other methods, the approach often experienced collapses during training, resulting in infinities. Interestingly enough, in instances where training persisted for a limited set of parameters (usually low k_Z and high β), MVIB generated good latent spaces, evidenced by their relatively high classification accuracy.

3.7.4 Additional MNIST Results

In this section, we present supplementary results derived from the methods in Tbl 3.1.

3.7.4.1 Additional Results Tables for the Best Parameters

We report classification accuracy using SVM on data X, and using neural networks on both X and Y.

Table 3.3: Maximum accuracy from a linear SVM and the optimal k_Z and β for variational DR methods on the X dataset. († fixed values)

Method	**Acc. %**	$k_{Z_{\textbf{best}}}$	95% $k_{Z_{\textbf{range}}}$	$\boldsymbol{\beta}_{\text{best}}$	95% $\beta_{\textbf{range}}$	$\boldsymbol{C}_{\text{best}}$
Baseline	57.8	784†	-	-	-	0.01
PCA	58.0	256	[32,265*]	-	-	0.1
CCA	54.4	256	[8,265*]	-	-	0.032
β-VAE	84.4	256	[128,265*]	4	[2,8]	10
DVIB	87.3	128	[4,265*]	512	[8,1024*]	0.032
DVCCA	86.1	256	[64,265*]	1†	-	31.623
β-DVCCA	88.9	256	[128,265*]	4	[1,128]	10
DVCCA-private	85.3	128	[32,265*]	1†	-	31.623
β-DVCCA-private	85.3	128	[32,265*]	1	[1,8]	31.623
MVIB	**93.8**	**8**	[8,16]	128	[128,1024*]	0.01
DVSIB	**92.9**	256	[64,265*]	256	[4,1024*]	1
DVSIB-private	**92.6**	256	[**32**,265*]	128	[8,1024*]	3.162

Table 3.4: Maximum accuracy from a feed forward neural network and the optimal k_Z and β for variational DR methods on the Y and the joined $[X, Y]$ datasets. († fixed values)

Method	**Acc. %**	$k_{Z\mathbf{best}}$	95% $k_{Z\mathbf{range}}$	$\boldsymbol{\beta}_{\text{best}}$	95% $\beta_{\mathbf{range}}$
Baseline	92.8	$784^\dagger$	-	-	-
PCA	97.6	128	[16,256*]	-	-
CCA	90.2	256	[32,256*]	-	-
β-VAE	**98.4**	64	[8,256*]	64	[2,1024*]
DVIB	90.4	128	[8,256*]	1024	[8,1024*]
DVCCA	91.3	16	[4,256*]	$1^\dagger$	-
β-DVCCA	97.5	128	[8,256*]	512	[2,1024*]
DVCCA-private	93.8	16	[2,256*]	$1^\dagger$	-
β-DVCCA-private	97.5	256	[**2**,256*]	32	[1,1024*]
MVIB	97.5	16	[8,16]	256	[128,1024*]
DVSIB	**98.3**	256	[**4**,256*]	32	[2,1024*]
DVSIB-private	**98.3**	256	[**4**,256*]	32	[2,1024*]
Baseline-joint	97.7	$1568^\dagger$	-	-	-
joint-DVCCA	93.7	256	[8,256*]	$1^\dagger$	-
β-joint-DVCCA	**98.9**	64	[8,256*]	512	[2,1024*]
joint-DVCCA-private	93.5	16	[4,256*]	$1^\dagger$	-
β-joint-DVCCA-private	95.6	32	[4,256*]	512	[1,1024*]

3.7.4.2 t-SNE Embeddings at Best Parameters

Figures 3.16 and 3.17 display 2d t-SNE embeddings for variables Z_X and Z_Y generated by various considered DR methods.

3.7.4.3 DVSIB-private Reconstructions for Best Parameters

Figure 3.18 shows the t-SNE embeddings of the private latent variables constructed by DVSIB-private, colored by the digit label. To the extent that the labels do not cluster, private latent variables do not preserve the label information shared between X and Y.

Table 3.5: Maximum accuracy from a neural network the optimal k_Z and β for variational DR methods on the X dataset. († fixed values)

Method	**Acc. %**	$k_{Z_{\mathbf{best}}}$	95% $k_{Z_{\mathbf{range}}}$	$\boldsymbol{\beta}_{\text{best}}$	95% $\beta_{\mathbf{range}}$
Baseline	92.8	$784^\dagger$	-	-	-
PCA	91.9	64	[32,256*]	-	-
CCA	72.6	256	[256,256*]	-	-
β-VAE	93.3	256	[16,256*]	256	[2,1024*]
DVIB	87.5	4	[2,256*]	1024	[4,1024*]
DVCCA	87.5	128	[8,256*]	$1^\dagger$	-
β-DVCCA	92.2	64	[8,256*]	32	[2,1024*]
DVCCA-private	88.2	8	[8,256*]	$1^\dagger$	-
β-DVCCA-private	90.7	256	[4,256*]	8	[1,1024*]
MVIB	**93.6**	8	[**8**,16]	256	[128,1024*]
DVSIB	**93.9**	128	[**8**,256*]	16	[2,1024*]
DVSIB-private	92.8	32	[8,256*]	256	[4,1024*]

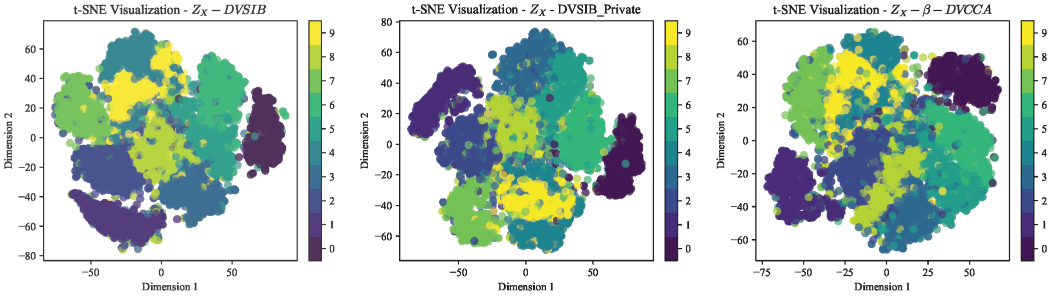


Figure 3.16: t-SNE X

3.7.4.4 Additional Results at 2 Latent Dimensions

We now demonstrate how different DR methods behave when the compressed variables are restricted to have not more than 2 dimensions, cf. Figs. 3.19, 3.20.

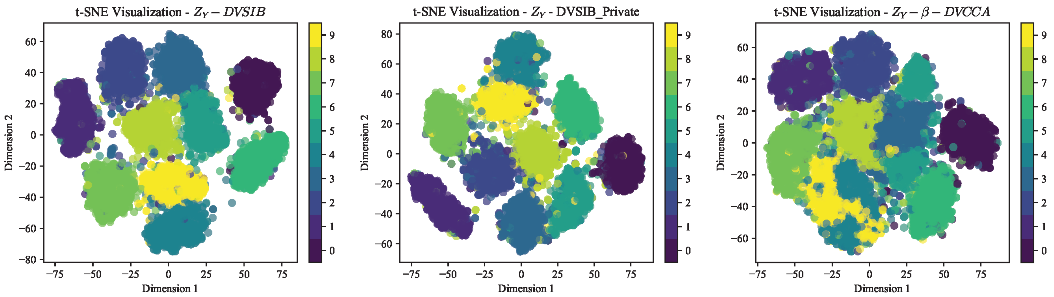


Figure 3.17: t-SNE Y

3.7.5 DVSIB-private Reconstructions at 2 Latent Dimensions

Figure 3.21 shows the reconstructions of the private latent variables constructed by DVSIB-private, colored by the digit label, rotations, scales, and noise factors for X (up), and Y (bottom). Private latent variables at 2 latent dimensions preserve a little about the label information shared between X and Y, but clearly preserve the scale information for X, even at only 2 latent dimensions.

3.7.5.1 Testing Training Efficiency

We tested an SVM's classification accuracy for distinguishing digits based on latent subspaces created by DVSIB, β-VAE, CCA, and PCA trained using different amounts of samples. Figure 3.5 in the main text shows the results for 60 epochs of training with latent spaces of dimension $k_{Z_X} = k_{Z_Y} = 64$. The DVSIB and β-VAE were trained with $\beta = 1024$. Figure 3.22 shows the SVM's classification accuracy for a range of latent dimensions (from right to left): $k_{Z_X} = k_{Z_Y} = 2, 16, 64, 256$. Additionally, it shows the results for different amounts of training time for the encoders ranging from 20 epochs (top row) to 100 epochs (bottom row). As explained in the main text, we plot a log-log graph of $100 - A$ versus $1/n$. Plotted in this way, high accuracy appears at the bottom, and large sample sizes are at the left of the plots. DVSIB, β-VAE, and CCA often appear linear when plotted this way, implying that they follow the form $A = 100 - c/n^m$. Steeper slopes m on these plots correspond to a faster increase in the

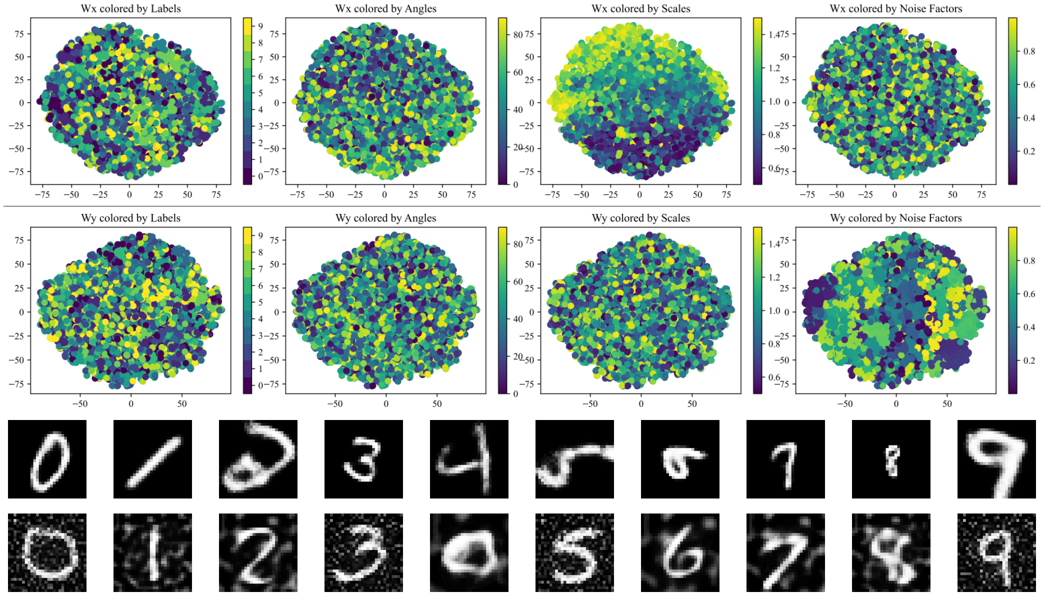


Figure 3.18: Private embeddings of DVSIB-private colored by labels, rotations, scales, and noise factors for X (up), and Y (middle). Reconstructions of the digits using *both* shared and private information (bottom) show that the private information allows to produce different backgrounds, scalings, and rotations.

accuracy with the sample size. This parameter sweep shows that the tested methods have not had time to fully converge at low epoch numbers. Additionally, increasing the number of latent dimensions helps the SVMs untangle the non-linearities present in the data and improves the corresponding classifiers.

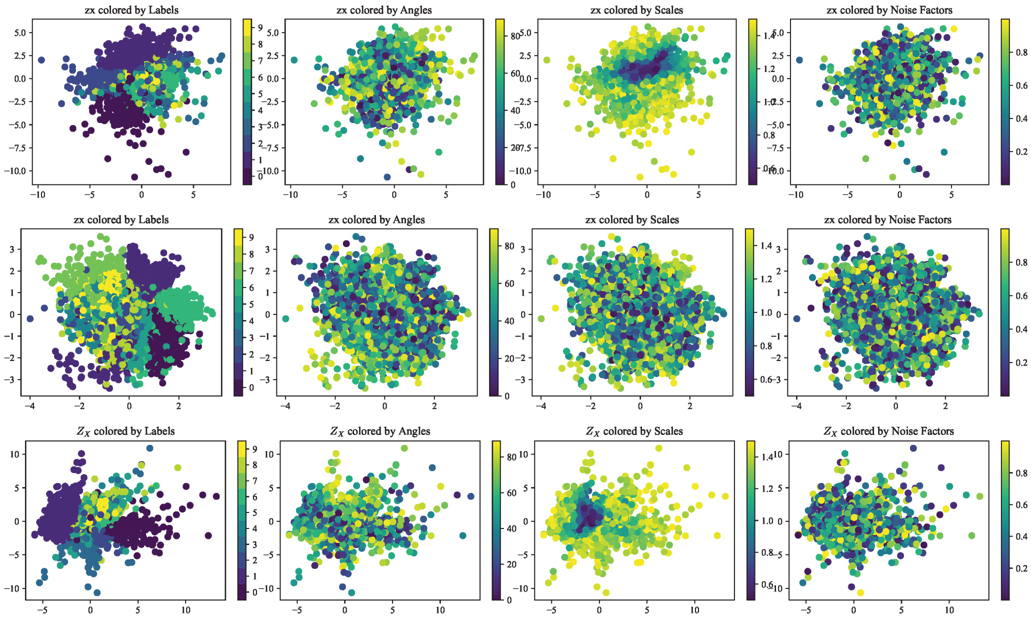


Figure 3.19: Clustering of embeddings when restricting k_{Z_X} to two for DVSIB, DVSIB-private, and β-DVCCA, results on the X dataset.

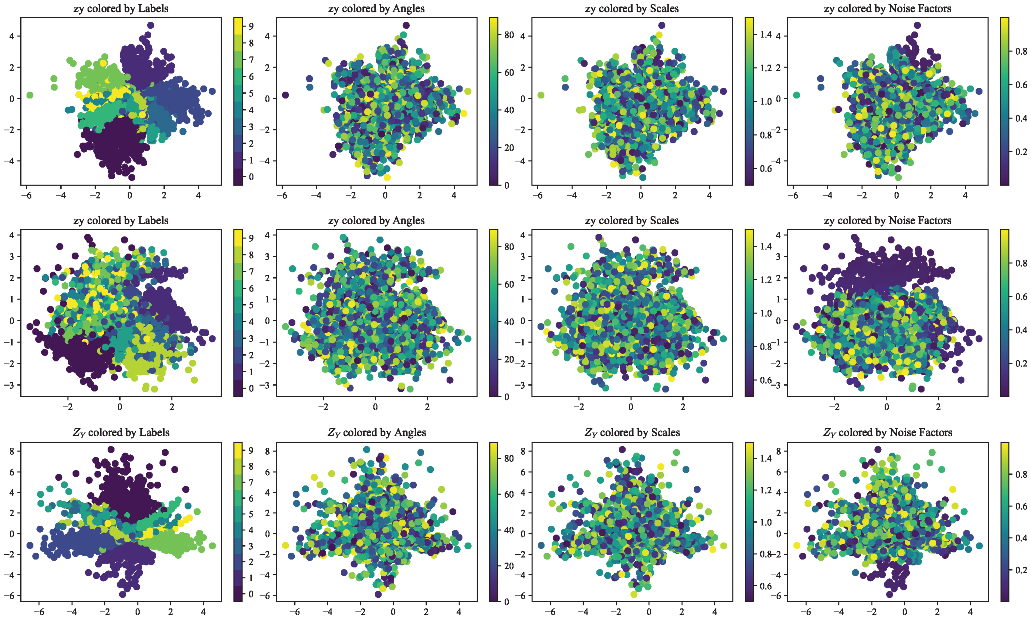


Figure 3.20: Clustering of embeddings when restricting k_{Z_Y} to two for DVSIB, DVSIB-private, and β-DVCCA, results on the Y dataset.

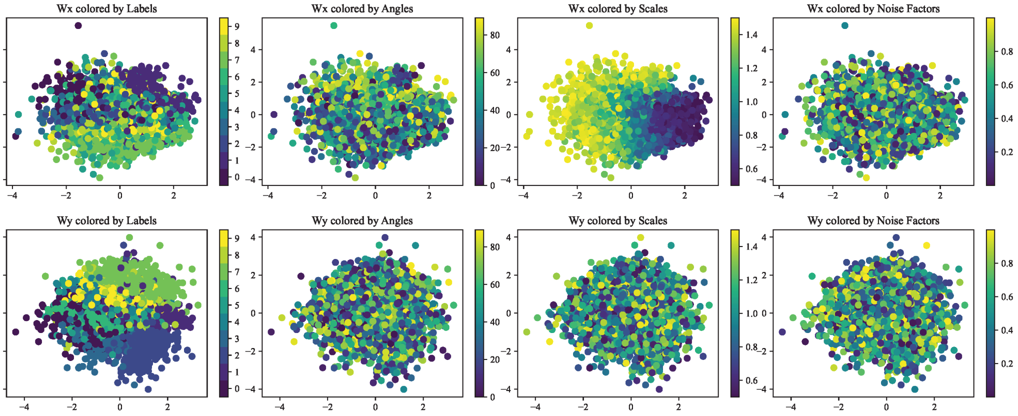


Figure 3.21: Private embeddings of DVSIB-private colored by labels, rotations, scales, and noise factors for X (top), and Y (bottom).

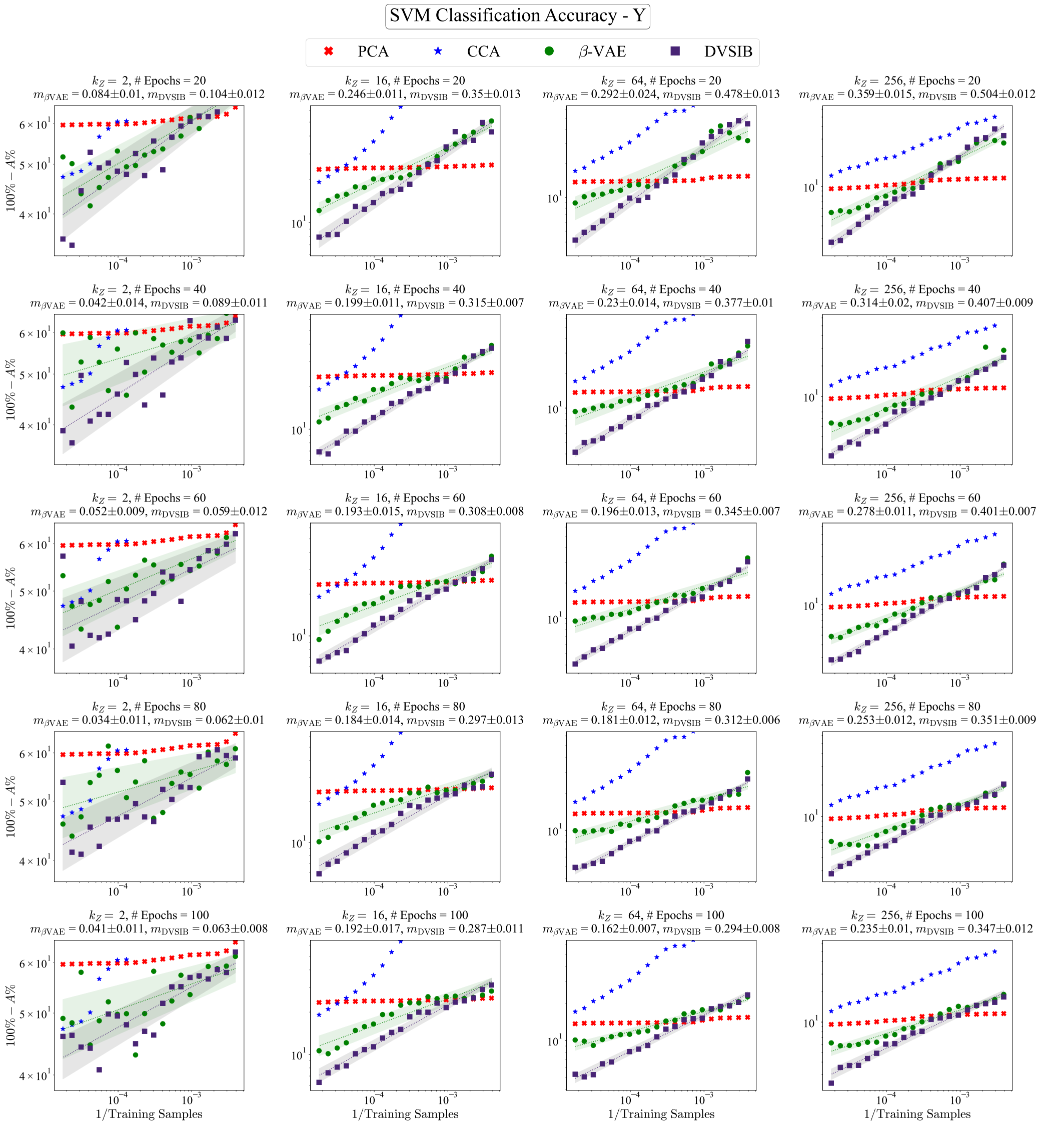


Figure 3.22: Log-log plot of $100 - A$ vs $1/n$. DVSIB has a steeper slope than β-VAE corresponding to faster convergence with fewer samples for DVSIB. Plots vary $k_Z = 2, 16, 64, 256$ and training epochs $20, 40, 60, 80, 100$.

Chapter 4

Efficient Estimation of Mutual Information in Very Large Dimensional Data

4.1 Summary

[1]Mutual information (MI) is a measure of statistical dependencies between two variables. It is a common tool in data analysis in many fields [85, 145, 138, 132, 149, 44]. Thus, accurate estimators of MI from empirical data are needed. However, such estimation is a hard problem, and there are provably no estimators that are universally good for finite datasets [8, 115, 73]. Commonly used estimators perform poorly on high dimensional data, which is a staple of modern experiments. Recently, a series of promising machine learning based MI estimation methods have been introduced. However, it remains unknown how their performance depends on the data set size and on the structure of nonlinearities in the data, as well as on hyperparameters of

[1] This section is based on an ongoing work with K. Michael Martini and Ilya Nemenman. The development of the idea for this chapter was a collaborative effort among all three authors. The code was written by K. Michael Martini and myself. The figures presented in this chapter are produced by myself. The text was jointly written by all authors.

the estimators, such as the dimensionality of the space used by the neural networks to embed the data and on the duration of training. There are also no accepted tests to signal when the estimators should or should not be trusted. In this Chapter, we systematically explore the dependence of MI estimators on properties of the data sets and on their hyperparameters. We propose and verify a protocol for accurate estimation of MI, with explicit checks for reliability and consistency of the estimators. We show that one can estimate MI reliably from data sets where the number of samples is of the order of the number of dimensions in the data, provided that the statistical dependencies in data can be summarized accurately via embedding in a low dimensional space. This opens opportunities for the use of machine learning based methods for estimating MI from real world datasets.

4.2 Introduction

Mutual information (MI) is a measure of statistical dependence between two variables [133]. It is a fundamental quantity in many different disciplines. It captures both linear and nonlinear associations, is reparameterization invariant, and is zero iff the variables are statistically independent. These and other qualities make it a tool of choice for data analysis applications in diverse fields [145, 112]. An even wider application of MI as a statistical analysis tool is hampered by the well known difficulty of estimating it from data. Indeed, for continuous variables X and Y, MI, measured in *nats*, is

$$I(X;Y) = \int dx\ dy\, p(x,y) \ln \frac{p(x,y)}{p(x)p(y)}, \tag{4.1}$$

where x and y are specific values of the variables, $p(\cdot)$ are the probability density functions of their arguments, and the integration is over the domain of the variables (for discrete X and Y, the integrals are replaced by sums, and probability densities by probabilities). Because MI is a nonlinear function of p, an unbiased estimate of

p plugged into Eq. (4.1) results in a biased estimate of I. For typical situations, the bias of estimators is a bigger problem than the variance. This was noticed soon after MI was introduced [105], and many attempts have been made to design MI estimators that would correct this bias. We now know that, even for discrete variables, no estimator can be unbiased universally, for all underlying distributions, until the number of samples per possible outcome is large [115]. For continuous variables, the situation is even worse, since MI is reparameterization invariant, and so must be its estimators, while learning in a reparameterization covariant way is impossible [73].

Nonetheless, significant advances have been made in the field of MI estimation. For continuous variables, which is the focus of this work, the most commonly used estimator is by Kraskov et al. [93], and its later modifications [72]. These use the statistics of distances between neighboring data points to estimate the inhomogeneity of p and hence its MI. While no guarantees of convergence of any estimator to the true value can be made, a common heuristic has been developed [141]. It involves applying the estimator to a subset of the total data, varying the subset size, and verifying that the estimator does not drift statistically significantly as the subset increases towards the full data, thus signifying the absence of the sample-size dependent bias [138, 72]. Empirically, estimating MI using these approaches is only practical when the dimensionality of both X and Y is, at most, of order 10 [93, 71].

As the number of applications of MI has increased, and the progress of traditional methods has stalled, the need for better methods for estimating MI is now higher than ever. A promising development has been the use of neural networks (NN) methods to estimate MI via first estimating the deviation of $p(x, y)$ from the product of its marginals using NNs applied to the sampling data [15, 16, 109, 43, 122, 136]. Nominally, these methods can work even for very large dimensional data. However, in practice, they suffer from multiple drawbacks. First, most of these methods have been tested only on synthetic data with simple multivariate dependence structures and essentially infinite number of samples. Hence their ability to estimate MI in real world scenarios is unknown. Second, since universally good MI estimation for continuous variables is impossible, it is essential for any estimator to have internal consistency checks, which would signal to the user whether the output can or cannot be trusted. Such checks are either at their early stage of development or have not been developed for the NN estimators at all [41, 136]. Finally, NN estimators depend on a number of hyperparameters, such as criteria for stopping the training. How to choose these parameters to get an unbiased, low-variance estimate remains unknown.

In this work, we systematically study NN-based MI estimators. We apply them to synthetic and real world datasets to illustrate their strengths and limitations. We start with multivariate Gaussian data, which is the traditional testing setup. Some methods fail even for these simple cases, especially if the dimensionality of data increases, suggesting that they cannot be trusted for nonlinear, real-world datasets. We observe that the successful methods overcome the curse of dimensionality in MI estimation by first explicitly constructing a low-dimensional embedding of the data and then estimating MI in this lower-dimensional embedding space. We develop a protocol for choosing optimal hyperparameters for NN estimators and for checking if an estimator is biased. We show that systematically treating NN estimators as a dimensionality reduction problem addresses many challenges inherent in existing approaches. Overall, we show that this approach can estimate MI when the number of samples is as small as of the same order as the number of dimensions in the variables X and Y (not exponential in them!), provided an efficient low-dimensional representation of data can be constructed. Further, it is easy to check if the output of the estimator can be trusted.

4.3 Background and Previous Work

4.3.1 Estimation of Mutual Information

Estimating MI from finite data is a challenge, as discussed above. The magnitude of the problem can be understood from a simple argument. Suppose that we estimate MI for continuous variables, with each of the components of the variables bounded in a range A. Suppose further that the distribution $p(x, y)$ is smooth, so that the linear size of its smallest feature is a. Then the MI will be estimated well when the number of samples is $N \gg (A/a)^K$, where K is the joint dimensionality of X and Y, $K = K_X + K_Y$. Even for smooth probability distributions, where A/a is only slightly larger than 1, one needs $N \sim \exp(K)$ samples to estimate MI accurately—the usual curse of dimensionality. Not knowing the parameterization, in which p is smooth, or having unbounded variables makes the problem even harder. While the community has developed many MI estimation methods for continuous variables, none have been able to break this curse and work with dimensions larger than $K \sim 10$ [93] (see [72], which has pushed this limit). In contrast, most modern data are high-dimensional: for example, images have thousands of pixels, or one can record activity of thousands of neurons.

This inability of traditional methods to deal with the curse of dimensionality gave rise to NN based approaches. Deep NNs can capture complex nonlinear dependencies in large-dimensional data, sometimes from surprisingly few samples [95]. In the context of MI estimation, the class of so called variational deep learning methods has proven to be the most useful [15, 16, 109, 43]. The basic idea is simple: we have no access to either the joint probability distribution $p(x, y)$ or the marginals $p(x)$, $p(y)$, and we only have samples from them. However, we can transform the problem of estimating MI, that is, of evaluating the integral in Eq. (4.1), into a problem of evaluating what's known as the *critic* and the *normalization*. For example, for the MINE estimator of MI [17], we write $p(x,y) = p(x)p(y)e^{T(x,y)}/\mathcal{Z}_{\text{norm}}$. Here $T(x,y)$ is the critic, parameterized by a neural network that takes in samples of x and y and returns a single number $T(x,y)$. And $\mathcal{Z}_{\text{norm}} = \int dx\, dy\, p(x)p(y)e^{T(x,y)}$ is the normalization. While one cannot guarantee that $p(x,y)$ with approximate critic and normalization will normalize properly, one optimizes this approximation over all models implementable by a NN — hence the variational nature of the approach. With the estimates of T and $\mathcal{Z}_{\text{norm}}$ available, the integral in Eq. (4.1) is then evaluated by Monte Carlo sampling.

Other NN methods for MI estimation change the way of parameterizing the normalization factor (such as NWJ [109], Improved MINE [122], clipped MINE – SMILE [136], etc.) Some change the training protocol, such as by reformulating the problem as a contrastive learning problem (e. g., InfoNCE [151]).

4.3.2 Overview of NN information estimators

Here we briefly introduce NN-based MI estimation methods analyzed in this work. A more holistic review can be found in Refs. [122, 136].

MINE & SMILE [17, 136] both use the above mentioned critic factorization of $p(x,y)$ resulting in the estimator:

$$I_{\text{MINE}}(X,Y) \geq \mathbb{E}_P\left[T(x,y)\right] - \log\left[\mathbb{E}_Q\left(e^{T(x,y)}\right)\right], \tag{4.2}$$

where the first expectation is over the empirical joint probability density, and the second over the product of the empirical marginals. This Monte Carlo sampling leads to biased gradients, which is typically mitigated via a weighted running average over batches [17].

However, the MINE estimator can have a large variance. To solve this problem, the SMILE estimator clips the joint to marginal density ratio between $e^{-\tau}$ and e^{τ}, where τ is some parameter:

$$I_{\text{SMILE}}(X, Y) \geq \mathbb{E}_P[T(x, y)] - \log \left[\mathbb{E}_Q \left(\text{clip}(e^{T(x,y)}, e^{-\tau}, e^{\tau})\right)\right] . \tag{4.3}$$

Smaller τ decreases the variance, but at a cost of a larger bias. At $\tau \to \infty$, SMILE reduces to MINE. In what follows, we use $\tau = 5$.

InfoNCE [151] uses NNs to estimate the conditional distribution $p(y|x)$ as a function of x and y, which is the critic $T(y, x)$. The resulting estimate is

$$I_{\text{InfoNCE}}(X; Y) \geq \mathbb{E}_P \left[\sum_i^n \log \frac{T(y_i, x_i)}{\frac{1}{K} \sum_{j=1}^n T(y_i, x_j)}\right] , \tag{4.4}$$

where the expectation is over the empirical joint distribution, and n is the batch size. This is a form of contrastive predictive coding. The estimator is more biased than the others considered here but also has less variance. The largest MI value InfoNCE can output is the logarithm of the batch size used for training [122, 136].

Different critics. We consider two types of critics, separable [12] and concatenated [69], which correspond to different factorizations of the critic function $T(x, y)$. Separable critics are of the form $T(x, y) = g(x) \cdot h(y)$, where the functions g and h are implemented via NN embedding into a space of reduced dimension k, i. e. $g : X \to \mathbb{R}^k$, and $h : Y \to \mathbb{R}^k$. Concatenated critics use a single NN that takes in the concatenated x and y and produces one output per pair, i. e. $T : X \times Y \to \mathbb{R}$. A third combined type of critic, often called a bilinear critic [151, 64, 146], is of the form $T(x, y) = f(g(x), h(y))$. Similar to the separable critic, the functions g and h are implemented via neural network embeddings into a space of reduced dimension k. This is further passed to another concatenated layer(s), similar to the concatenated critic, which produces one output per pair, i.e., $T : X \times Y \to \mathbb{R}$.

The choice of critic is important [148, 58]. For example, the separable critic allows for changing the number of embedding dimensions k for both X and Y, which plays a significant role. If k is smaller than the intrinsic, latent dimensionality of K_X for X (or K_Y for$Y$), it is likely that the estimator will not capture the full amount of information between the variables. On the other hand, if k is very large, it could lead to what we call "information washout", where a small amount of information is distributed among many dimensions, making it indistinguishable from statistical noise in all of them, and hence increasing the variance of the estimator dramatically. Additionally, we will show later that the saturation of information for intermediate values of k signals the success of the estimator in capturing the mutual information. Another potentially important aspect in choosing a separable vs. a concatenated critic is that the former allows the estimator a better chance of capturing the information when variables X and Y have drastically different internal structures (e. g., different entropies), without variability in one variable drowning that in the other, as could happen for the latter. On the other hand, the concatenated critic is more general, allowing for broader relations between the embeddings. For example, in a concatenated critic, the first dimension of X can be mixed with the second dimension of Y, while, in the separable critic, the first dimension of $g(x)$ interacts only with the first dimension of $h(y)$. However, this mixing comes at the cost of not being able to change the internal dimensionality of the embedding, an important test as we will discuss. Bilinear critics could address this issue by having two separate spaces before allowing them to mix.

Overall, the choice of the critic depends on the question one is asking and *a priori* knowledge about the structure of the data set in question. In this work, we do not aim at a comprehensive evaluation of different critics, and hence focus only on the two basic critics, the separable and the concatenated ones. Similarly, our goal is not a comprehensive comparative analysis of different NN estimators. Thus, even though we explored many NN estimators, including NWJ [109], Jensen-Shannon [110], Donsker and Varadhan [43], and TUBA [16], we focus on SMILE and InfoNCE, relegating others to SI. We make this choice since (i) these estimators provide examples of high variance/low bias (SMILE) and high bias / low variance (InfoNCE); (ii) none of the other estimators performed uniformly better than these two; and (iii) some of the other methods fail additivity and other consistency checks that MI estimators must obey [136].

4.3.3 Problems of NN mutual information estimators

While NN methods for MI estimation have become popular, the community has not yet addressed serious concerns about them: NN methods are rarely tested on real-world (non-Gaussian) data with only a finite number of samples available.

To be able to compare the estimates to a known correct answer, a typical (and sometimes the only) test bed is data drawn from relatively low-dimensional ($K_X, K_Y \sim 10$) Gaussian distributions, where the true MI can be calculated analytically. However, such tests are woefully insufficient. First, simpler, traditional methods [93, 72] would be sufficient for $K \sim 10$, for either Gaussian or non-Gaussian data. Testing must involve large dimensional data, $K \gtrsim 100$, where traditional methods start failing. Yet, only a few NN methods have been systematically tested in this regime.

Further, for Gaussian data, MI and $X - Y$ correlation matrices are related via analytical expressions. Such correlation-based MI estimation sets a natural benchmark for optimal MI estimation (correlation matrices can be estimated accurately when $K/N \ll 1$ [24]). If a correlation-based approach works, but NN methods do not, the latter are incapable of optimally utilizing the data. This would make it exceedingly unlikely that NN methods would be able to produce good MI estimates for more complicated, non-Gaussian data at similar sampling ratios. As we show below, most NN methods perform worse—sometimes much worse—than correlation-based estimation, so that sweeping generalizations about their accuracy are hardly warranted.

Finally, NN based MI estimators are typically validated using effectively infinite data [122, 136]. While unlimited data during training is useful for establishing asymptotic consistency of methods, it removes overfitting, unrealistically enhancing the methods' performance. As mentioned above, MI estimators suffer from sample size dependent biases. Thus, success with infinite data does not guarantee success on real-world, finite size datasets. In practice, one often has $N \sim K$. This renders tests conducted in an infinite data regime not very useful. We will show that it is possible to produce unbiased MI estimators even for some of these severely undersampled, high dimensional cases, provided the data can be accurately embedded into a low dimensional space.

4.3.4 MI Estimation as a DR problem

As argued in Sec. 4.3.1, the accuracy of estimators depends on the dimensionality of the data. Thus, to increase the accuracy, a natural approach is to reduce the dimensionality of X and Y to low dimensional descriptions Z_X and Z_Y, respectively, and then estimate $I(Z_X; Z_Y)$ as a proxy for $I(X;Y)$. By the data processing inequality [133], $I(Z_X; Z_Y) - I(X;Y) \equiv \Delta I \leq 0$, save for possible statistical fluctuations. How tight this probabilistic bound is depends on the quality of the dimensionality reduction (DR). Compressing X and Y independently may result in keeping the variation in each of the variables, but not the covariation, resulting in large $|\Delta I|$. The analysis in the preceding Chapters of this dissertation suggests that, to avoid this problem, one needs to compress the variables *simultaneously* [101, 3], while maximizing $I(Z_X; Z_Y)$, and hence decreasing $|\Delta I|$.

While, to our knowledge, this has not been emphasized in the literature previously, the critics in the NN based estimators reviewed in Sec. 4.3.2, indeed, internally perform SDR for the data X and Y. This is the easiest to see in a separable critic, for which one trains two networks to perform the following reductions: $Z_X = g(X)$ and $Z_Y = h(Y)$. The critic then is $T(X,Y) = g(X) \cdot h(Y) = Z_X \cdot Z_Y$, which is a specific choice that effectively enforces orthogonality of the embeddings: the ith component of g is allowed to have information only with the ith component of h. This approach is similar to other SDR methods we considered in the previous Chapters of this Dissertation, and, specifically, the Deep Variational Symmetric Information Bottleneck (DVSIB) [2]. The key difference is that DVSIB uses variational probabilistic encoders, rather than deterministic, feedforward networks as used in the separable critic. Additionally, DVSIB includes two reconstruction decoder networks, $X = X(Z_X)$ and $Y = Y(Z_Y)$, which are not present in the MI estimator (though see Chapter 5 for discussion of the removal of the reconstruction from the DVSIB architecture).

The analogy with SDR for the concatenated critic is less clear. However, even here, the NN still maps the combined vector $\{X, Y\}$ into a lower-dimensional vector space Z simultaneously, from which then the critic is evaluated in the output layer of the NN. This approach is a deterministic analog of another SDR method, the Joint Deep Variational Canonical Correlations Analysis (DVCCA) [2, 158]. A clearer analogy can be observed with the bilinear critic, where two networks are used to compress X and Y, but they are jointly combined into one variable Z. This can be viewed as a deterministic mapping similar to the original Deep Variational Canonical Correlation Analysis (DVCCA) [158].

4.4 Results

We start with the case of infinite training data, which is the common way of testing NN-based MI estimation algorithms. Starting with low-dimensional, Gaussian data and progressing to high-dimensional, nonlinear data, we already observe some of the pitfalls of NN methods. We then show that none of the available algorithms can naively be trusted for the real-world-like, high dimensional regime with limited data. We then propose to use a more careful implementation of the NN based estimators to explicitly change the dimensionality of the low-dimensional embedding space, in which we are maximizing MI between the embeddings of X and Y. With additional self-consistency checks, this solves the problem of estimating MI from large-dimensional, undersampled data.

4.4.1 Infinite Data

4.4.1.1 Low-dimensional data

A typical test case for NN based MI estimation uses samples of X and Y from correlated Gaussian distributions with $K_X, K_Y = 5 \dots 20$ dimensions. Typically, there are no correlations among components of X or Y, but each X_i is correlated with Y_i with a correlation coefficient ρ. A new batch of data, $\mathcal{O}(10^2 \dots 10^3)$ samples, is generated at each training step, typically for $\mathcal{O}(10^3 \dots 10^4)$ steps, resulting in an unrealistic number of samples (typically $N \sim \mathcal{O}(10^6)$). This might seem like a small dataset for modern machine learning tasks. However, for many physical and biological applications, obtaining a dataset of such a size is prohibitively hard. Consequently, $K/N \to 0$, and many methods can perform well in these tests (although not all do). The performance is shown in Fig. 4.1A,B for SMILE and InfoNCE. (This Figure is similar to standard figures shown in [17, 136], for example). Both methods work well for small MI values, though SMILE exhibits a large variance, suggesting that averaging MI estimates over multiple steps of training is essential. At larger MI, SMILE starts developing a bias, but InfoNCE completely saturates at ln(batch size), Fig. 4.1B, confirming that it cannot be trusted in this regime. Additional results for other methods can be found in the SI (Fig. 4.6)

This relative success of NN methods should come as no surprise. As shown in Fig. 4.1A,B, for jointly Gaussian data, we can calculate MI from the empirical correlation matrix of X and Y, $I = \frac{1}{2} \ln \frac{|C|}{|C_{XX}|\,|C_{YY}|}$, where C is the joint correlation matrix, and C_{XX} and C_{YY} are the marginal correlation matrices (numerically, one needs to carefully remove directions with zero correlations before computing the

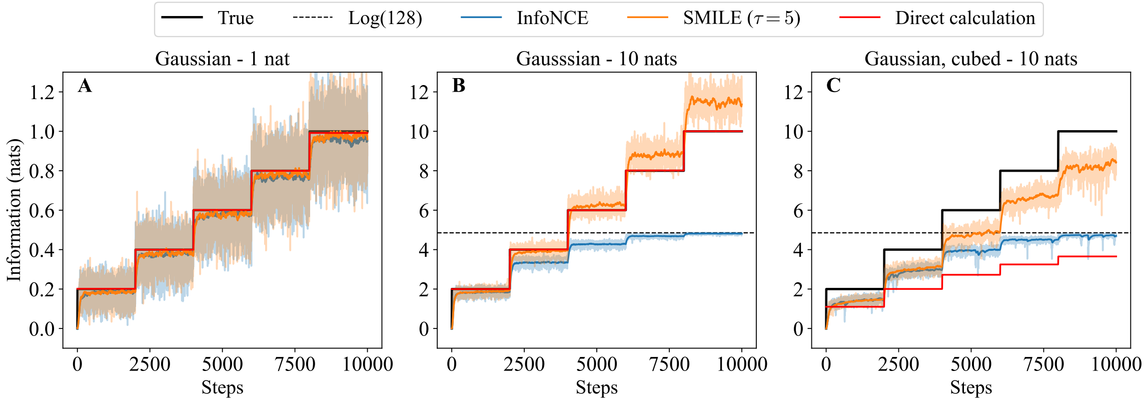


Figure 4.1: **MI estimation for low-dimensional distributions.** We use a common "staircase" protocol for exploring the estimation for different MI values. Here MI jumps after every 2000 steps of training, with every step consisting of a batch of 128 samples. The maximum information is 1 nat in panel (A), and 10 nats in panels (B) and (C). We sample from correlated Gaussian distributions with $K_X = K_Y = 10$ and choose the correlation coefficients that result in the needed MI. All panels show the true information, estimates I_{InfoNCE}, I_{SMILE} ($\tau = 5$) with a concatenated critic (with 2 hidden layers, each with 256 neurons), and the MI estimate from the empirical correlation matrix (denoted as Direct calculation). The correlation estimate uses data from all of the steps preceding the current one for a given MI value. With just one step, the correlation-based MI estimate is hard to distinguish from the true MI. For NN methods, we show their value within each training step (thin lines) and an average smoothed over 100 steps (thick lines). (A) shows that, when the true MI is small, all methods work well. When MI is high, (B), correlation-based estimation still works, while NN methods degrade: SMILE overestimates and has a large variance, and InfoNCE saturates at $\ln 128$ nats (logarithm of the batch size). In (C) we add a cubic nonlinearity to the data (see text). Now correlation-based method, non-surprisingly, underestimates MI. However, the effect of the nonlinearity on NN methods is weaker.

determinants). The error of MI estimation based on the correlations is $\sim K/N$ [24], and it is empirically negligible compared to the bias and the variance of NN methods, Fig. 4.1A,B, especially for large MI. While the correlation-based MI is the optimal benchmark, which NNs cannot hope to match, the relatively large discrepancy between the optimum and the NN estimators is a red flag: large variance and bias for these simple data suggest that outputs of NN-based estimators on more complex datasets should be suspect.

Cubic nonlinearity. To test how MI estimation methods behave on nonlinear data, we reparameterize the data with an injective continuous nonlinear transformation.

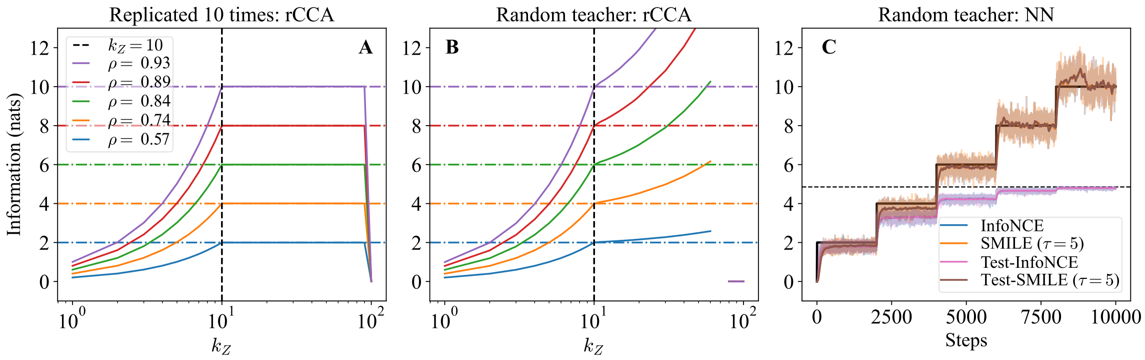


Figure 4.2: **MI estimation for high-dimensional data.** (A) rCCA was applied to the 100 dimensional X and Y, each consisting of 10 copies of data from Fig. 4.1B, to produce the reduced descriptions Z_X and Z_Y. $I(Z_X; Z_Y)$ estimated via the correlation matrix is plotted. As the dimensionality of the reduced space, k_Z becomes larger than 10 (the number of independent X and Y), it is possible to represent all correlations in the Z_X, Z_Y space, and the MI reaches the correct value. (B) Similar analysis for the random features model, produced from 10 independent X' and Y'. As the number of linear embedding dimensions increases past 10, there is no obvious saturation, and the MI keeps growing since additional dimensions allow to focus on different parts of the nonlinear relations among the variables. (C) InfoNCE and SMILE with a concatenated critic (with 2 hidden layers, each with 256 neurons) on a random features model, with a staircase increase in MI values, as in Fig. 4.1. Here both X' and Y' are 100 dimensional. Compared to Fig. 4.1C, bias of SMILE nearly disappeared, likely because nonlinearities are weaker, and because SMILE can now use the overcomplete data to estimate correlations more precisely. Additionally, we show the results of evaluating both methods on a fresh batch of test data, not used in training. The test and the training curves nearly overlap—in this effectively infinite data regime, there is no overfitting.

Specifically, we keep X unchanged and set $Y \to Y^3$. While the mutual information remains constant under this transformation, the linear correlation changes. This results in the failure of simple linear methods, Fig. 4.1C, underscoring the need for nonlinear NN approaches. At the same time, NN methods degrade only a little compared to Gaussian data: both SMILE and InfoNCE have similar variances and somewhat larger biases compared to the Gaussian case, Fig. 4.1B.

4.4.1.2 High dimensional data

Oversampling. Typical modern experiments have $K \gg 1$, with $K \sim N$. However, the data often can be approximated well with low-dimensional models [47, 10, 159, 137, 114, 108, 70]. How NN based estimators perform for such high dimensional, but intrinsically simpler data is unknown. We start exploring this with a simple case: starting with 10-dimensional X' and Y', as in Sec. 4.4.1.1, we replicate the variables ten times each into X and Y, respectively, so that $K_X = K_Y = 100$. Note that $I(X;Y) = I(X';Y')$. A standard approach for high dimensional data is to perform DR on X, Y and then to estimate the correlation matrix and MI $I(Z_X;Z_Y)$ in the reduced space. For retaining shared information between datasets, Simultaneous DR, such as regularized CCA (rCCA)[2] [76, 152, 171, 3] can be used. Figure 4.2 shows that, indeed rCCA applied to these data results in accurate MI estimation.

Random features model. We generate nonlinear, yet low-dimensional data using teacher NNs. Specifically, we take the 10-dimensional data from Sec. 4.4.1.1, pass them through a NN with one fully connected hidden layer with 1024 neurons and a sigmoidal nonlinearity, which then eventually outputs a $K = 100$ dimensional X' and Y'. Synaptic weights and biases of this teacher NN are initialized with default Pytorch initialization. rCCA, predictably, fails on these data (Fig. 4.2B): as we increase the number of embedding dimensions k_Z, the reduced variables focus on different parts of the joint probability distribution $p(x, y)$, overestimating MI. In contrast, both SMILE and InfoNCE estimate MI well, Fig. 4.2C, even better than in Fig. 4.1C, presumably because the large K allows many ways to detect all statistical structures. As always, for large MI, InfoNCE saturates. Finally, Fig. 4.2C, compares MI values on training and new, test data—in this effectively infinite data regime, there is no overtraining

[2] Regularized canonical correlation analysis (rCCA) is a technique that finds linear combinations of two sets of variables that are maximally correlated (cf. Sec 2.4.1.3.2). We fit an rCCA model from [33] to the equivalent training samples (batch size x number of steps per correlation value in the staircase setup) of data, specifying that the model should yield k_Z dimensions. This allows us to calculate the correlation matrix in the reduced space. When calculating the correlation, it is important to be cautious, especially if $k_Z > K_Z$ (K_Z being the true latent dimensionality of X and Y), as we might have dimensions that contribute very minimally to the correlation. If we calculate the determinant of the correlation matrix as a product of its singular values, these small correlations can lead to numerical instabilities. Therefore, we employ a threshold (typically $10^{-6} \sim 10^{-10}$) to disregard any contribution from singular values below this threshold.

and no difference between the two.

4.4.2 Finite data

Figure 4.2 shows that NN methods can estimate MI reliably even for some high-dimensional distributions with nonlinear dependencies. However, Fig. 4.1 illustrates that NN methods do not utilize the available data optimally. Since estimating MI is a hard problem *precisely* because datasets are finite, it is crucial to understand how the estimator performance depends on the amount and the structure of the data. We explore this on a real and a synthetic dataset, with an even larger dimensionality of X and Y, $K_X = K_Y = 784$, and by varying the amount of data N available for training ($N \sim \mathcal{O}(10^5)$).

Noisy MNIST. We use the Noisy MNIST dataset, introduced in Ref. [2] as an adaptation of Refs. [94, 157, 158]. These data comprise two distinct views of data, X and Y, each of dimensionality $28 \times 28 = 784$ pixels, as shown in SI Fig. 4.5. The first view is an image of a digit from the standard MNIST dataset, subjected to a random rotation by an angle uniformly sampled between 0 and $\frac{\pi}{2}$, and to a scaling by a factor uniformly distributed between 0.5 and 1.5. The second view consists of another image with the same digit identity (but different instance) with an additional background layer of Perlin noise [120], with the noise factor uniformly distributed between 0 and 1. Both views are normalized to an intensity range of $[0, 1)$ and then flattened to form an array of $K_X = K_Y = 784$ dimensions. The dataset comprises a total of $55996 \sim 56k$ images for training and $\sim 7k$ images for testing. Overall, the only correlations between the two views are via the digit class, and all ten digits are represented nearly uniformly, so that $I(X; Y) \approx \ln 10$. Further, correlations are strongly nonlinear and dimensionality is high, so that the dataset is a natural testbed for MI estimation. A typical training and testing curves for InfoNCE with separable and concatenated critics are shown in Fig. 4.3. Panel A in Fig 4.3 shows that MI

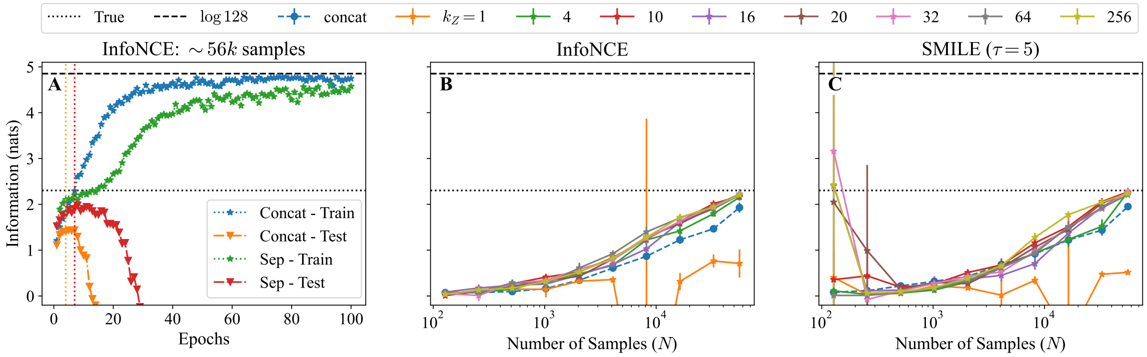


Figure 4.3: **MI estimation with limited data.** (A) Training and test curves for InfoNCE evaluated on a Noisy MNIST dataset as a function of training time, measured in epochs. The blue and orange curves represent a concatenated critic (with 4 hidden layers, each with 1024 neurons), while the green and red curves represent a separable critic (also with 4 hidden layers, each with 1024 neurons, and an embedding dimension of size 16). The vertical dotted orange and red lines indicate the points where the test information is maximized for each critic, and the corresponding training value is used as the heuristic value for the best estimate of information. The horizontal dotted line represents the theoretical value of the information, ln(10), and the horizontal dashed line represents the maximum value that InfoNCE can achieve, ln(batchsize). (B) We use maximum test information heuristic to estimate MI as a function of sample size with the InfoNCE estimator with different critics. The concatenated critic is shown with blue circles, while the separable critic, which varies by the number of embedding dimensions, is shown with squares in different colors. (C) Same as (B), but for the SMILE estimator ($\tau = 5$). In both (B) and (C), it the information increases with the number of samples N used for training, reaching the true value at large N. Additionally, the number of embedding dimensions has minimal effect beyond 4, suggesting that at least four dimensions are necessary to capture all the information between X and Y. We show means and standard deviations for all estimators, calculated from five independent trials for each data point.

estimates on training and test data diverge during training, well before the estimates reach true MI value, indicating overfitting. Thus, it is important to stop the training early. We do this by reporting the training value which is obtained when MI estimate on the test data peaks, which we denote as the *max test* heuristic[3].

Panels B and C show the mean MI values, averaged over five trials, based on the maximum test heuristic evaluated at different numbers of training samples for InfoNCE and SMILE, respectively. Both critics used for the estimators are built with 4 hidden layers, each containing 1024 neurons. Additionally, the separable critic, which allows for varying the embedding dimension, is tested from embedding dimension of 1 up to 256, for both X and Y. Apart from the significant fluctuations observed when the separable critic embedding is restricted to only one dimension or when trained on a very small number of samples, we observe an increase in MI closer towards the true value with more training samples, which is expected. Notably, we observe that varying the size of the embedding for the separable critic results in a saturation of MI after $k_Z = 4$. This indicates that $k_Z \geq 4$ is necessary to capture all the information between X and Y in this specific dataset. When the embedding dimension, k_Z, is too small to capture the underlying data structure, MI is underestimated by all methods. When $k_Z \geq 4$, and for as large as $k_Z = 256$, information estimates are accurate, as long as $N \gg K$. Note that since $\ln(10)$ is smaller than the logarithm of the batch size, InfoNCE and SMILE would work well, though InfoNCE has a smaller variance, as expected. This observation can be used as a consistency check for the estimators: changing the dimensionality of the embedding space should eventually lead to saturation, reflecting the intrinsic dimensionality of the data, and observing this is one evidence of accuracy

[3] One might also think of other heuristics. For example, we can consider a *zero test* heuristic, in which we choose to report the train value that corresponds to the zero test value. This means that we have no information left between X and Y and essentially cannot trust any train value beyond this point. This approach might be useful for downstream tasks where we want to stop our training once we have captured all possible information. However, we observe that this is not a good heuristic for accurately estimating the information, as it is likely to result in overfitting. We have also found this to be true empirically.

of the estimation. Performing this check via adjusting the embedding size is a key advantage of using a separable critic, and aligning the dimensionality of the embedding method with the dimensionality of the data is important for successfully capturing all relevant shared information [3].

Large dimensional Gaussians with varying number of signal dimensions.

The above results demonstrate the success of the estimators in accurately capturing the MI in nonlinear higher-dimensional spaces, for a relatively high number of samples[4]. We can also see that changing the number of embedding dimensions for the separable critic can be a valuable tool in detecting the intrinsic dimensionality of the system. However, it is not yet clear why these methods work in the first place, given that we are using few samples, $\mathcal{O}(10^4 - 10^5)$, for $\mathcal{O}(10^3)$ dimensions. We hypothesize that the reason for the success is that NN MI estimators perform SDR (in two distinct spaces, as for the separable critic, or in one joint space, as for the concatenated critic). We demonstrate that existence of an accurate low-dimensional representation of data is essential for MI estimation by NN methods to work in Fig. 4.4. For this, we generate two Gaussian variables X and Y with 784 dimensions each and a fixed amount of information $I(X;Y) = \log 10$ nats, to mimic the MNIST dataset. However, this amount of information is distributed among a different number of correlated components in different examples, $K_Z = 2^0, 2^1, \ldots, 2^9$. The correlation in each component is chosen such that the total amount of information sums to the same desired $I(X;Y)$. We test this setup at a relatively high number of samples, $N = 10^5$, to eliminate the effects of finite sampling and verify whether the estimators can find the same amount of information if distributed among more or fewer intrinsic latent dimensions. Similar to Fig. 4.3, we evaluate the estimators with concatenated

[4] It is hard to clearly conclude in this example if the estimators are unbiased, as we ran out of samples. However, if we are to generate more samples, we should expect asymptotic behavior as N increases.

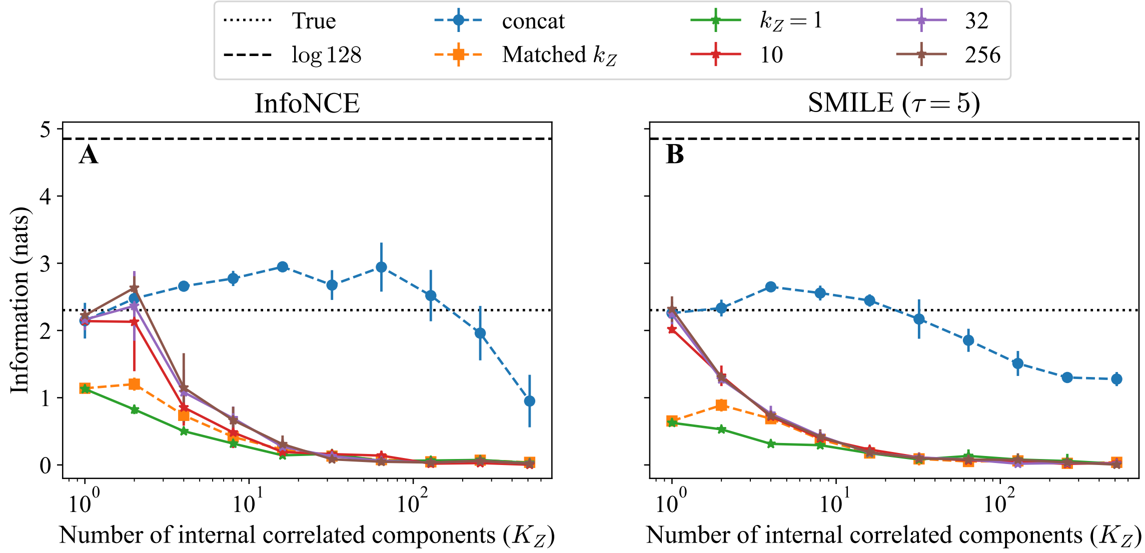


Figure 4.4: **Large Dimensional Gaussian with Varying Number of Signal Dimensions.** The theoretical value of information, log(10), is shown as a dotted line, and log the batch size, log(128), is shown as a dashed line. The plots show the performance of InfoNCE and SMILE ($\tau = 5$) estimators in capturing MI in high-dimensional Gaussian data. We generated two Gaussian variables X and Y, each with 784 dimensions, and a fixed amount of information $I(X;Y) = \log 10$ nats, distributed among a varying number of correlated components $K_Z = 2^0, 2^1, \ldots, 2^9$ on the x-axis. For each setup, we evaluated the estimators using both concatenated and separable critics. The concatenated critic (blue circles) uses a single embedding space, while the separable critic (squares) uses embeddings of different fixed sizes $k_Z = 1, 10, 32, 256$ and a matched size corresponding to the true dimensionality of the data. The plots demonstrate that the InfoNCE estimator effectively captures MI when the information is distributed across fewer than 10 dimensions for the separable critic. However, its performance deteriorates when the information is spread across 32 or more dimensions. Conversely, the concatenated critic shows robustness, with a noticeable decline in performance only at $K_Z \sim 128$ dimensions.

critics, fixed-size embeddings for separable critics $k_Z = 1, 10, 32, 256$, and also when matching the embedding size of the critic with the true dimensionality of the data (denoted as Matched, $k_Z = K_Z$). We see that, indeed, for a separable critic, the estimators capture MI effectively if it is distributed in $K_Z < 10$. However, when the information is distributed among 32 dimensions or more, the estimators fail for various embedding sizes. Surprisingly, the concatenated critic starts failing similarly only around $K_Z \sim 128$. We suspect that this is because the concatenated critic freely mixes the different components of the variables, without imposing a dot product structure in the latent space, as separable critics do, allowing the critic to have more freedom in capturing small amounts of information distributed among a relatively high number of correlated dimensions.

4.5 Discussion and MI Estimation Guidelines

4.5.1 Guidelines for MI estimation from high-dimensional data

The analysis above leads to suggesting the following practical guidelines for estimating MI from high-dimensional data. This is adapted from Ref. [72], using the procedures developed there to test for the self-consistency of the estimators, and hence possible bias.

1. Use rCCA; evaluate MI based on the correlation matrix in k_Z-dimensional embedding space; evaluate $I_{\mathrm{EST}}(k_Z)$. If saturation is observed at high k_Z, report the value. If not, then the data is nonlinear; proceed to the next item.

2. Partition the data into training/test sets (90%/10%). Use a NN estimator of choice for I_{EST}. If the amount of information is expected to be low (less than ln(batchsize), using InfoNCE is preferred. If not, SMILE is usually a good

option, but pay attention to the variance. Stop training at maximum of MI on test data.

3. Vary k_Z and the amount of data analyzed, which can be achieved by subsampling (without replacements, see [72]) from the full dataset. Perform multiple training runs for each (k_Z, N) pair, varying the subsample and the random seed for training and estimate the standard deviation over the runs, $\sigma_I(k_Z, N)$.

4. Choose $\hat{k}_Z$ in the range where $I_{\text{EST}}(k_Z)$ is stable within σ_I (sufficient expressivity, but no undersampling).

5. If I_{EST} is stable over k_Z and N, report $I_{\text{EST}}(\hat{k}_Z, N)$ as the MI estimate. Otherwise, no reliable MI estimate has been found.

4.5.2 Discussion

MI is a difficult quantity to estimate, but it is an important quantity to estimate well. New NN estimators are a promising direction, addressing different regimes of the bias/variance tradeoff for the estimation, which were unaccessible with previous methods. However, MI estimation is a hard problem for finite data, and nonlinear data (otherwise use linear methods!), and these estimators have not been tested for these cases. Here we showed that all standard NN based neural estimators fail when the dimensionality of data increases, well within the range of today's experimental datasets. We have shown how one can view the estimators as an SDR technique, reducing the dimensionality of the data and then estimating MI with NN estimators in the reduced space. We have also provided heuristics for verifying whether the estimators' output can be trusted. For this, one needs to verify the stability of the output to the hyperparameter (the number of embedding dimensions) and to the amount of data (via varying the number of training samples), at least when the size of the training subsample and the full sample are not drastically different.

Finally, we made the first steps towards explaining how MI estimation—which is provably hard—can be reliable for large-dimensional data, even with relatively small sample sizes. We argue that reliable estimation is only possible when the data admits a low-dimensional latent structure. This is also supported by the argument in Ref. [101] that fluctuations in estimation of mutual information scale in proportion to the size of the latent space, and not the observable space, though this analysis was done for discrete variables only. We hope that additional tests will support this hypothesis, conclusively proving that it is possible to reliably estimate MI in the undersampled regime $K > N$, as long as the true latent dimensionality of the data is $K_Z \ll N$.

4.6 Supplemental Information

4.6.1 Neural networks architecture and Technical details

The estimators' implementation is adapted from Ref. [136]. The critics are implemented using Multi-Layer Perceptrons (MLPs) with either 2 or 4 hidden layers (in addition to the input and output layers), ReLU activations, and Xavier uniform initialization [53]. For the separable critic, two networks are trained separately but simultaneously for X and Y, with input dimensionalities K_X and K_Y, and output dimensionality k_Z. For the concatenated critic, a single network is trained with an input dimensionality of $K_X + K_Y$ and an output dimensionality of 1. The estimators are trained using the Adam optimizer [86] with a learning rate of 5×10^{-4}. The estimators' implementations are done in PyTorch [116] and trained on various GPUs. The rCCA models are trained using the library in Ref. [33], where the regularization c factor is fine-tuned to produce the highest test value, and the corresponding train value is reported. Parameters not explicitly mentioned are set to their defaults.

4.6.2 Supplemental figures

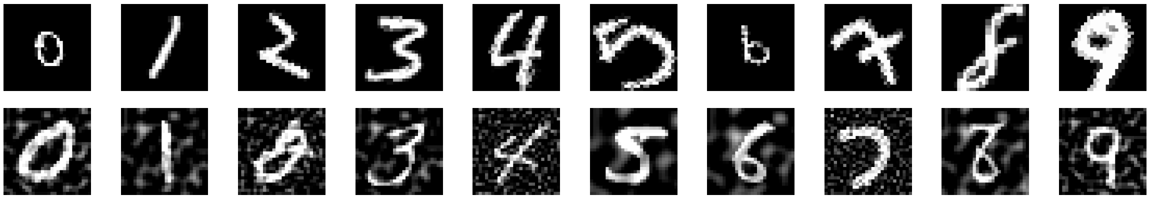


Figure 4.5: **Samples from Noisy MNIST data set** [2]. The data set contains $\sim 56k$ training and $\sim 7k$ test digit pair, X and Y. In each pair, the same digit (but its two different instances) are corrupted by scaling / rotation, X, and by background noise, Y.

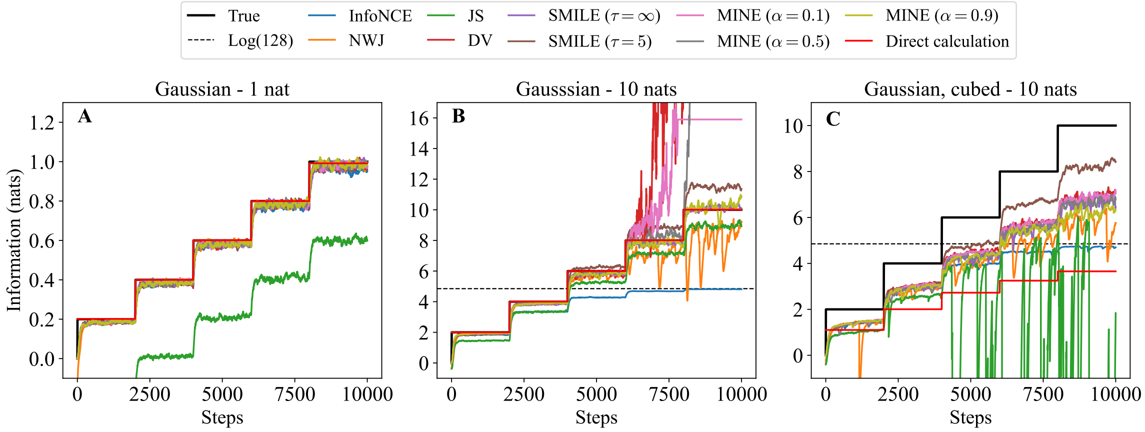


Figure 4.6: **Performance of NN-based methods for MI estimation.** An extension to Fig. 4.1. Where we used a common "staircase" protocol for exploring the estimation for different MI values. Here MI jumps after every 2000 steps of training, with every step consisting of a batch of 128 samples. The maximum information is 1 nat in panel (A), and 10 nats in panels (B) and (C). We sample from correlated Gaussian distributions with $K_X = K_Y = 10$ and choose the correlation coefficients that result in the needed MI. All panels show the true information, estimates I_{InfoNCE}, I_{JS} [110], I_{NWJ} [109], I_{DV} [43], I_{SMILE} ($\tau = 5, \tau = \infty$) [136], I_{MINE} ($\alpha = 0.1, \alpha = 0.5, \alpha = 0.9$) [17] with a concatenated critic (with 2 hidden layers, each with 256 neurons), and the MI estimate from the empirical correlation matrix (denoted as Direct calculation). The correlation estimate uses data from all of the steps preceding the current one for a given MI value. With just one step, the correlation-based MI estimate is hard to distinguish from the true MI. For NN methods, we show their value for an average smoothed over 100 steps. (A) shows that, when the true MI is small, all methods work well (except JS). When MI is high, (B), correlation-based estimation still works, while some NN methods estimates are close to the correct value (MINE ($\alpha = 0.9$) and SMILE ($\tau = \infty$)), other methods degrade: DV, MINE ($\alpha = 0.1, \alpha = 0.5$) overshoots with an extremely large variance. SMILE ($\tau = 5$)) overestimates and has a large variance. JS underestimates, and NWJ underestimates with a large variance, and InfoNCE saturates at $\ln 128$ nats (logarithm of the batch size). In (C) we add a cubic nonlinearity to the data (see text). Now correlation-based method, non-surprisingly, underestimates MI. However, the effect of the nonlinearity on NN methods is weaker (except on JS which degrades greatly).

Chapter 5

Discussions

Here I aim to summarize the findings of the Dissertation, emphasizing the key insights from each of the preceding Chapters and to outline potential avenues for future research.

The main message of the Dissertation is as follows: In modern scientific data analysis, we often encounter high-dimensional and possibly multimodal datasets, where the number of samples is of the same order of magnitude as the number of dimensions. For such data, the goal is to find an interpretable, modelable low-dimensional representation via application of DR methods. The Dissertation demonstrated that DR is not merely a data preprocessing step, not requiring much thinking. Instead, the structure and the usefulness of the constructed low-dimensional representations of data depend on the DR method used. Thus one should carefully consider which DR method to employ, ensuring that it aligns with the structure of the data and the question we are addressing. For instance, if we seek a low-dimensional representation of a multimodal dataset that captures shared information between modalities (i. e., covariation), then Simultaneous DR methods will require less data and will produce more useful representations. Even simpler, if the underlying structure of a dataset suggests linearity, linear methods suffice to obtain comprehensive low-dimensional descriptions that capture all relevant information. Additionally, in this Dissertation, I showed that many DR methods can be systematized within a single generalized framework, the Deep Variational Multivariate Information Bottleneck (DVMIB), which facilitates the translation of dependency graphs for data compression and reconstruction into practical methodology. Furthermore, the development of the new method within this framework, Deep Variational Symmetric Information Bottleneck (DVSIB), holds promise as a valuable tool across various fields. Additionally, the Dissertation demonstrated that low-dimensional representation of data helps in the estimation of important statistics of the data. For example, we showed that we can reliably estimate Mutual Information (MI) in large K-dimensional datasets from only $N \sim K$ samples if the data, indeed, has a good low-dimensional latent approximation. This is a major improvement in the field of MI estimation, where, until recently, one expected to need exponentially large datasets to estimate information in high dimensional data.

The Dissertation opens new venues for subsequent research. Some of these venues are already being explored, showing promising results. As one of such examples, below I illustrate the utility of DVSIB on a simple, but exciting problem: discovering canonical coordinates for a dynamical system from experimental data. Namely, we will use time lapses of images of a physical pendulum to derive the angle and the angular velocity as the two dynamical variables for this system. Secondly, I discuss the utility of DVSIB-like methods in the field of neuroscience, and how they could be a valuable tool in understanding various brain-brain, brain-behavior, and behavior-behavior interactions. With our approaches, we can obtain low-dimensional representations that capture the most covariation between different modalities, allowing for better modeling and understanding of underlying phenomena.

Below I summarize the findings of each Chapter and how they relate to each other. Then I follow up with a deeper discussion of potential future research directions.

5.1 On Linear DR Methods

In Chapter 2, I presented several key findings regarding linear dimensionality reduction methods and the implications of using different methods for different scenarios. Firstly, I argued that the simple linear model that we studied can capture many realistic features of experimental datasets, along with its flexibility to accommodate more than two modalities of data. We established the superiority of SDR methods over IDR methods when the objective is to capture covariation between modalities rather than mere variation. Supported by empirical evidence and theoretical insights, we strongly advocated for the practical preference for SDR methods, particularly rCCA, in identifying shared signals across different data modalities. This assertion was further reinforced by our exploration of a nonlinear dataset, the noisy MNIST, and corroborated by various sources from the literature, collectively emphasizing the efficiency of SDR methods in extracting covariation. We also addressed the efficacy of SDR techniques in low sampling situations, underscoring the need to align the dimensionality of reduced descriptions with the actual dimensionality of the shared signals. Moreover, we introduced a diagnostic test for differentiating between shared and self signals in data, providing practitioners with a valuable tool for characterizing complex datasets. In summary, Chapter 2 supports what I believe to be a fundamental, albeit often overlooked, principle of data science: SDR methods outperform IDR methods in detecting shared signals. The Chapter also connects to the secondary theme of the entire Dissertation: in data analysis applications, it is important to match the analysis methods to the underlying structure of the data. Despite certain mentioned limitations, such as linearity of the methods (discussed in Chapter 3), and linearity of the metric (discussed in Chapter 4), by focusing on a linear mixing system, Chapter 2 provides an important intuition, which can be extended to nonlinear dimensionality reduction techniques and many practical applications.

5.2 On DVMIB

In Chapter 3, we introduced a new framework based on MIB principles for deriving variational loss functions tailored for different DR applications. Through this framework, we developed a novel variational method called DVSIB, which compresses variables X and Y into latent variables Z_X and Z_Y respectively, while maximizing the mutual information between Z_X and Z_Y. Notably, DVSIB produces two distinct latent spaces, a feature highly desirable in various applications, while achieving superior data efficiency compared to existing methods in terms of classification accuracy. By illustrating the derivation process of variational bounds for terms common to all examined DR methods, we offer a comprehensive library of typical terms in Appendix 3.7.1, which can serve as a reference for deriving additional DR techniques. Moreover, we (re)-derived several prominent DR methods, including β-VAE, DVIB, DVCCA, etc., showcasing the versatility and applicability of our framework. Through implementation and evaluation, we demonstrated that methods aligning more closely with the structure of dependencies in the underlying data tend to yield more useful latent spaces, as evidenced by dimensionality and reconstruction accuracy metrics. Notably, SDR methods like DVSIB and DVSIB-private emerged as top performers, offering separate latent spaces for X and Y and enabling the capture of additional information beyond shared labels with a sample efficiency better than in competing methods. This quality – being more sample efficient [61, 55, 56, 130] – previously mentioned in the literature with no clear understanding and conflicting arguments [57] is now confirmed and understood better. Furthermore, our framework extends beyond variational approaches, allowing for the implementation of deterministic models, such as autoencoders. By leveraging specialized encoder and decoder networks, our framework can accommodate various constraints and symmetries, thus potentially serving as a versatile toolkit for a wide range of DR methods. With the availability of tools and code [1], I aim to facilitate the adoption of this approach across diverse problem domains.

5.3 On Efficient Estimation of Mutual Information

Chapter 4 investigated the estimation of mutual information within the lens of SDR approaches discussed in Chapters 2 and 3. The MI estimation is challenging, with no universally good estimators for finite data sets, necessitating the use of assumptions. Traditional methods in the literature are limited in their application to modern experimental data, as they cannot effectively handle high dimensionality (they work well up to $\mathcal{O}(10)$ dimensions, whereas many modern experimental setups generate data with $\mathcal{O}(10^3)$ dimensions or more). Neural Networks (NN) based methods were introduced to mitigate such issues, and they offer promising results. However, the literature lacks rigorous testing and evaluation of NN based estimation methods. For instance, ML methods are typically tested on low-dimensional Gaussian data with effectively infinite samples, a scenario that eliminates overfitting but does not reflect real-world research problems with limited samples and high dimensionality. Nevertheless, we demonstrated that, if the data are truly linear, simple estimation based on empirical correlation matrices suffices; otherwise, more complicated methods are necessary. Specifically, in nonlinear cases, appropriate nonlinear methods are required. If the data is linearly embedded in a high-dimensional space, suitable DR methods (like rCCA, as shown in Chapter 2) can embed it into a lower-dimensional space, where MI can be estimated, again, based on empirical correlation matrices. However, selecting the number of latent dimensions is crucial in this case to avoid overfitting. In cases where the data is linear but embedded nonlinearly (e. g., with a frozen neural network), even the best linear DR methods, such as rCCA, are insufficient, as determining the correct dimensionality becomes challenging. Already existing ML methods are suitable for effectively infinite data scenarios without overfitting concerns.

However, for more realistic finite data scenarios, addressing overfitting and when to stop training, is necessary. We viewed existing ML methods as SDR approaches, which simultaneously embed large-dimensional data into smaller dimensional latent spaces, where NN based methods can be used to estimate MI with high accuracy. We developed various consistency checks and heuristics to determine the reliability of the results. This Chapter aimed to demystify such methods and provide a practical guide for their usage in realistic scenarios. It also highlights the utility of SDR approaches in general. As it turns out, successful MI estimators could be viewed as SDR methods as well.

5.4 On Discovering Coordinates of Dynamics

[1]So far, previous Chapters, largely revolved around static datasets, where the relationship between modalities is captured within each (X, Y) pair. We have not yet considered dynamics, which is where many interesting physics problems lie. In fact, dynamical data can be cast in a manner consistent with our multiview setup. For example, consider a dynamical system, where X represents the current observations and Y denotes the future ones [40]. In this context, past and future could be symmetrically defined as a fixed number of past and future snapshots of the system state [35]. Or, similar to a predictive information definition [22, 20], we can define the future as consisting of an infinite number of snapshots (in practice, it suffices to use a large number, larger than the autocorrelation time), and the past would then be described by a long sequence of snapshots of the system state. Here then the shared representation between the past and the future is some generalized coordinates of the system's dynamics.

[1] This section is in part based on an ongoing work with K. Michael Martini and Ilya Nemenman. The development of the idea was a collaborative effort among all three authors. The code was written by K. Michael Martini and Myself. The figures presented in this section are produced by myself. The text was written by myself and jointly revised by all authors.

5.4.1 The Setup

To illustrate this, let us take a simple physical pendulum as shown in Fig. 5.1. Its dynamics are perfectly described by the angle and angular velocity at any given time t. The phase portrait of ω vs θ is shown in Fig. 5.2[2]. The left subplot shows the exact phase space calculated analytically from the differential equations of a physical pendulum with the same properties as the pendulum in Fig. 5.1, with different starting conditions and centered at zero to be between $-\pi$ and π. The middle subplot is for the actual physical pendulum with data obtained from [35]. The right subplot shows the same data but in polar coordinates, which will become relevant later. The phase space of a pendulum is equivalent to an infinite cylinder due to the periodic nature of the angle θ and the unbounded nature of the angular velocity ω. θ wraps around to form the circumference of the cylinder, while ω extends along its length. One can project the cylinder onto a plane using polar coordinates $theta$ and r as follows:

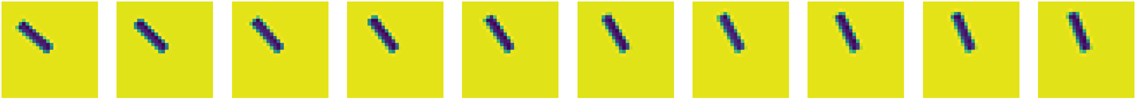


Figure 5.1: Individual frames from a physical single pendulum's motion. The dataset and more information can be found in [35]. Each image represents a frame captured during the pendulum's real motion. The sampling rate is 60 Hz, and each frame is 28×28 pixels, with a total of 60 frames per experiment. Each experiment has different initial conditions, with a total of 1200 experiments. The pendulum is of mass of 1 kg, and a length of 0.5 m.

$$\begin{aligned} \theta &\to \theta \\ \omega + \text{offset (or: offset} - \omega) &\to r, \end{aligned} \tag{5.1}$$

with the offset larger than the maximum angular velocity in the experiments, so that the radius r is positive. The offset arises from compressing the negative range of ω

[2] The direction arrows are suppressed for clarity, in this figure and the subsequent figures as well.

into a small region, creating a hole in the polar plot. Additionally, the system has two fixed points: a stable fixed point at $(\theta, \omega) = (0, 0)$, where trajectories are periodic, and an unstable fixed point at $(\theta, \omega) = (\pi, 0)$ (or $(-\pi, 0)$), where trajectories diverge. A proper embedding from any algorithm should reflect these topological characteristics: the hole due to the cylindrical structure of the phase space, one stable fixed point, and one unstable fixed point.

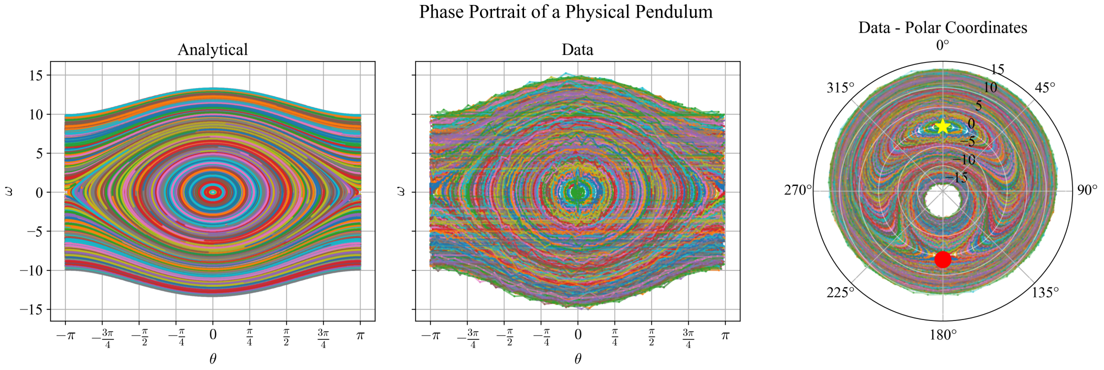


Figure 5.2: Phase space portraits of a physical pendulum for ω vs θ. The left subplot shows the exact phase space calculated analytically from the differential equations of a physical pendulum with the same properties as the pendulum in Fig. 5.1, with different initial conditions, then wrapped between $-\pi$ and π. Each trajectory is for a new experiment with different initial conditions, and the color scheme is arbitrary. The middle subplot is for the actual physical pendulum with data obtained from [35] and shown in Fig. 5.1, where the physical quantities θ and ω are obtained separately from the videos by different analysis, in which the angles and their derivatives are calculated by tracking the pendulum using computer vision tools, as described in [35]. The right subplot shows the same data but in polar coordinates. The yellow star is the stable fixed point, while the red circle is the unstable one.

5.4.2 DVSIB For Dynamics

Using insights from previous chapters, we leverage a DVSIB-like structure to compress the movie data in the hope to discover generalized coordinates of the physical pendulum. In this setup, we choose X to be two consecutive "past" frames at time steps t and $t + 1$, while Y then represents the "future" with frames at time steps $t + 2$ and $t + 3$. Our goal is to embed these data into Z_X and Z_Y, maximizing the mutual information between the latter. We expect the low-dimensional embeddings to represent interpretable generalized coordinates for this system. Since we do not explicitly need the reconstruction of the movie images from the latent variables in this task, we can turn off the reconstruction term [3]. Additionally, we want to impose time translation symmetry in our reduction process, in the sense that the algorithm learns we go from observables (the frames formed into X and Y) to the latent variables (Z_X and Z_Y) in the same way. Thus we employ the same encoder network for both the past and the future frames, as it forces the past encoder to learn the same compression that the future encoder does (i.e, $p(Z_X|X) = p(Z_Y|Y)$). The resultant loss function is:

$$L = I^E(X; Z_X) + I^E(Y; Z_Y) - \beta I^D(Z_X; Z_Y), \tag{5.2}$$

where $I^E(X; Z_X)$ and $I^E(Y; Z_Y)$ are approximated with the same neural network.

5.4.3 Preliminary Results

Figure 5.3 illustrates the preliminary results obtained from the training (additional details about the architecture of the networks and training procedure are in the SI 5.4.5.1). To illustrate that the method learned the underlying physics, we directly input the past and future frames into our model, and we require our model to give us two dimensional Z_X and two dimensional Z_Y (when we are designing the networks, we choose the size of the low dimensional spaces –among other parameters–, 3d embeddings are shown next in Fig. 5.4).Then we plot the resulting embeddings Z_X (or Z_Y), colored by the values of various quantities, which we know to be important in this

[3] Reconstruction adds complexity in terms of compute and data. This is because we are learning distributions in the form of $p(X|Z)$, which has the dimensionality of X. Unless needed for generative tasks, or to stabilize compressed variables and avoid representation collapse, reconstruction should be minimized or turned off. Representation collapse occurs when Z_X and Z_Y become the same variable, or independent of the data. This phenomenon, leading to multiple independent trivial solutions, is discussed in [101]. The washout problem (the degradation of information as it propagates through the layers of a neural network) is also mentioned in [35] as a known issue in various ML algorithms.

problem. Specifically, we know that the two key variables governing this system are the angle θ and the angular velocity ω. The values of θ and ω are obtained separately through another procedure described in the paper, from which we obtained the datasets [35]. The embeddings we obtain are nonlinear manifolds in two dimensions.

Interestingly, our embeddings successfully capture the dynamically relevant variables within this 2D space as shown in Fig. 5.3. When we color the embeddings Z_X (and similarly for Z_Y, due to the symmetric compression, Z_Y is the same as Z_X shifted by two frames) based on θ (left), a clear gradient of angles becomes evident within our manifold along the "angular" direction. Moreover, the points at 0 and 2π in the angle space are connected, indicating a continuous representation. Additionally, when we color the embeddings by ω (middle), we observe a gradient along the "second" dimension of the manifold, revealing that along this second nonlinear embedding, we recover the angular velocity ω. More importantly, when we color the embeddings as individual trajectories (right), we recover a topologically correct phase space of ω vs θ in polar coordinates, which was suggested from the coloring by θ or ω individually that θ is encoded in the angular direction, and ω is encoded in the radial one. The resemblance between the topology of the results in Fig. 5.3 and the polar representation of the true phase space in Fig. 5.2 is evident. We recover the hole due to the parametrization of ω in the radial direction, the two fixed points in the right place, and the nearly circular trajectories near the stable fixed points. The model, indeed, learned that the relevant features between the frames of the past and the future are generalized coordinates that resembles the true θ and ω of the pendulum.

Even if we increase the dimensionality of Z_X and Z_Y as in Figure 5.4, the model continues to learn a deformed 2D manifold (as shown in the last subplot). This is shown in the continuity of the angular direction, the gradient in the radial direction, the fixed points in the phase space, and the hole in it –albeit not as clear as in the 2d situation, probably due to suboptimal training. These observations indicate that

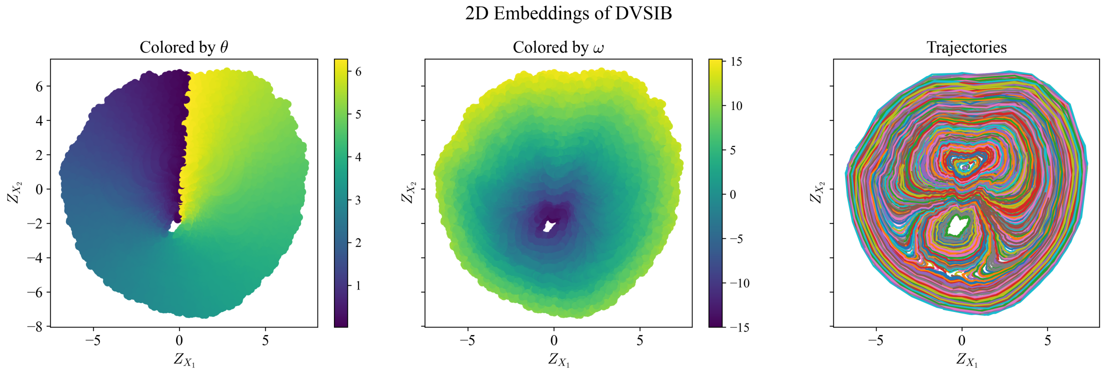


Figure 5.3: Left: Embeddings Z_X in 2D colored by θ. The gradient of angles within the manifold along the "angular" direction is clearly visible, with points at 0 and 2π connected, indicating a continuous representation. Middle: Embeddings Z_X in 2D colored by ω. A gradient along the "second" dimension of the manifold is observed, indicating recovery of the angular velocity ω. Right: The same embeddings, but colored as individual trajectories. We observe that these embeddings represent the phase space of the pendulum in polar coordinates, as shown in Fig. 5.2, subject to an arbitrary rotation in θ and a shift in ω. The embeddings are results of training a simple implementation of 5.2 with frames of experiments of single physical pendulum obtained from [35], the $I(Z_X; Z_Y)$ term is trained as part of DVSIB with SMILE ($\tau = 5$) estimator with a concatenated critic (the networks architecture is described in detail in SI 5.4.5.1).

although the system is observed in high dimensionality, its latent variables are 2d, even when allowed to occupy 3d. While one may have expected this knowing the second-order nature of Newton's laws and the fact that we only used 2 movie frames to define both the past and the future, the dimensionality of the data is $\sim 10^3$, so the fact that the DVSIB for dynamics architecture detects the 2d structure is nontrivial.

It's important to note that our approach is an "out-of-the-box" application of the method, and we do not impose constraints on the embeddings. And while we could end up with a twist or a collapse in our embeddings (as shown in 3d in Fig. 5.4, and while other training parameters could lead to the same behaviour in 2d as well– not shown), the 'twists' in the manifold appear to be points of ambiguity near the unstable fixed points (as we do not see that along the θ direction), where the model is unable to resolve the changes from positive to negative ω. Nevertheless, even with this twist, Figure 5.4 still showcases a clear separation of the relevant dynamical

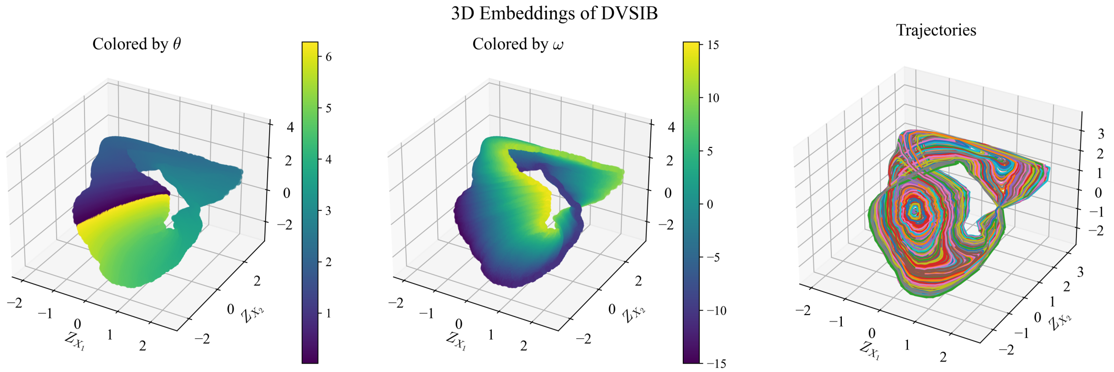


Figure 5.4: Left: Embeddings Z_X in 3D colored by θ. The gradient of angles within the manifold along the "angular" direction is visible, with points at 0 and 2π connected, indicating a continuous representation. Middle: Embeddings Z_X in 3D colored by ω. A gradient along the "second" dimension of the manifold is observed, indicating recovery of the angular velocity ω, albeit with a twist near the unstable fixed point. Right: The same embeddings, but colored as individual trajectories. We observe that these embeddings represent closed trajectories around the stable fixed point, albeit with an ambiguity near the unstable one. Different training parameters (different number of neurons and hidden layers in the critic) could lead to different embeddings, some of which are shown in the SI Fig. 5.4.5.2. The embeddings are results of training a simple implementation of Eq. 5.2 with frames of experiments of single physical pendulum obtained from [35], the $I(Z_X; Z_Y)$ term is trained as part of DVSIB with SMILE ($\tau = 5$) estimator with a concatenated critic (the networks architecture is described in SI 5.4.5.1).

variables that the model has learned directly from the experimental videos, without any preprocessing or further embedding. A possible second step would be learning the dynamics, symbolically, in this low dimensional space, characterized by a handful of dimensions rather than the apparent high dimensionality of the videos. There are multiple methods in the literature one can use for learning such differential equations [27, 42, 31].

5.4.4 Additional Questions and Remarks

While the preliminary results for using DVSIB for dynamics are promising, indicating that we can recover the system's latent variables directly from observations with minimal additional constraints, there are still relevant questions that need further exploration.

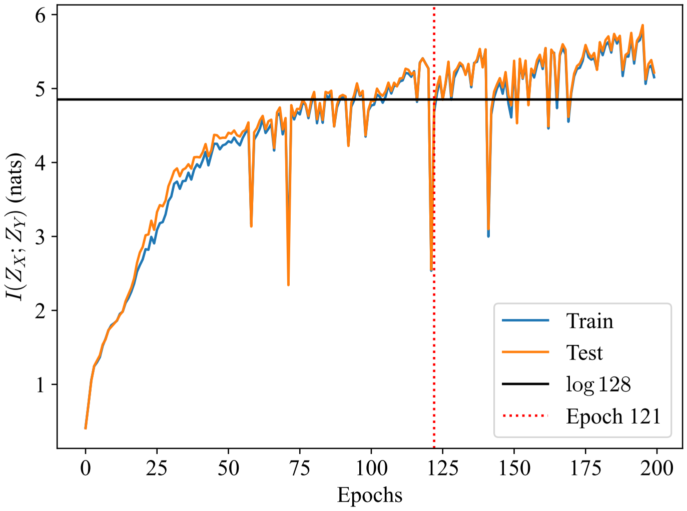


Figure 5.5: Information between embeddings of the past and the future for the pendulum dataset, $I(Z_X; Z_Y)$, during the training. The training curve evaluates a subset of the frames used during the training (100 experiments out of 1000 used for training, each experiment having 60 frames) and another subset for the test that was not seen during the training (100 other experiments with 60 frames each). The training and test curves are almost on top of each other. We notice fluctuations in the training that decrease the amount of information severely. These fluctuations are often observed in the training embeddings, as seen in Fig. 5.6 for epoch 127, where the embedding changes significantly, likely indicating an optimizer-related phenomenon (often called loss spikes[4] [169, 165]) that the training encountered.

When to stop training? As discussed in Chapter 4, the MI estimators (SMILE with $\tau = 5$ in this case) can overfit during training, and we have to make sure that we are reporting the true information, not just an arbitrary number produced by the model during its training. However, when we examine the training curves to obtain the embeddings in Fig. 5.4, for example, we notice that the model does not overfit. This is evident from the curves of $I(Z_X; Z_Y)$ for the train and test datasets in Fig. 5.5 being

[4] Loss spikes are an abrupt increase –or decrease in the case of MI– in the loss function value, typically occurring due to the network encountering regions in the loss landscape with sharp gradients or suboptimal local minima. These spikes often arise from instabilities in the optimization process, where the training dynamics momentarily lead the model into unfavorable regions, causing the loss to increase before stabilizing again as the model continues to learn and adapt.

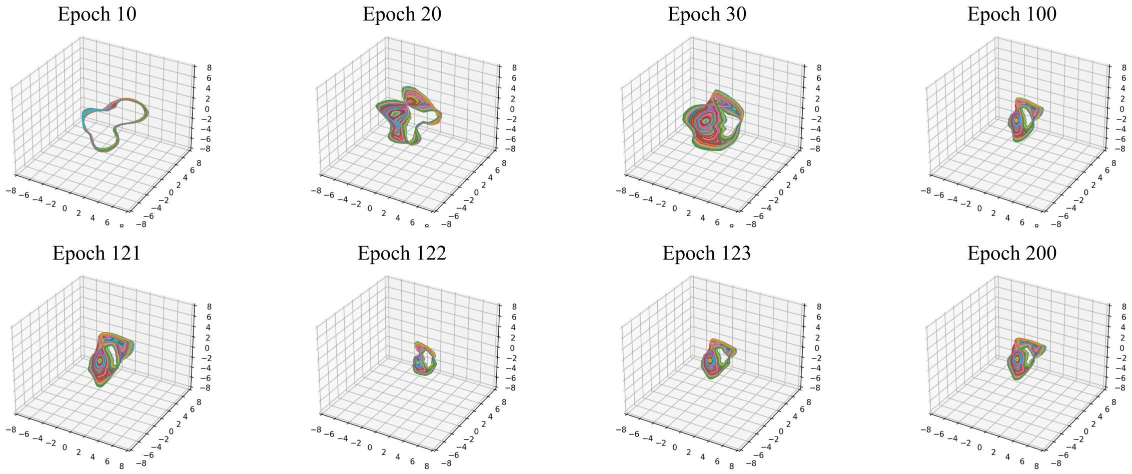


Figure 5.6: Embeddings of 100 experiments evaluated at specific training epochs. We observe the evolution of the embeddings from a straight line to the final structure obtained at the end. Notably, after 100 epochs, the training becomes relatively stable, as also reflected in Fig. 5.5, with fluctuations that can cause the embeddings to change within their space, as shown in epoch 127, for example.

on top of each other[5], which could be attributed to the simplicity of the image we are trying to compress—it is one body on a static background that moves slightly between each frame and the subsequent one. However, we observe fluctuations in the training curves that could be attributed to getting stuck in local minima, as seen in Fig. 5.6, where the embeddings at epoch 127 moved in space. Thus, additional investigation is required to explore these fluctuations, and whether a gentler optimization with a smaller (or adaptive) learning rate, for example, would alleviate this issue. We think that the approach to choose the stopping condition for this problem is by considering some metric of consistency and saturation of the training curves. For example, [131] considers the correlations between different embeddings (for different experiments, for example) as a sign of good training, which could be useful in this problem as well.

[5] At some points during the training, the test curve is slightly higher than the training curve, which is primarily due to the finite data set size used in evaluation, which produces fluctuations. During training, the model uses mini-batches of 128 frames each. However, the evaluation of MI is performed on 100 experiments, each with 60 frames, resulting in 6000 frames for both training and testing to be loaded onto the GPU simultaneously. This creates a significant computational overhead. While it is possible to increase this number to increase the accuracy, it would require considerably larger computational resources than we have at our disposal.

The interpretation of mutual information: MI calculated between the past and future embeddings of the system could be a good proxy for the success of the system in recovering the relevant variables. In this problem, an analytical calculation of the MI between the sets of frames in the past and the future would provide a theoretical bound to check if the model actually learned the underlying dynamics. There are known aspects of this problem from physics or experiments, such as the sampling rate (i. e., the time between frames; if this time interval approaches zero, becoming truly continuous, the mutual information should become infinite), as well as the length and weight of the pendulum, and the governing equations of its motion. However, it is not straightforward to calculate the MI between multiple frames of the past and the future, which will require further work. While analytically calculating the MI might not be possible for scenarios where we do not know the answer ahead of time, it would be beneficial in establishing benchmark problems to assert the utility of DVSIB for inferring dynamics (or identifying its shortcomings). Establishing the utility of MI as a measure of goodness of the training would be valuable not only for the general training of DVSIB for dynamics, but also for addressing additional questions as the following.

How many frames to use for X (and Y)? The previous results (Figs 5.3, 5.4) were obtained for two frames for both the past and the future, X and Y. Since we know that the dynamics of the pendulum are captured within the variable θ and its first derivative, $\dot{\theta} = \omega$, and the derivative is calculated from two time points, we should expect that increasing the number of frames would not increase the MI by much if the finite difference between two time points is a good approximation for the derivative. Thus, we would expect to see a saturation in the measurement of MI with the addition of more frames. We should be able to see such saturation in our data, but we leave this analysis for future work.

How many dimensions for Z_X (and Z_Y)? We have already explored the cases for 2D and 3D low-dimensional embeddings in Figs. 5.3 and 5.4. We observed that 2D is sufficient to replicate the true phase space of the pendulum. In 3D, the embeddings exhibited similar topological features to the 2D case, though with a twist in the embeddings. With more careful training, it might be possible to achieve a clear 2D embedding within the 3D space as well. This is because we have, in theory, only two latent variables θ and ω. Thus, adding more possible dimensions would not add more dynamical variables, and we should expect a saturation in the measurement of MI with the addition of more dimensions. However, the extra dimensions might allow different parameterizations for the latent variables. We leave the exploration of the interplay between the extra embedding dimensions, the ease of training, and detecting more useful parameterization of the latent space to future work.

Finally: By customizing DVSIB to match the structure of the problem and to consider explicitly the time translation symmetry, the method was able to learn the proper coordinates for the underlying dynamics directly from observations (movies) without preprocessing or *ad hoc* assumptions. While this approach is still under development, we believe that its potential for use in many future applications is vast.

5.4.5 Supplementary Information

5.4.5.1 Neural networks architecture used for dynamical inference

The loss function Eq. (5.2) is implemented as follows:

- $I^E(X; Z_X)$ is implemented as a fully connected feed-forward neural network. The network has an input layer of 1568 neurons (each frame is 28x28 pixels, and we have two frames flattened and stacked together), two hidden layers with 1024 neurons each, and two nodes in the output layer that correspond to the means and variances of a gaussian distribution that we learned. Each node is of the size of the dimensionality of Z_X, which is 2 for Fig. 5.3 and 3 for Fig. 5.4.

$I^E(Y; Z_Y)$ is implemented in a similar manner.

- The MI estimator network is a concatenated critic that takes both Z_X and Z_Y embeddings, stacks them together, and then passes them through a fully connected feed-forward neural network with one hidden layer of 32 neurons and an output layer of size 1. The outputs for different inputs in the final layer are calculated based on the SMILE approximation to MI (cf. Chapter 4).
- All layers in all networks are initialized with Xavier Uniform initialization[6] [53].
- The networks are trained with the ADAM optimizer [86] with a learning rate of 5×10^{-5}, and $\beta = 256$.

5.4.5.2 Additional embeddings at different parameters

The embeddings change as the parameters (β, number of hidden layers in the MI critic, number of neurons in these layers, learning rate, etc.) change. The following plots show different embeddings after training for 200 epochs and using the trained encoders to obtain the embeddings of all the training experiments in 3d space. Alternative embeddings evaluated for other parameter values are shown in Fig. 5.7.

5.5 On Neural Activity and Behaviour

Understanding the mutual interplay between neural activity and behavior is the Holy Grail for deciphering brain-body interactions. Recent technological advancements enable simultaneous recordings of activity from tens of thousands of individual neurons across various brain regions with exceptional temporal and spatial precision [140, 139,

[6] Xavier uniform initialization is a method used to initialize the weights of neural networks by drawing them from a uniform distribution. The weights are sampled from a range of $[-\sqrt{6/(n_{\text{in}} + n_{\text{out}})}, \sqrt{6/(n_{\text{in}} + n_{\text{out}})}]$, where n_{in} and n_{out} represent the number of input and output units, respectively. This initialization helps to maintain the variance of activations and gradients across layers, allowing for stable and efficient training.

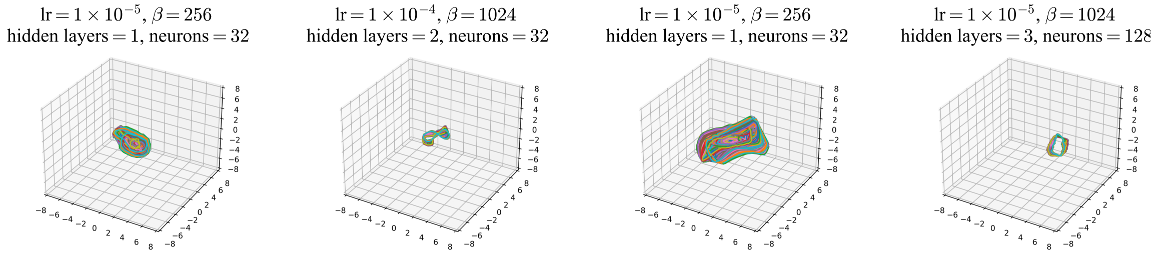


Figure 5.7: Embeddings of training experiments evaluated at the 200th training epoch for different parameters of learning rate for the optimizer, β for the DVSIB loss, number of hidden layers, and number of neurons in those hidden layers. We observe different behaviors, suggesting the need for a careful training. We can see that all of them developed some form of confined manifold. This manifold is regular in some situations and irregular in others.

150]. Simultaneously, there is a growing effort in theoretical research to develop decoders capable of translating neural signals into behavior [137, 114, 70, 108, 92, 113]. These decoders often reveal that a low-dimensional representation of neural data is sufficient for decoding behavior, shedding light on how the brain integrates signals from different regions to execute specific tasks. This alignment with low-dimensional representations suggests the existence of population variables within neural activity that correspond to behavior [119, 52, 159]. Besides facilitating data interpretation, these representations may lay the groundwork for theoretical frameworks describing the underlying dynamics of both neural activity and resulting behavior, which directly impact how we understand different diseases and how to mitigate their effects with prosthetic devices [10, 77], or drugs [29, 74, 9].

However, despite considerable progress, a comprehensive theoretical framework describing the relationship between neural activity and behavior across diverse experimental conditions remains elusive. Different DR methods are at the core of the ongoing research. Traditional DR methods are tailored to specific problem classes, primarily focusing on reducing variability in neural activity without considering its covariation with behavior. We have shown in synthetic and real world examples that such approaches may overlook dimensions significant for describing covariation, rendering them unsuitable for this purpose.

A more effective approach to address this covariation is by considering the neural activity and behavior simultaneously. There are few examples in the neuroscience literature that consider behavior while reducing neural activity. Such methods typically compress only the recorded neural activity conditioned on the behavior or presented stimuli [130, 81, 131]. In the language of Chapter 3, and with X being the recorded neural activity and Y being the resultant observed behavior, we are dealing within an Information Bottleneck framework, compressing X to Z, while Z is maximally informative about Y. Such an approach might be useful if we are considering a low-dimensional behavior Y. This could be, for example, a left-right motion on a treadmill or a mechanical arm that can be moved in only a few directions [131, 108]. However, once the behavior or stimulus is high-dimensional, (for example, recorded videos of motion or a high-dimensional visual stimulus), a reduction of the behavior Y is needed as well. Such a reduction is usually done separately, by means of extracting useful information based on prior knowledge. For example, [102] extracts and tracks joint positions when considering recorded motion as the behavior. Similarly, [30] extracts important features from videos often used as visual stimuli. However, such separate independent reduction makes us question the validity of the approach, as it might overlook relevant features of covariation due to their low variation (and vice versa).

However, there are a few exceptions that compress both the behavior and neural activity simultaneously during the reduction. For example, the method in Ref. [54], which can also be cast in Chapter 3 language as Deep Variational CCA with private information, compresses both. In such a setup, X, the neural activity is reduced to a shared part, Z, and a private part W_X, and both Z and W_X are important to recover the full behavior X. Similarly, Y is reduced to the shared part Z, and a private part W_Y. Then, Z is the relevant mixed space of both that can be used for further study.

However, while this last example is performing simultaneous reduction, it produces three spaces, one for each of the two modalities that is relevant to them uniquely, and a shared one with mixed units that is relevant for both, making it unclear how to measure and interpret the embeddings as neural activity and/or behaviour in such a mixed latent space. DVSIB, on the other hand, is the first method (to our knowledge) that allows for two distinct latent spaces for the two different modalities X and Y, with different properties, yet maximally informative. The new avenues opened by such approaches are interesting and worth studying. For instance, DVSIB and related techniques could be used as supervised dimensionality reduction instead of regular methods like PCA which most practitioners use. Or it can be used to align recorded activity in one part of the brain with other parts of the brain. In this case, a perfect alignment should correspond to maximum mutual information.

5.6 Final Thoughts

In this Dissertation, I aimed to clarify the concept of dimensionality reduction, illustrating that a specific method chosen for it matters a lot more than a mere preprocessing step before data modeling. I explored the overarching principle that, to address a data-driven research question effectively using DR methods, we must align the structure of our methodology with the essence of the inquiry.

DR requires careful method selection and an understanding of expected outcomes. Our results show that, when we want to find the shared information among multiple data sources, simultaneous DR is the preferred approach. We provided extensive discussions, designed novel tools, introduced new heuristics and consistency checks to support this methodology. Personally, I find a great appeal in the idea of obtaining low-dimensional descriptions of complex systems, recognizing that the underlying simplicity often lies within the apparent complexity. I envision this Dissertation as a step towards providing tools and insights for physicists engaged in the timeless pursuit of uncovering simplicity amidst complexity. While many physicists are rightfully fascinated by the maxim that “more is different,” [6] perhaps we can similarly embrace the idea that more can also be simple.

Bibliography

[1] Eslam Abdelaleem and K. Michael Martini. Deep variational multivariate information bottleneck code repository. GitHub, 2023. URL `https://github.com/KMichaelMartini/DVMIB`.

[2] Eslam Abdelaleem, Ilya Nemenman, and K Michael Martini. Deep variational multivariate information bottleneck–a framework for variational losses. *arXiv preprint arXiv:2310.03311*, 2023.

[3] Eslam Abdelaleem, Ahmed Roman, K Michael Martini, and Ilya Nemenman. Simultaneous dimensionality reduction: A data efficient approach for multimodal representations learning. *arXiv preprint arXiv:2310.04458*, 2023.

[4] Wickliffe C Abraham. How long will long-term potentiation last? *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358 (1432):735–744, 2003.

[5] Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. URL `https://arxiv.org/abs/1612.00410`.

[6] Philip W Anderson. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396, 1972.

[7] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical

correlation analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[8] András Antos and Ioannis Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19(3-4): 163–193, 2001.

[9] Lawrence G Appelbaum, Mohammad Ali Shenasa, Louise Stolz, and Zafiris Daskalakis. Synaptic plasticity and mental health: methods, challenges and opportunities. *Neuropsychopharmacology*, 48(1):113–120, 2023.

[10] Panagiotis K Artemiadis and Kostas J Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *IEEE transactions on robotics*, 26(2):393–398, 2010.

[11] Francis R Bach and Michael I Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley*, 2005.

[12] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

[13] Richard T Baillie, G Geoffrey Booth, Yiuman Tse, and Tatyana Zabotina. Price discovery and common factor models. *Journal of financial markets*, 5(3):309–321, 2002.

[14] Feng Bao. Disentangled variational information bottleneck for multiview representation learning. In *Artificial Intelligence: First CAAI International Conference,*

CICAI 2021, Hangzhou, China, June 5–6, 2021, Proceedings, Part II 1, pages 91–102. Springer, 2021.

[15] David Barber and Felix Agakov. Information maximization in noisy channels: A variational approach. *Adv. Neural Inf. Proc. Syst.*, 16, 2003.

[16] David Barber and Felix Agakov. The IM algorithm: A variational approach to information maximization. *Adv. Neural Inf. Proc. Syst.*, 16(320):201, 2004.

[17] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[18] Adrian Benton, Huda Khayrallah, Biman Gujral, Dee Ann Reisinger, Sheng Zhang, and Raman Arora. Deep generalized canonical correlation analysis. *arXiv preprint arXiv:1702.02519*, 2017.

[19] Gordon J Berman, Daniel M Choi, William Bialek, and Joshua W Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014.

[20] William Bialek and Naftali Tishby. Predictive information. *arXiv preprint cond-mat/9902341*, 1999.

[21] William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural computation*, 13(11):2409–2463, 2001.

[22] William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural computation*, 13(11):2409–2463, 2001.

[23] Magnus Borga, Tomas Landelius, and Hans Knutsson. *A unified approach to pca,*

pls, mlr and cca. Linköping University, Department of Electrical Engineering, 1997.

[24] J-P Bouchaud, Laurent Laloux, M Augusta Miceli, and Marc Potters. Large dimension forecasting models and random singular value spectra. *The European Physical Journal B*, 55:201–207, 2007.

[25] Jean Philippe Bouchaud, Laurent Laloux, M Augusta Miceli, and Marc Potters. Large dimension forecasting models and random singular value spectra. *The European Physical Journal B*, 55:201–207, 2007.

[26] Serena Bradde and William Bialek. Pca meets rg. *Journal of statistical physics*, 167:462–475, 2017.

[27] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[28] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. Cleaning large correlation matrices: Tools from random matrix theory. *Physics Reports*, 666:1–109, 2017. ISSN 0370-1573. doi: https://doi.org/10.1016/j.physrep.2016.10.005. Cleaning large correlation matrices: tools from random matrix theory.

[29] Qingjiu Cao, Yufeng Zang, Li Sun, Manqiu Sui, Xiangyu Long, Qihong Zou, and Yufeng Wang. Abnormal neural activity in children with attention deficit hyperactivity disorder: a resting-state functional magnetic resonance imaging study. *Neuroreport*, 17(10):1033–1036, 2006.

[30] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[31] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[32] Sarath Chandar, Mitesh M Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural Computation*, 28(2):257–285, 2016. doi: 10.1162/NECO_a_00801.

[33] James Chapman and Hao-Ting Wang. Cca-zoo: A collection of regularized, deep learning based, kernel, and probabilistic cca methods in a scikit-learn style framework. *Journal of Open Source Software*, 6(68):3823, 2021.

[34] Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for gaussian variables. *Advances in Neural Information Processing Systems*, 16, 2003.

[35] Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science*, 2(7):433–442, 2022.

[36] Wynne Chin and P. Newsted. Structural equation modeling analysis with small samples using partial least square. *Statistical Strategies for Small Sample Research*, 01 1999.

[37] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013.

[38] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al.

The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013.

[39] Lucy J Colwell, Yu Qin, Miriam Huntley, Alexander Manta, and Michael P Brenner. Feynman-hellmann theorem and signal identification from sample covariance matrices. *Physical Review X*, 4(3):031032, 2014.

[40] Felix Creutzig, Amir Globerson, and Naftali Tishby. Past-future information bottleneck in dynamical systems. *Physical Review E*, 79(4):041925, 2009.

[41] Paweł Czyż, Frederic Grabowski, Julia Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators. *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Bryan C Daniels and Ilya Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature communications*, 6(1):8133, 2015.

[43] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

[44] Adrienne Fairhall, Eric Shea-Brown, and Andrea Barreiro. Information theoretic approaches to understanding circuit function. *Current opinion in neurobiology*, 22(4):653–659, 2012.

[45] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. In *8th International Conference on Learning Representations*. OpenReview. net, 2020.

[46] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. *arXiv preprint arXiv:2002.07017*, 2020.

[47] Philipp Fleig and Ilya Nemenman. Statistical properties of large data sets with linear latent features. *Physical Review E*, 106(1):014102, 2022.

[48] Mario Forni and Luca Gambetti. The dynamic effects of monetary policy: A structural factor model approach. *Journal of Monetary Economics*, 57(2): 203–216, 2010.

[49] Simon Freyaldenhoven. Factor models with local factors—determining the number of relevant factors. *Journal of Econometrics*, 229(1):80–102, 2022.

[50] Nir Friedman, Ori Mosenzon, Noam Slonim, and Naftali Tishby. Multivariate information bottleneck. *arXiv preprint arXiv:1301.2270*, 2013.

[51] Francesco Fumarola, Bettina Hein, and Kenneth D Miller. Mechanisms for spontaneous symmetry breaking in developing visual cortex. *Physical Review X*, 12(3):031024, 2022.

[52] Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

[53] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[54] Rabia Gondur, Usama Bin Sikandar, Evan Schaffer, Mikio Christian Aoi, and Stephen L Keeley. Multi-modal gaussian process variational autoencoders for neural and behavioral data. In *The Twelfth International Conference on Learning Representations*, 2024.

[55] Dale L Goodhue, William Lewis, and Ron Thompson. Pls, small sample size, and statistical power in mis research. In *Proceedings of the 39th Annual Hawaii*

International Conference on System Sciences (HICSS'06), volume 8, pages 202b–202b, 2006. doi: 10.1109/HICSS.2006.381.

[56] Dale L Goodhue, William Lewis, and Ron Thompson. Does pls have advantages for small sample size or non-normal data? *MIS Quarterly*, 36(3):981–1001, 2012. ISSN 02767783.

[57] Dale L Goodhue, Ron Thompson, and William Lewis. Why you shouldn't use pls: Four reasons to be uneasy about using pls in analyzing path models. In *2013 46th Hawaii International Conference on System Sciences*, pages 4739–4748. IEEE, 2013.

[58] Lukas Gosch. Shortcomings and new perspectives on mutual information for representation learning. *Scientific Internship Report 10/2019-04/2020*, 2020.

[59] Marc-André Gosselin and Greg Tkacz. Evaluating factor models: An application to forecasting inflation in canada. Technical report, Bank of Canada, 2001.

[60] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[61] Joe F Hair, Christian M Ringle, and Marko Sarstedt. Pls-sem: Indeed a silver bullet. *Journal of Marketing Theory and Practice*, 19(2):139–152, 2011. doi: 10.2753/MTP1069-6679190202.

[62] Joe Hair Jr, Joseph F Hair Jr, G Tomas M Hult, Christian M Ringle, and Marko Sarstedt. *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage publications, 2021.

[63] Riitta Hari and Lauri Parkkonen. The brain timewise: how timing shapes and supports brain function. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668):20140170, 2015.

[64] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[65] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[66] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

[67] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[68] Geoffrey E Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

[69] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[70] Andrew Holbrook, Alexander Vandenberg-Rodes, Norbert Fortin, and Babak Shahbaba. A bayesian supervised dual-dimensionality reduction model for simultaneous decoding of lfp and spike train signals. *Stat*, 6(1):53–67, 2017.

[71] Caroline M Holmes and Ilya Nemenman. Estimation of mutual information for real-valued data with error bars and controlled bias. *Physical Review E*, 100(2): 022404, 2019.

[72] Caroline M Holmes and Ilya Nemenman. Estimation of mutual information for real-valued data with error bars and controlled bias. *Phys. Rev. E*, 100(2): 022404, 2019.

[73] Timothy Holy and Ilya Nemenman. On impossibility of learning in a reparameterization covariant way. Technical report, Tech. Rep. NSF-KITP-03-123, KITP, UCSB, 2002.

[74] W Michael Hooten. Chronic pain and mental health disorders: shared neural mechanisms, epidemiology, and treatment. In *Mayo Clinic Proceedings*, volume 91, pages 955–970. Elsevier, 2016.

[75] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[76] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 1936. doi: 10.1007/978-1-4612-4380-9_14.

[77] Kejia Hu, Mohsen Jamali, Ziev B Moses, Carlos A Ortega, Gabriel N Friedman, Wendong Xu, and Ziv M Williams. Decoding unconstrained arm movements in primates using high-density electrocorticography signals for brain-machine interface use. *Scientific reports*, 8(1):10583, 2018.

[78] Shizhe Hu, Zenglin Shi, and Yangdong Ye. Dmib: Dual-correlated multivariate information bottleneck for multiview clustering. *IEEE Transactions on Cybernetics*, 52(6):4260–4274, 2020.

[79] Teng-Hui Huang, Aly El Gamal, and Hesham El Gamal. On the multi-view information bottleneck representation. In *2022 IEEE Information Theory Workshop (ITW)*, pages 37–42. IEEE, 2022.

[80] Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O'Donovan. The goa database: gene ontology annotation updates for 2015. *Nucleic acids research*, 43(D1):D1057–D1063, 2015.

[81] Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34:29379–29392, 2021.

[82] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Multi-view representation learning via total correlation objective. *Advances in Neural Information Processing Systems*, 34:12194–12207, 2021.

[83] Xiong-Fei Jiang, Bo Zheng, Fei Ren, and Tian Qiu. Localized motion in random matrix decomposition of complex financial systems. *Physica A: Statistical Mechanics and its Applications*, 471:154–161, 2017.

[84] Mahdi Karami and Dale Schuurmans. Deep probabilistic canonical correlation analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 8055–8063, May 2021. doi: 10.1609/aaai.v35i9.16982. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16982`.

[85] Dustin Keys, Shukur Kholikov, and Alexei A Pevtsov. Application of mutual information methods in time–distance helioseismology. *Solar Physics*, 290: 659–671, 2015.

[86] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[88] Christoph Kirst and Marc Timme. How precise is the timing of action potentials? *Frontiers in Neuroscience*, 3:1029, 2009.

[89] Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian exponential family projections for coupled data sources. *arXiv preprint arXiv:1203.3489*, 2012.

[90] Adam G Kline and Stephanie E Palmer. Multi-relevance: Coexisting but distinct notions of scale in large systems. *arXiv preprint arXiv:2305.11009*, 2023.

[91] Ned Kock and Pierre Hadaya. Minimum sample size estimation in pls-sem: The inverse square root and gamma-exponential methods. *Information systems journal*, 28(1):227–261, 2018.

[92] John W Krakauer, Asif A Ghazanfar, Alex Gomez-Marin, Malcolm A MacIver, and David Poeppel. Neuroscience needs behavior: correcting a reductionist bias. *Neuron*, 93(3):480–490, 2017.

[93] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69(6):066138, 2004.

[94] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

[95] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015.

[96] Changhee Lee and Mihaela Van der Schaar. A variational information bottleneck

approach to multi-omics data integration. In *International Conference on Artificial Intelligence and Statistics*, pages 1513–1521. PMLR, 2021.

[97] Marco Lorenzi, Andre Altmann, Boris Gutman, Selina Wray, Charles Arber, Derrek P Hibar, Neda Jahanshad, Jonathan M Schott, Daniel C Alexander, Paul M Thompson, et al. Susceptibility of brain atrophy to trib3 in alzheimer's disease, evidence from functional prioritization in imaging genetics. *Proceedings of the National Academy of Sciences*, 115(12):3162–3167, 2018.

[98] Marco Lorenzi, Andre Altmann, Boris Gutman, Selina Wray, Charles Arber, Derrek P Hibar, Neda Jahanshad, Jonathan M Schott, Daniel C Alexander, Paul M Thompson, et al. Susceptibility of brain atrophy to trib3 in alzheimer's disease, evidence from functional prioritization in imaging genetics. *Proceedings of the National Academy of Sciences*, 115(12):3162–3167, 2018.

[99] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE transactions on Knowledge and Data Engineering*, 27(11):3111–3124, 2015.

[100] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114 (4):507–536, 1967.

[101] K Michael Martini and Ilya Nemenman. Data efficiency, dimensionality reduction, and the generalized symmetric information bottleneck. *arXiv preprint arXiv:2309.05649*, 2023.

[102] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[103] Pankaj Mehta and David J Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

[104] Leenoy Meshulam, Jeffrey L Gauthier, Carlos D Brody, David W Tank, and William Bialek. Coarse graining, fixed points, and scaling in a large population of neurons. *Physical review letters*, 123(17):178103, 2019.

[105] George Miller. Note on the bias of information estimates. *Information theory in psychology: Problems and methods*, 1955.

[106] Rotem Monsa, Michael Peer, and Shahar Arzy. Processing of different temporal scales in the human brain. *Journal of Cognitive Neuroscience*, 32(11):2087–2102, 2020.

[107] Kevin P Murphy. *Probabilistic machine learning: an introduction.* MIT press, 2022.

[108] Nikhilesh Natraj, Daniel B Silversmith, Edward F Chang, and Karunesh Ganguly. Compartmentalized dynamics within a common multi-area mesoscale manifold represent a repertoire of human hand movements. *Neuron*, 110(1):154–174.e12, 2022. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2021.10.002.

[109] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Information Theory*, 56(11):5847–5861, 2010.

[110] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Adv. Neural Inf. Proc. Syst.*, 29, 2016.

[111] Eakasit Pacharawongsakda and Thanaruk Theeramunkong. A comparative study on single and dual space reduction in multi-label classification. In Andrzej M.J.

Skulimowski and Janusz Kacprzyk, editors, *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, pages 389–400, Cham, 2016. Springer International Publishing. ISBN 978-3-319-19090-7.

[112] Stephanie E Palmer, Olivier Marre, Michael J Berry, and William Bialek. Predictive information in a sensory population. *Proc. Natl. Acad. Sci.*, 112(22): 6908–6913, 2015.

[113] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[114] Rich Pang, Benjamin J Lansdell, and Adrienne L Fairhall. Dimensionality reduction in neuroscience. *Current Biology*, 26(14):R656–R660, 2016.

[115] Liam Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, 2003.

[116] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[117] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[118] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron

Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[119] Matthew G Perich, Sara Conti, Marion Badi, Andrew Bogaard, Beatrice Barra, Sophie Wurth, Jocelyne Bloch, Gregoire Courtine, Silvestro Micera, Marco Capogrosso, et al. Motor cortical dynamics are shaped by multiple distinct subspaces during naturalistic behavior. *BioRxiv*, pages 2020–07, 2020.

[120] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3): 287–296, 1985.

[121] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/poole19a.html`.

[122] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Int. Conf. Machine Learning*, pages 5171–5180. PMLR, 2019.

[123] Marc Potters and Jean-Philippe Bouchaud. *A First Course in Random Matrix Theory: For Physicists, Engineers and Data Scientists.* Cambridge University Press, 2020.

[124] Lin Qiu, Vernon M Chinchilli, and Lin Lin. Variational interpretable deep canonical correlation analysis. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

[125] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[126] Jason W Rocks and Pankaj Mehta. Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models. *Physical review research*, 4(1):013201, 2022.

[127] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[128] Jeremy B Rudd. Underlying inflation: Its measurement and significance. *FEDS Notes. Washington: Board of Governors of the Federal Reserve System, September 18, 2020*, 2020.

[129] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*, 2018.

[130] Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021.

[131] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960): 360–368, 2023.

[132] Jangir Selimkhanov, Brooks Taylor, Jason Yao, Anna Pilko, John Albeck, Alexander Hoffmann, Lev Tsimring, and Roy Wollman. Accurate information transmission through dynamic biochemical signaling networks. *Science*, 346 (6215):1370–1373, 2014.

[133] Claude Elwood Shannon. A mathematical theory of communication. *Bell Syst. Techn. J.*, 27(3):379–423, 1948.

[134] Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in neural information processing systems*, 32, 2019.

[135] Dong Song, Brian S. Robinson, Rosa H.M. Chan, and Theodore W. Berger. Chapter 7 - identification of neural plasticity from spikes. In Denise Manahan-Vaughan, editor, *Handbook of in Vivo Neural Plasticity Techniques*, volume 28 of *Handbook of Behavioral Neuroscience*, pages 135–151. Elsevier, 2018. doi: https://doi.org/10.1016/B978-0-12-812028-6.00007-0. URL `https://www.sciencedirect.com/science/article/pii/B9780128120286000070`.

[136] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

[137] Simon Sponberg, Thomas L Daniel, and Adrienne L Fairhall. Dual dimensionality reduction reveals independent encoding of motor features in a muscle synergy for insect flight control. *PLOS Computational Biology*, 11(4):1–23, 04 2015. doi: 10.1371/journal.pcbi.1004168.

[138] Kyle H Srivastava, Caroline M Holmes, Michiel Vellema, Andrea R Pack, Coen PH Elemans, Ilya Nemenman, and Samuel J Sober. Motor control by precisely timed spike patterns. *Proc. Natl. Acad. Sci. (USA)*, 114(5):1171–1176, 2017.

[139] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.

[140] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D. Harris. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 364(6437):eaav7893, 2019. doi: 10.1126/science.aav7893.

[141] Steven P Strong, Roland Koberle, Rob R De Ruyter Van Steveninck, and William Bialek. Entropy and information in neural spike trains. *Phys. Rev. Lett.*, 80(1):197, 1998.

[142] Milan Studenỳ and Jirina Vejnarová. The multiinformation function as a tool for measuring stochastic dependence. *Learning in graphical models*, pages 261–297, 1998.

[143] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

[144] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018.

[145] Claire Tang, Diala Chehayeb, Kyle Srivastava, Ilya Nemenman, and Samuel J Sober. Millisecond-scale motor encoding in a cortical vocal area. *PLoS Biol.*, 12 (12):e1002018, 2014.

[146] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[147] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[148] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[149] Shinsuke Uda. Application of information theory in systems biology. *Biophysical reviews*, 12(2):377–384, 2020.

[150] Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature neuroscience*, 25(1):11–19, 2022.

[151] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[152] Hrishikesh D Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 1976. doi: 10.1016/0304-4076(76)90010-5.

[153] Joshua T Vogelstein, Eric W Bridgeford, Minh Tang, Da Zheng, Christopher Douville, Randal Burns, and Mauro Maggioni. Supervised dimensionality reduction for big data. *Nature communications*, 12(1):2872, 2021.

[154] Zhibin Wan, Changqing Zhang, Pengfei Zhu, and Qinghua Hu. Multi-view information-bottleneck representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10085–10092, 2021.

[155] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[156] Qi Wang, Claire Boudreau, Qixing Luo, Pang-Ning Tan, and Jiayu Zhou. Deep multi-view information bottleneck. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 37–45. SIAM, 2019.

[157] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015.

[158] Weiran Wang, Xinchen Yan2 Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016.

[159] Ryan C Williamson, Brent Doiron, Matthew A Smith, and M Yu Byron. Bridging large-scale neuronal recordings and large-scale network models using dimensionality reduction. *Current opinion in neurobiology*, 55:40–47, 2019.

[160] Kenneth G Wilson. The renormalization group and critical phenomena. *Reviews of Modern Physics*, 55(3):583, 1983.

[161] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2): 109–130, 2001. ISSN 0169-7439. doi: https://doi.org/10.1016/S0169-7439(01)00155-1. PLS Methods.

[162] Hok Shing Wong, Li Wang, Raymond Chan, and Tieyong Zeng. Deep tensor cca for multi-view learning. *IEEE Transactions on Big Data*, 8(6):1664–1677, 2021.

[163] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[164] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[165] Zhongwang Zhang and Zhi-Qin John Xu. Loss spike in training neural networks. *arXiv preprint arXiv:2305.12133*, 2023.

[166] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

[167] Zhihao Zheng, J Scott Lauritzen, Eric Perlman, Camenzind G Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B Fisher, Nadiya Sharifi, et al. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743, 2018.

[168] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[169] Libin Zhu, Chaoyue Liu, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Catapults in sgd: spikes in the training loss and their impact on generalization through feature learning. *arXiv preprint arXiv:2306.04815*, 2023.

[170] Robert S Zucker and Wade G Regehr. Short-term synaptic plasticity. *Annual review of physiology*, 64(1):355–405, 2002.

[171] Finn Årup Nielsen, Lars Kai Hansen, and Stephen C Strother. Canonical ridge analysis with ridge parameter optimization. *NeuroImage*, 1998. doi: 10.1016/s1053-8119(18)31591-x.