UNIVERSITY OF
CAMBRIDGE

Probabilistic perspectives in scientific representation learning

Aditya Ravuri

Wolfson College

This thesis is submitted on 18th February 2026 for the degree of Doctor of Philosophy

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This thesis does not exceed the prescribed limit of 60 000 words.

Aditya Ravuri

18th February 2026

Abstract

Probabilistic perspectives in scientific representation learning

Aditya Ravuri

In this thesis, we introduce a coherent probabilistic framework for dimensionality reduction, and show how it can be used to understand aspects of wider representation learning. We motivate the search for a probabilistic paradigm by showing that in science, probabilistic models can be used to identify cancer-cells, enable conservation, accelerate climate science, and improve drug discovery, often in ways outside their intended use. In scientific pipelines, various algorithms are used to generate representations of data for downstream analysis. How they relate to each other, however, in general, is not well understood. A unified understanding of the methods has the potential to unlock new methods for science.

We introduce one such unification, with the aim of contributing to statistical model understanding, by showing how a single probabilistic framework "ProbDR" underpins many classical dimensionality reduction methods. ProbDR presents the algorithms as inference algorithms of probabilistic models, highlighting what modelling and semantic assumptions these methods make, and how they are constrained. We present model transformations within ProbDR, showing that its different forms of model specification are equivalent. We then show that insights from model equivalences approximately yield new analytical inference algorithms. ProbDR also highlights the form of constraints necessary for latent variable models—the explicit usage of case-dependent similarity estimators. In the last part of the thesis, we show that ideas of the framework extend to other areas of representation learning, explaining aspects of self-supervised learning models and transformers, leading to improvements in their performance, validated empirically on language and vision tasks. This exemplifies that the insights of our probabilistic framework can be applied to areas not of primary focus. With these ideas, we take a step towards describing characteristics of a paradigm for representation learning.

Acknowledgements

Removed for review.

Contents

Notation & Glossary

n	number of data points.
d	number of data dimensions.
d_q	number of latent dimensions.
$\mathbf{X}$	**explanatory** variable set, often **latent**, in $\mathbb{R}^{n \times d_q}$.
$\mathbf{Y}$	**response** data set, typically in $\mathbb{R}^{n \times d}$, with a row $\mathbf{Y}_i$ corresponding to a data point.
$X \sim \mathcal{D}$	a random variable X is sampled from / follows a distribution $\mathcal{D}$.
$\mathcal{L}(\theta)$	a loss function, minimised w.r.t. θ. Typically inversely related to a log-probability (or a lower bound on it) $\mathcal{L} = -\mathcal{E}$.
$\mathcal{E}(\theta)$	a log-likelihood, evidence lower bound or similar function, maximised w.r.t. θ.
$\log \mathcal{D}(x'\|\theta, ...)$	the log probability (density) corresponding to a distribution $\mathcal{D}$ with parameters $\theta, ...$ evaluated at the value x'.
$\hat{\mathbf{S}}$	an estimate of an empirical data-data covariance matrix, for example, calculated as $\mathbf{Y}\mathbf{Y}^\top/d$ with centered data $\mathbf{Y}$.
$\hat{\mathbf{\Gamma}}$	an estimate of an empirical precision matrix, i.e. the inverse of a covariance matrix.
$\mathbf{A}$	a symmetric adjacency matrix.
$\mathbf{L}$	a graph Laplacian matrix.
GL	a graph Laplacian matrix.

$\mathbf{M} \sim \mathcal{MN}(\boldsymbol{\mu}, \Sigma_r, \Sigma_c)$	a matrix normal distribution with parameters $\boldsymbol{\mu}$, Σ_r, Σ_c. The following equivalence holds, $\text{vec}(\mathbf{M}) \sim \mathcal{N}(\text{vec}(\boldsymbol{\mu}), \Sigma_c \otimes \Sigma_r)$. In almost all cases in the thesis, we use the notation for summarising the following; if $\mathbf{M}_{:j}$ is the j-th column of $\mathbf{M}$ and has the following distribution independent to other random variables, $\forall j : \mathbf{M}_{:j} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, then $\mathbf{M} \sim \mathcal{MN}(\mathbf{0}, \Sigma, \mathbf{I})$.
$\mathcal{S}^{d_q - 1}$	unit hypersphere in d_q dimensions.
vMF	von Mises-Fisher distribution.
$\mathcal{W}$	a Wishart distribution with a parametrisation such that, if $\mathbf{S} \sim \mathcal{W}(\mathbf{M}, \nu)$, then, $\mathbb{E}(\mathbf{S}) = \nu\mathbf{M}$.
$\mathcal{W}^{-1}$	an inverse-Wishart distribution with parametrisation such that, if $\mathbf{S}^{-1} \sim \mathcal{W}^{-1}(\mathbf{M}^{-1}, \nu)$, then, $\mathbf{S} \sim \mathcal{W}(\mathbf{M}, \nu)$.
centrality parameter	assuming an (inverse-)Wishart distribution $\mathbf{M} \sim \mathcal{W}^{\{-1\}}(\mathbf{C}, \nu)$, we denote $\mathbf{C}$ as the "centrality" parameter (instead of the scale parameter as is done traditionally), as $\mathbb{E}(\mathbf{M}) \propto \mathbf{C}$.
$\mathbf{I}$	identity matrix.
$\mathbf{H}$	centering matrix $\mathbf{H} = \mathbf{I} - \mathbf{1}\mathbf{1}^T/n$.
$\mathbf{M}^+$	the Moore-Penrose pseudo-inverse of the matrix $\mathbf{M}$.
$\mathcal{I}$	indicator function.
p	a density function. We denote $p_{\mathcal{M}}$ to denote the density of a model $\mathcal{M}$ specified in the text, or a corresponding distribution, for example, $p_{\mathcal{N}}$ corresponding to the density of a normal distribution.
$\mathbb{P}$	a probability measure. We sometimes denote the probability $\mathbb{P}(X \in A)$ as $\mathbb{P}_X(A)$ or $\mathbb{P}(X)$ when the set A is unimportant for exposition.

$\sigma(.)$	a sigmoid (inverse-logistic) or softmax function.
$\odot$	element-wise multiplication.
$\otimes$	Kronecker product.
$\mathfrak{R}, \mathfrak{I}$	real and imaginary parts.
$\overset{+}{=}$	$a \overset{+}{=} b \implies a = b + k$ for some uninteresting constant k.
$\mathbf{U}, \mathbf{\Lambda}$	typically used for Eigenvectors and Eigenvalues of a matrix.
$\mathrm{KL}(q\|\|p)$	Kullback—Leibler divergence between densities q and p.
$k(\cdot, \cdot)$	typically a kernel function. We denote the matrix shorthand as, $\Sigma = K(\mathbf{X}, \mathbf{X})$ meaning that $\Sigma_{ij}(\mathbf{X}) = k(\mathbf{X}_i, \mathbf{X}_j)$.
PSD	positive semi-definite.
expm	matrix exponential. exp refers to an element-wise exponential.
stop-grad	an operation stop-grad$[f(x)]$ sets the gradients of the argument $f(x)$, to zero.
semantic	applies to model assumptions or characteristics, referring to the "in words" meaning of the models (as opposed to its formal mathematical description). For example, the "in words" meaning of a log-normal distributed variable could be that a variable is positive with its log being symmetrically distributed. By semantic consistency, we mean that two models correspond to equivalent "in words" readings.
probabilistic grammar	probabilistic models forming grammars refers to the fact that many models, and probabilistic programming language statements translate to an "in words" description of a data characteristic being modelled.

kNN	k-nearest neighbour (graph).
major eigencomponents	the d_q major/principal eigenvectors and eigenvalues of a matrix $\mathbf{S}$ correspond to the largest d_q eigenvalues.
minor eigencomponents	the d_q minor eigenvectors and eigenvalues of a matrix $\mathbf{S}$ correspond to the smallest d_q eigenvalues.
$d^2(.,.)$	the Euclidean distance metric, $d^2(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$. We sometimes overload the notation by writing $d^2_{ij}(\mathbf{X})$ to mean $d^2_{ij}(\mathbf{X}_i,\mathbf{X}_j)$.

CHAPTER 1

Introduction

Representation learning is a widely used set of ideas within machine learning and artificial intelligence (Bengio et al., 2013), and has close ties to statistical modelling. It is a core step in many scientific workflows, as scientists convert raw data into features more actionable for computers and/or humans. Scientific representation learning presents practical constraints, as prior knowledge is typically available at the outset without a universally agreed-upon strategy to incorporate it. There is a need for interpretable output, which isn't guaranteed by many methods. In this thesis, in chapter 2, we show that case studies exist across drug discovery, climate science, and genetics. A variety of algorithms are used for representation learning, particularly within the biological sciences (such as t-SNE; Luecken and Theis (2019)), with mechanisms of operation that are hard to modify systematically and lack explicit modelling semantics. Although there is a widespread effort to understand methods in the field, there is no widely adopted, unified set of explicit rules that governs how methods are to be constructed, constrained, and compared with one another. Such a rule-set would streamline problem solving and address the construction and comparability problems faced in scientific representation learning.

To address this problem, we describe the central tenet of a probabilistic paradigm, using the language of probability and statistics, that explains many methods in representation learning. The reason for our study of representation learning from a probabilistic perspective are as follows. Statistical models, when well understood, act as rough high-level grammars (in the sense of a modelling language) with which one can talk about and represent data and associated knowledge of interest. We use the term probabilistic "grammars" in this thesis frequently

as a metaphor for compositional rule-sets for model construction. In other words, we use "grammars" in the sense that we can define a formal language to compile logical real-world representations to probabilistic models (using probabilistic programming languages, reviewed by van de Meent et al. (2021)).

A probabilistic representation learning paradigm would make constructing scientifically-tailored methods accessible to practitioners through the high-level grammars that the models would correspond to. Moreover, probabilistic modelling fits into representation learning contexts, as we can model noise and uncertainties within data explicitly at variable scales. Probabilistic models also bring the following practical strengths. Firstly, they enable the composability of methods and model extension in light of atypical data or prior information. Secondly, they can aid identifiability because priors and models can significantly constrain the search space of algorithms. Thirdly, probabilistic modelling often narrows the objective of interest to the posterior distribution's log-density. This makes automatic inference possible, particularly through the use of probabilistic programming languages (Ghahramani, 2015; Gelman et al., 2013). Fourthly, probabilistic models are typically robust to changes in the problem's representation and are therefore more suitable for representing models of the world. Lastly, probabilistic models enable analyses outside their original use, for example, classifiers storing information about a point's density, discussed in example 2.1.

Therefore, in this thesis we attempt to unify the various algorithms in scientific representation learning, particularly dimensionality reduction (DR) methods, from a probabilistic perspective, and show that such a paradigm has strengths commonly seen with probabilistic frameworks. Moreover, we use our framework in chapter 4 to suggest principled architecture modifications to transformers, suggesting that our ideas extend to unintended impactful use-cases. Concretely, in this thesis, we pose the question:

> **What ideas and models unify representation learning from a probabilistic perspective**?

The answer will be:

> The studied representation learning methods **perform inference in a unified probabilistic model, acting on a minimal aspect of the data**, potentially framed variationally with a fixed variational target representing observations.

An overview of the thesis is as follows. In the **background (chapter 2)**, we present an overview of many ideas revealing the nature of probabilistic models in representation learning, and show that practitioners typically compose models into end-to-end pipelines using high-level semantics. This presentation is in line with other reviews of representation learning, for example, that of Bengio et al. (2013); Murphy (2023). We exemplify in this chapter **how** prior scientific knowledge enters modelling practice—classical Bayesian priors are not the only (or even the most common) form of knowledge input. In fact, it is the explicit estimation and holding static of domain knowledge guided quantities, that constrains models across many use-cases. In many real-world cases, we find that methods already conform to the idea that one must limit degrees of freedom of models for the inference of useful latent variables. In this chapter, we show however, that probabilistic models are not the only tool for representation learning, and that there are many **algorithms** that process data and produce outputs of downstream interest, which currently lack probabilistic interpretations. The key question posed by this chapter is, given our organisation of representation learning methods, **where do such algorithms fit?** As an example, for biological applications such as single-cell data analysis, practitioners use many dimensionality reduction algorithms without probabilistic interpretations to compress high-dimensional data points into lower-dimensional real-valued representations. The representations are for downstream problems such as cluster identification, real-valued phenotypes or causal factors, and temporal ordering within the data (Luecken and Theis, 2019). Outputs of such algorithms, and the algorithms themselves, can be highly uninterpretable and difficult to modify given new information or atypical data. In the thesis, we provide probabilistic interpretations and show concretely that these algorithms, which are widely used, are generative models for minimal statistics of the data.

In the core chapter of the thesis **(chapter 3)**, to address the problem of the proliferation of algorithms that exist without comparability to known probabilistic models, we introduce the **ProbDR framework**—a probabilistic framework we develop by examining various dimensionality reduction methods used in science. It sheds light on how these methods are constrained and how they can be compared to each other. Through this framework, we observe *what* representation learning methods model—an estimate of the data's covariance. Furthermore, by studying algorithms that work well in practice, we can uncover modelling strategies that work well across domains and show that they are broadly coherent. Through the ProbDR framework, we demonstrate how many modern algorithms constrain their inference: by modelling only a

few aspects of the data. Therefore, specification of relatively simple models is possible without the risk of overfitting/overparameterisation (allowing for an Occam's-razor effect). In other words, through constraint imposed by the estimation of a very specific characteristic of the data, there is less misspecification, as all models, particularly latent variable models, tend to experience a mismatch between data characteristics and modelling assumptions.

A limitation of the work is that ProbDR is a *derived* framework, i.e., one obtained by studying classical methods as they are, which is not without issues, simply due to the nature of the algorithms it seeks to explain. For example, most methods examined perform inference for data characteristics in such a way that induction to new, unseen data points cannot be done easily. ProbDR interpretations share this characteristic. Moreover, the studied methods recover point estimates of latent variables; we leave the sampling behaviour analysis of ProbDR to future work. Other model frameworks may underpin representation learning more clearly, that are more interpretable, and that are easier to perform inference with. Specifically, we argue that appropriately constrained GPLVMs (Lawrence, 2005) and edge detection models presented in chapter 3 (section 3.3) provide a clearer route to the development of new methods. However, ProbDR achieves the goals set for a unifying framework and is, to our knowledge, the first of its kind to explain a wide variety of methods in the field from a probabilistic modelling perspective.

In the last part of the thesis **(chapter 4)**, as a validation of the study of the methods above, we show how the characteristics of the ProbDR framework extend to representation learning more widely. This exemplifies a core characteristic of probabilistic models, that insights from one area of study typically do extend to other areas where a model is reused. We focus specifically on the transformer architecture and show potential improvements through insights gleaned from the ProbDR framework. These insights will show that **transformers derived using our ideas result in a graph diffusion** step in place of an attention-based update, a change which leads to increased performance using nanoGPT on a language task and a vision task.

In the next section, we summarise the ideas of the thesis in more detail.

1.1 Thesis outline

The thesis is split into three content chapters, the background, the framework describing dimensionality reduction, and the extensions to representation learning.

In the **background**, chapter 2, a summary of the major ideas in representation learning is presented. The aim of the chapter is to show our perspective on how probabilistic methods in representation are organised, before showing that there are various algorithms that don't cleanly sit in this view. We categorise different ideas in probabilistic representation as,

- **Projective methods**: Many methods of constructing representations are functions of the inputs. These include random projections, featurising functions, density estimators, and functions that act as regressors and classifiers. We present ideas relating to the usage of such methods for science, showing what probabilistic models enable. As an example, we show that confidence levels of classifiers can correspond to how in-domain or out-of-domain an input is.

- **Generative methods**: These methods (often probabilistic models) find latent representations that describe the statistics of observed data or the data directly, such as covariances or distance matrices. Relevant observations that can be made about the presented model classes are described: for example, they demonstrate how model misspecification can arise and lead to failure modes in optimisation.

- **Dimensionality reduction and neighbour embedding methods**: Many methods for dimensionality reduction are motivated by matching a matrix of high-dimensional distances using a low dimensional matrix. Other methods perform link/edge prediction on graphs, with distances between learnt latent embeddings describing the probabilities of an edge existing between two nodes in the data graph. The latter can typically be formulated as contrastive algorithms. We pose the question, where should these methods sit? In the next chapter, we show that the algorithms correspond to inference algorithms in generative models for the covariance (or more generally, data similarity measures).

Through case studies, this thesis demonstrates how the usage of methods across climate science use-cases can be accelerated using probabilistic programming, how conservation activities are enhanced using principles of representation learning, how observations of models made in speech can facilitate the discovery of proteins for drug-discovery, and how constraints can indicate the properties that cancer-like cells share, thereby improving biological understanding for rare-disease research. Conversely, we also see what is shared by many of these case-studies: that they constrain many aspects of their models through case-specific estimation of some quantities that are of interest.

In the next chapter, we describe the **ProbDR** framework, that explains many classical coordinate-focused dimensionality reduction methods. Coordinate-focused meaning that the methods can be seen to explicitly represent and perform full-form inference for latent coordinates/embeddings, as opposed to seeking a parametric function that performs dimensionality reduction as a function of the data. This is an emergent framework, i.e., one obtained by studying what objectives dimensionality reduction methods optimise and interpreting these objectives as inference algorithms within probabilistic models. Due to the nature of probabilistic models, we show that "**what**" (i.e. the random variable, or the data statistic) is modelled by various DR methods can be unified coherently, i.e., with the assumptions resulting from model specification remaining unchanged after transformation of the random variable. We show (in section 3.2 and section 3.3) that every DR algorithm studied performs a variant of **covariance estimation**, within a model class where a covariance estimate $\mathbf{S}$ is described using a Wishart or inverse-Wishart distribution that has as the centrality parameter, a covariance kernel akin to those used with Gaussian processes,

$$\mathbf{S}|\mathbf{X} \sim \mathcal{W}^{\{-1\}}(\mathbf{X}\mathbf{X}^T + \beta K(\mathbf{X}, \mathbf{X}) + \gamma \mathbf{I}_n, \nu),$$

with latent $\mathbf{X}$ found through maximum a-posteriori inference. All hyperparameters (e.g., β, γ, ν), the choice of distribution (Wishart or inverse-Wishart) and data covariance estimators $\mathbf{S}(\mathbf{Y})$ are fixed and known, and depend on the algorithm that is being interpreted. The central matrix k corresponds to a non-linear kernel matrix, which as we shall see will be a Cauchy (rational quadratic) covariance kernel. The covariance estimator is the regular estimator $\mathbf{S} = \mathbf{Y}\mathbf{Y}^T/d$ only in the case of GPLVM and its extensions or simplifications, and in most cases involves instead (a pseudo-inverse of) a graph Laplacian (GL) matrix encoding a nearest-neighbour graph. We present efficient inference options inspired by SGNS (Levy and Goldberg, 2014) for some of the models, drawing connections between the two main interpretations within ProbDR that are of the form,

$$\texttt{transformed data} \sim \texttt{simple model}$$

and,

$$\texttt{raw data} \sim \texttt{complex model}.$$

Finally, in this chapter describing ProbDR, we show that inference within the model above is

equivalent to KL-minimisation, with the variational constraint over the covariance/precision parameter determined using the data, and being completely fixed at the outset of inference. This interpretation will be helpful for the last part of the thesis.

We will then focus on **extensions** in chapter 4, showing that constructions in wider representation learning, such as some self-supervised learning algorithms and transformers, can be seen to do approximate inference within ProbDR, and share similar ideas. We show that transformers correspond to unrolled optimisation, as presented by Yu et al. (2023), with a different perspective on the model used (described in section 4.3.2). This improves performance of the architecture—noting that in our interpretation of transformers corresponding to unrolled probabilistic Laplacian Eigenmaps, a stabilising constraint appears naturally as a graph Laplacian in the place of the classical attention matrix (interpreted as an adjacency matrix). We demonstrate that this leads to higher validation scores using nanoGPT on a language task and a vision task. This demonstrates an empirical application of the lessons learnt from the derivation of the ProbDR framework, in a setting that was not anticipated during the construction of the framework.

A visual summary of these ideas is given in fig. 1.1, showing that the thesis follows the plan described thus far. This is that the background presents the landscape of methods, then, we interpret the methods as maximum likelihood methods within the ProbDR framework, which have equivalent variational views, and finally show that ideas of the framework can be related to transformers, an unintended case-study for our work.

In the next section, we list the publications that make up the thesis to highlight the specific contributions made.

1.2 Publication list and contributions

Major contributions, i.e. (joint-)first and second author publications, and two papers with significant software contributions are presented below, and throughout the thesis, these are coloured grey to highlight the contributions and original material in the thesis, for example, **Ravuri et al. (2023)**. Details of contributions are listed below. Code for all presented results is at **github:infprobscix/phd**.

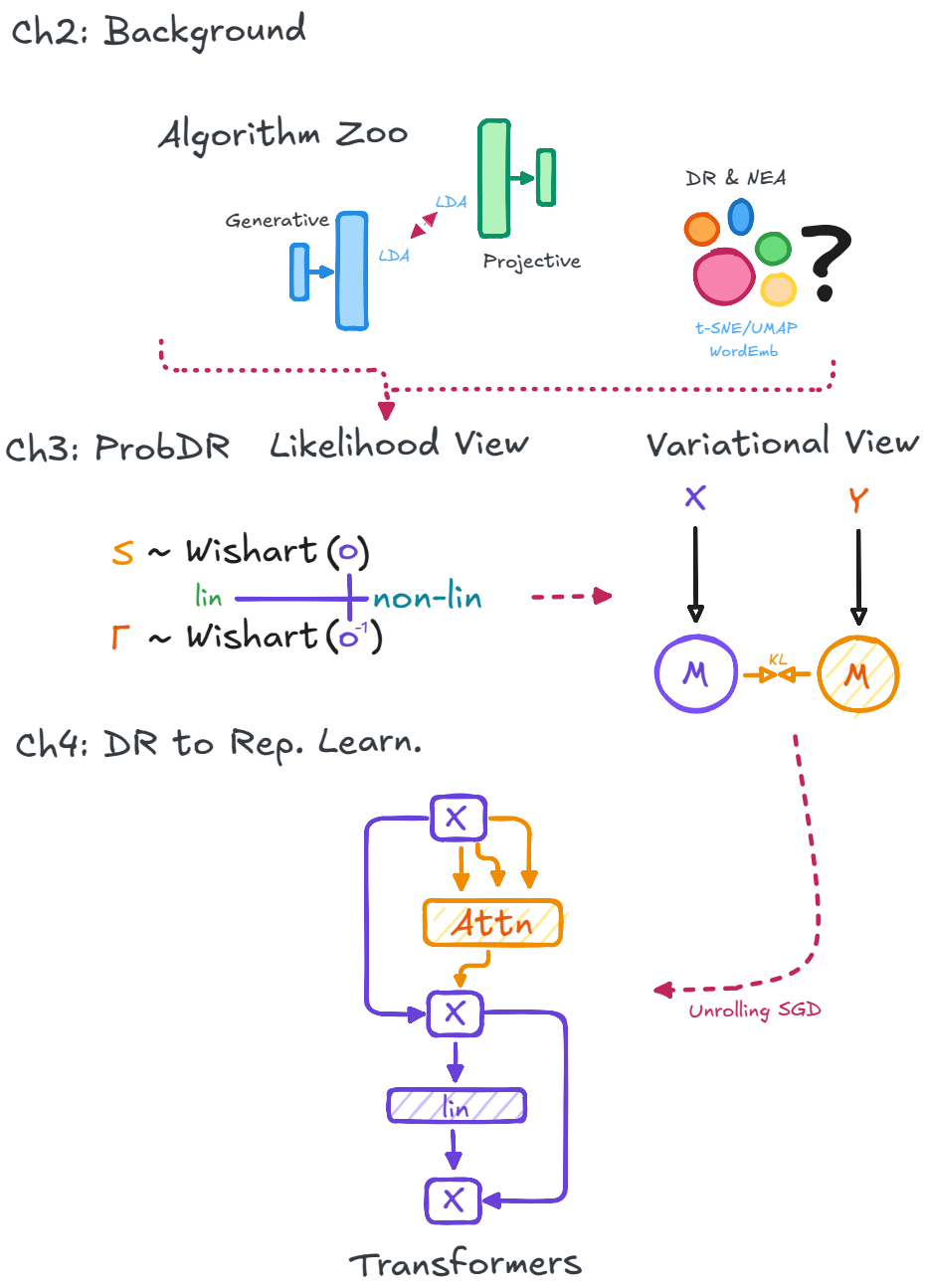


Figure 1.1: A visual summary of the thesis. In the **background**, we present the landscape of large ideas in representation learning showing projective and generative methods, and equivalences between them, finally posing the question, where should neighbour embedding (and other dimensionality reduction algorithms) sit? In the **ProbDR** chapter, we show that (quasi) maximum-likelihood estimation for the covariance/precision leads to many of the algorithms in the field, with connections between them, and that this has an equivalent KL-minimisation view. We then show in the **transformers** chapter that ideas of the framework extend to SSL, and KL minimisation in probabilistic Laplacian Eigenmaps can explain the transformer architecture.

Case Studies in Science (mentioned in Chapter 2 and the Appendices)

- **Ravuri and Lawrence (2025a)**: Protein Language Model Zero-Shot Fitness Predictions are Improved by Inference-only Dropout, *1st author, Workshop Track @ MLCB, 2025.*

Contributions include all technical ideas, experiments, code and write-up.

- **Ravuri et al. (2025)**: Weakly supervised latent variable inference of proximity bias in crispr gene knockouts from single-cell images, *1st author, LMRL @ ICLR, 2025.* Contributions include the main implementation (excluding the fine-tuning of the auto-encoder that was used and the single-cell centroid identification of RXRX3 images) and the probabilistic interpretation of the method.

- **Ravuri et al. (2024b)**: On Feature Learning for Titi Monkey Activity Detection, *1st author, VIHAR @ Interspeech 2024.* Contributions include all technical ideas, experiments and write-up. Jen Muir contributed the subject-matter expertise, experimental direction, validation and datasets. This paper is a short technical summary and experimental extension of a longer piece of work for which Eric Meissner also contributed code and the initial implementation that was a precursor to this work.

- **Ravuri et al. (2024a)**: Uncertainty as a predictor: Leveraging self-supervised learning for zero-shot MOS prediction, *1st author, SASB @ ICASSP 2024.* Contributions include most technical ideas, experiments, code and write-up.

- **Zhao et al. (2024)**: Scalable Amortised GPLVMs for Single Cell Transcriptomics Data, *2nd author, MLGenX @ ICLR 2024.* Acted as advisor and peer-coder on this project and set up a minimal script that achieved scVI performance using a linear variational GPLVM. SZ performed all ablations of this code, derived insights, and drafted the publication.

- **Ravuri et al. (2022b)**: Modelling technical and biological effects in scRNA-seq data with scalable GPLVMs, *joint 1st author, MLCB 2022.* Contributions include main technical interpretation and secondary interpretations in the appendices, the experiment related to the replication of the results of Kumasaka et al. (2021) and most of the software.

Software Papers (mentioned in Chapter 2 and the Appendices)

- **Lalchand et al. (2022)**: Generalised GPLVM with stochastic variational inference, *2nd author, 2022, AISTATS.* Contributions include a large part of the software, its design, proof-reading, project input, and multiple experiments (including the `mocap`, `mnist`, `pca-flow`, `poisson` experiments).

- **Ravuri et al. (2022a)**: Ice Core Dating using Probabilistic Programming, *1st author, AGU, & GPSMDS @ Neurips 2022.* Contributions include all of the implementation, some project steering, and drafting of the publication.

- **Griffiths et al. (2023)**: GAUCHE: a library for Gaussian processes in chemistry, *joint 1st author, NeurIPS 2023.* Contributions include a wrapper that makes external graph kernels usable via plug-and-play, and two experiments (that perform scalar property prediction using the wrapper, and another experiment that uses `torch`'s Weisfeiler-Lehman feature function).

- **Mostowsky et al. (2024)**: The GeometricKernels Package: Heat and Matérn Kernels for Geometric Learning on Manifolds, Meshes, and Graphs, *JMLR, 2025.* Contributions include the implementation of kernels on graphs and the `scipy.sparse` interfacing.

Core theoretical contributions (these form Chapter 3: ProbDR)

- **Ravuri et al. (2023)**: Dimensionality Reduction as Probabilistic Inference, *1st author, AABI & Stancon 2023*[1]. Contributions include all technical ideas, experiments, code and write-up, except for Appendix D of the paper (a mean-field view), which is not presented in this thesis. Francisco Vargas devoted a lot of time for proof-checking ideas and proof-reading the math. VL helped with proof-reading and provided feedback on the writing.

- **Ravuri and Lawrence (2024)**: Towards One Model for Classical Dimensionality Reduction: A Probabilistic Perspective on UMAP and t-SNE, *1st author, AABI 2025.* Contributions include all technical ideas, experiments, code and write-up.

Extended contributions (this forms Chapter 4: Extensions of ProbDR)

- **Ravuri and Lawrence (2025b)**: Transformers as Unrolled Inference in Probabilistic Laplacian Eigenmaps: An Interpretation and Potential Improvements, *1st author, SPIGM,*

[1]A version of this work was first submitted for review in February 2022 as "A Unifying Probabilistic Perspective on Graph Latent Variable Models" and was presented in poster form throughout that year. It introduced the core graphical model and a variational interpretation of the (t-)SNE and UMAP objectives. Independently, the variational idea appeared in the second preprint of Van Assel et al. (2022), released in June 2022. As these developments occurred in parallel, they are treated as concurrent work. To the best of our knowledge, our dual-probabilistic formulation of MCA has not appeared in prior literature.

UniReps @ NeurIPS Workshops 2025. Contributions include all technical ideas, experiments, code and write-up.

Other contributions throughout the thesis

Unpublished or ideas in the appendices that use original work are highlighted grey.

CHAPTER 2

Background

In the thesis, we aim to uncover the probabilistic models that underpin methods in representation learning. In the background, we establish a description of the probabilistic terminology used, and present an example of how probabilistic interpretations are constructed, which is needed for the presentation of the ideas in later chapters. In addition to this, we also contextualise the thesis by providing an overview of ideas in probabilistic representation learning, organised as **projective** and **generative** models. Yet a significant number of algorithms in the field lack explicit probabilistic interpretations; we provide a brief review of such algorithms. This absence raises a critical question that drives our inquiry: where should these algorithms sit given our categorisation and what, in modelling terms, do they do? We briefly mention case-studies (and expand on them in appendix A) showing how the ideas of the background appear in scientific modelling, to lend credibility to the ideas presented.

In later chapters, we interpret the dimensionality reduction and neighbour embedding algorithms introduced in section 2.5 as performing covariance estimation or edge-detection. All the while, as we see in our scientific examples, the algorithms constrain degrees of freedom by using specific estimators for statistics of the data. For example, some methods use the graph Laplacian as an implicit proxy for the data covariance, rather than using the standard estimator. Future work should focus on precisely what these estimators estimate in high-level terms.

The sections in this chapter are organised as follows. Section 2.1 provides a background on the mathematical tools within probability and statistics useful for scientific modelling. Section 2.2 shows how one can construct probabilistic interpretations from loss functions, and particularly, we see how variational interpretations can arise by studying the Griffin-

Lim algorithm. In section 2.3 and section 2.4, alongside providing an exposition of major ideas in probabilistic representation learning, we also mention examples in science, showing that estimators constrain degrees of freedom and/or showcase probabilistic models being successfully used in ways that perhaps contrasts with their intended use-case. The case-studies are provided as an **ideological prologue** to the rest of the thesis, but we leave a longer exposition of the ideas to the appendix. In section 2.5, we provide a background on commonly used dimensionality reduction methods and representation learning methods, which all use the notion of data-data similarity and ask the question: where should these algorithms sit within our taxonomy?

2.1 Background on probabilistic modelling

In this section, we provide a brief overview of the building blocks of probabilistic and statistical modelling, vital to the exposition of model constructions and probabilistic interpretations that make up the majority of the thesis.

Statistical modelling is a framework for science[1]. A modeller follows the process illustrated in fig. 2.1, by specifying a model representing variables of interest and their joint relationships, and performing inference given the model to say something about unobserved variables given the data, generating knowledge[2] (Gelman et al., 2013; Murphy, 2012). Standard texts on statistical modelling cover the mathematical content of this section, including Gelman et al. (2013); Murphy (2012).

The section proceeds as follows. We will describe the basic mathematical setup of probabilistic modelling, then provide a commentary on building models and performing inference within them, and finally give a brief commentary on how models form high-level probabilistic grammars.

[1]There are many parallels between aspects of statistical modelling and the philosophy of science. For example, in classical hypothesis testing, one never "proves" a hypothesis, only fails to reject it, in accordance with ideas relating to falsification (Popper, 1959). Inference results and hypotheses cannot be detached from the underpinning model, in accordance with ideas in the philosophy of science around theory-ladenness of observations and the Quine-Duhem thesis, which says that a hypothesis cannot be tested in isolation. Ideas such as Occam's razor and the strength of induction can be formalised within classes of models using model-fit evaluation metrics, such as likelihoods and information criteria. Moreover, the extent to which models represent the world, and whether they are to be understood simply as abstract models of logic that shed light on emergent phenomena, relate to ideas in scientific realism.

[2]Data analysis can also be seen as a special case of this process; for example, scatter plots represent visual estimators of a (conditional/joint) distribution's statistics.

2.1.1 Mathematical setup

This subsection provides a brief mathematical overview of the terms used in the thesis. A variable of interest is typically represented as a **random variable**, defined as a function $X : \Omega \mapsto \mathbb{R}$, that represents an event $\omega \in \Omega$ as a real number, where $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space; Ω represents the set of possible outcomes, $\mathcal{F}$ is the event space, representing events that may be observed. This set contains combinations of the elements/sets in Ω, allowing for more complex questions to be posed regarding the probability measure. For example, Ω may contain six elements corresponding to the sides of a six-sided die, whereas the event space is the sets of subsets of Ω (formally, a $\sigma-$algebra), allowing one to define probabilities of events such as $\{6\}$—i.e., a die landing on a six, or $\{4, 5, 6\}$—i.e., the die landing on four or above. A probability measure finally, $\mathbb{P} : \mathcal{F} \mapsto [0, 1]$, is a function that maps elements of the event space $e \in \mathcal{F}$ to a number between zero and one. A **stochastic process** $\{X_t\}_{t \in T}$ is a collection of random variables, indexed by a set T. The probability measure must satisfy four key properties, attributed to Kolmogorov (Kolmogorov, 1950).[3] The first three **axioms** are that,

1. it is non-negative,
2. the probability of observing *some* event in the event space is one, and,
3. the probability of observing one of a disjoint countable set of events is the sum of the probabilities of the events.

The fourth is a **definition** that provides the grammar with which to reason about conditional probabilities and a way to compute the probability of an event given another, $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$.

A probability density p is a function that, if it exists[4], obeys $\int_{x \in A} p(x) d\mu(x) = \mathbb{P}(X \in A)$, where μ is a measure (e.g. the Lebesgue measure), with respect to which p is defined.[5] As an example, consider a biased coin, with a probability of heads given by π. A model for observations of the coin may be represented by a collection of n identically distributed random variables, $\{X_i\}_{i=1}^n$. As this sequence of variables is exchangeable (i.e. the joint distribution is

[3] The axiomatization of probability, disentangling it from the two main interpretations of probability in terms of aleatory and epistemic uncertainty, was a key development in the field.

[4] There are distributions that can be defined on sets that have no valid Lebesgue measure, and therefore, a density with respect to the Lebesgue measure cannot exist.

[5] The definition of the probability density can be extended more generally as a Radon-Nikodym derivative, but this consideration is not needed for the work presented in this thesis.

invariant to permutation before the sequence is observed) and infinitely extendable, the theorem of De Finetti shows that there exists a **random variable** Π that takes values in $[0, 1]$, such that $\forall i : X_i|\Pi \sim \text{Bernoulli}(\Pi)$; given Π, the sequence becomes conditionally independent and identically distributed. This shows the treatment of a "parameter" (the variable that determines a distribution's properties) as a random variable[6]. Bayes' rule provides a way to do induction, i.e. inference for the unobserved random variable Π, given the ones that are observed, $\mathbf{X}_i$s,

$$\mathbb{P}(\Pi|X_1, \ldots, X_n) \propto \mathbb{P}(X_1, \ldots, X_n|\Pi)\mathbb{P}(\Pi).$$

Here, $\mathbb{P}(O|U)$ is typically given by a **model**, and $\mathbb{P}(U)$ describes the expected **prior** behaviour of the unobserved variable before any data is seen. Having established the basic axioms of probability, we now look at how these rules are composed into the *models* that form the basis of representation learning.

2.1.2 Probabilistic models

In this subsection, we describe briefly the usage of the term "model" within the thesis, before describing ideas of inference and probabilistic grammars. Probabilistic models enable communication of assumptions through explicit definitions of **what** is considered (i.e. random variables, stochastic processes) and **how** they behave—concretely through distributional assumptions, or through higher-level abstractions known to represent objects or processes of interest (Ghahramani, 2015; Gelman et al., 2013).

We define a probabilistic or statistical model (used interchangeably), for the purposes of this thesis, as a collection of observed and unobserved random variables and the probability distributions that characterise them, where the random variables represent, at least approximately, a hypothesised/abstracted real-world process—a data-generating process. Unobserved parameters govern observed relationships, with the model describing the joint behaviour of all random variables within the abstracted process. The behaviour of some of the random variables of the model is typically the object of study, and this process is termed inference.

[6]As opposed to another historical treatment of them as constant unknowns, as seen in Bernoulli's theorem (a case of the weak law of large numbers), which shows that $\bar{X} \to \pi$ where $\pi \in [0, 1]$ is the unknown constant. The different methods are known as "inverse-probability" on account of the object $\mathbb{P}(\text{parameter}|\text{data})$ being the object that is studied, as opposed to $\mathbb{P}(\text{data}|\text{parameter})$. This is however tangential to the separate debate regarding the interpretation of probability as measuring epistemic or aleatoric uncertainty—both sides for which, historical cases exist, that made quantitative measurements of the probabilities.

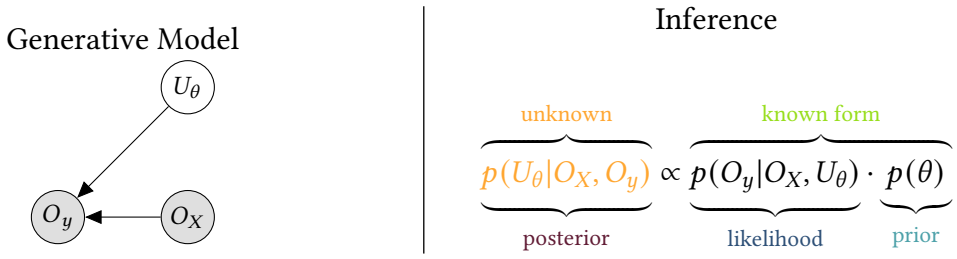


Figure 2.1: An illustration of a simple statistical modelling process. A model, in this case for knowns O_X, O_y and unknowns U_θ, is first written out as a graph representing the data generating process, then, inference follows the determination of the posterior over the unobserved variables using what is known, using Bayes rule.

The Bayes' rule introduced in the previous subsection (section 2.1.1) is the central argument with which inference can be done in Bayesian modelling. Figure 2.1 illustrates that the posterior over unobserved variables U is calculated as being proportional to the "likelihood" and "prior" functions. The function $\mathbb{P}(O|U)$ is calculated as the probability density taken by the model as a function of the unobserved variables or parameters $U(= U_\theta$ in fig. 2.1), evaluated at $O(= \{O_X, O_y\}$ in fig. 2.1), and is the **likelihood**. The likelihood principle (Birnbaum, 1962) posits that everything needed for the inference of U related to the data O is encoded by this function. The **prior** distribution encodes behaviour of the "unknowns" that is expected a priori, before experimentation[7]. We now present how this function is used in practice to extract information about unobserved quantities.

2.1.3 Inference

In this section, we introduce ways in which, given a model, how inference about unknowns can be done. Inference typically involves understanding the behaviour of the posterior distribution of the variables with which we are concerned. Some model and prior combinations, yield analytical posteriors, allowing one to read off characteristics (e.g. moments) and use standard samplers to compute quantities such as expected values of downstream variables, accounting for the uncertainty in the variables over which inference is done. In most real-world cases however, this is not possible. Below, we focus on three key ideas on approximating the posterior: by obtaining a point estimate for the parameters at which the posterior density is maximised, or by using methods that sample approximately from the posterior, and finally approximating

[7]An example of a prior is a Gaussian process, used to represent a distribution over an unknown function (Rasmussen and Williams, 2005). The distribution of the process can constrain the function space to smooth functions, periodic functions, etc.

the posterior by an analytical distribution that is "closest".

The following ideas can be motivated in several ways, for example, MAP estimation, which recovers point estimates, has a KL-minimisation view (Murphy, 2022). Another view follows by studying the behaviour of new data given the posterior, i.e., the posterior-predictive distribution, for example computed as $p(x^*|x) = \mathbb{E}_{p(\theta|x)}(p(x^*|\theta))$ when the posterior factorises appropriately. One can then approximate the expectation by assuming that the posterior is a Dirac-delta at the posterior-maximising value of the parameters. Lastly, in the risk-minimisation view of Bayesian decision-theory, one computes risk across scenarios determined by a posterior, to marginalise out uncertainties from abstract parameters, and describe the behaviour of actionable variables. This can be formulated as a scalar risk of an action A, $R(A) = \mathbb{E}_{p(\theta|x)}(L(A, \theta))$, where L is a *loss function* that measures the cost of acting as per A. MAP estimation can be recovered using specific choices of the loss function.

We start our presentation of inference ideas with point estimation of the parameters of the model—i.e. summarising the posterior over a random variable with one "representative" scalar.

Maximum A-Posteriori Inference

Abbreviated "MAP" inference, this results in point estimates by maximising the posterior with respect to parameters,

$$\hat{\theta} = \arg\max_{\theta} \log p(\theta|x).$$

In this thesis, we use the terminology "**maximum-likelihood estimation**" to mean the point-wise estimation of parameters by simply maximising the likelihood function (equivalent to the MAP view when the prior over the parameters is proportional to one). This is somewhat in contrast to the term's historical context, as maximum-likelihood estimators in i.i.d. data contexts are known to be asymptotically consistent (achieving the true parameter value in probability, under identifiability constraints) and efficient (i.e. achieving the best possible variance, the Cramér-Rao lower bound). Most problems we study in the thesis do not correspond to the i.i.d. data regime, and they present with significant model unidentifiability; unidentifiability arises when multiple parameter combinations lead to the specification of the same model. Therefore the properties known of maximum-likelihood estimators do not hold in our settings.

Although these estimates are often computationally tractable through optimisation, the likelihood/posterior may contain multiple local maxima (corresponding to different interpre-

tations of the data), or multiple global maxima (where the posterior is invariant to certain transformations in the parameter space). This estimation method can also fail when there are variables that need to be integrated/marginalised out. In these scenarios, Type-II MAP/MLE (maximum likelihood) estimation, is typically used,

$$\hat{\theta} = \arg\max_{\theta} \log p(x|\theta) = \arg\max_{\theta} \log \int p(x, w|\theta) dw,$$

which removes the effect of "nuisance" or uninteresting latent variables.

Next, we consider cases where point estimates fail or do not provide enough information.

Sampling, via Markov Chain Monte Carlo

Marginalisation of unobserved random variables may be needed if maxima with respect to the posterior distribution lie outside the typical set. Such a circumstance typically leads to uninteresting parameter choices that do not lead to good generalisation. Then, we sample directly from the posterior and analyse the behaviour of the distribution. Where the posterior does not have a known form, we use approximate sampling methods, for example, Markov-chain Monte-Carlo (MCMC) methods to build Markov chains of parameter samples, such that the marginal distribution of samples of these chains is proportional to the posterior. With samples obtained, one typically calculates expected values of statistics under the posterior distribution using Monte-Carlo,

$$\mathbb{E}_{p(\theta|x)}(f(\theta)) \approx n_s^{-1} \sum_{\theta \sim p(\theta|x)} f(\theta).$$

Sampling sheds light on uncertainties and other characteristics of a posterior (e.g., alternative interpretations of the data, such as whether a smooth high noise model fits the data or whether a jagged low noise function does, as demonstrated in Rasmussen and Williams (2005)), but it can be computationally expensive, and where the posterior is complex, building practical samplers can be difficult. Building samplers with desirable properties is a key effort in modern computational statistics. Therefore, we may be interested instead to perform approximate inference by approximating the posterior by one that has an analytical form, as we show below.

Variational Inference

When the treatment of uncertainty is desired but a full Bayesian treatment is computationally infeasible, we approximate a posterior using a simpler, analytically tractable distribution for reasons such as ease of understanding and sampling. In other scenarios, one may know the posterior factorises in a specific manner, and a variational posterior formulated according to that factorisation can aid in narrowing the search space for the posterior.

Given a posterior over latent variables z and parameters θ, $p_\theta(z|x)$, we aim to find an approximate posterior parameterised by ϕ, $q_\phi(z|x)$, such that the KL divergence $\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))$ is minimised. The metric arises naturally when certain arguments involving log-probabilities are made.

This divergence is often intractable, so we optimise a lower bound on the model evidence, known as the evidence lower-bound (ELBO). Consider the KL-divergence,

$$\begin{aligned}
\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) &= \int \log \frac{q_\phi(z|x)}{p_\theta(z|x)} q_\phi(z|x) dz \\
&= \mathbb{E}_q(\log q_\phi(z|x)) - \mathbb{E}_q\left(\log \frac{p_\theta(x|z)p(z)}{p_\theta(x)}\right) \\
&= \log p_\theta(x) - \mathbb{E}_q(\log p_\theta(x|z)) + \mathbb{E}_q\left(\log \frac{q_\phi(z|x)}{p(z)}\right) \\
&= \log p_\theta(x) - \underbrace{\left(\mathbb{E}_q(\log p_\theta(x|z)) - \mathrm{KL}(q_\phi(z|x)||p(z))\right)}_{ELBO} \geq 0.
\end{aligned}$$

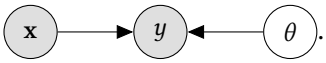
This shows that the term labelled the ELBO is less than or equal to the evidence. It also shows that, as the evidence $p_\theta(x)$ is constant w.r.t. ϕ[8], maximising the ELBO minimises the KL divergence between the approximate and true posteriors. Therefore, with variational inference, we specify an approximate posterior and turn the sampling problem into an optimization problem, wherein the ELBO is maximised as a function of ϕ and θ. *All* KL divergences in this thesis appear as the above—as backwards KL, $\mathrm{KL}(q||p)$.

This concludes our presentation of inference ideas, and we now provide a brief commentary on probabilistic models corresponding to high-level logical statements.

[8]The KL divergence is positive, therefore the ELBO forms a lower bound for the evidence even when we optimise for model parameters θ.

2.1.4 Probabilistic grammars and software

Now that we have reviewed the mathematics of statistics, models and inference within them, we briefly provide a short commentary on a core motivation for our search for probabilistic interpretations, which is that probabilistic models can correspond to high-level modelling semantics (which we detail further in the background, particularly for PCA and ICA). This section provides a short list of such high-level semantics, which we also call high-level "grammars" (as concrete Backus-Naur grammars can be specified to translate between high-level ideas and statements used, and probabilistic models), frequently used when building probabilistic models. Our view follows Jaynes (2003), who treats probability theory as the logical components of science. For us, a probabilistic grammar is a framework for composing this logic into complex models. Future work can uncover such semantics corresponding to the models presented in later chapters, specifically, what a "covariance" means in high-level terms, especially when constructed using binary data.

Statistical models represent the subset of the world in which a scientist is interested, and statisticians typically design models based on known abstract grammars that are associated with specific collections of probabilistic statements. Moreover, scientific model pipelines are typically constructed by chaining together multiple semantic components. For example, consider the model class typical in regression (e.g. generalised linear and additive model, GLM and GAM) contexts: 

. Given a positive random variable to model for example, in the absence of distributions derived from ground-up reasoning (e.g. limiting behaviour in the underlying physical systems), one may choose a Gamma distribution (an exponential family distribution) as it corresponds to a maximum-entropy distribution with a specific mean and mean-log. Alternatively, if the tails of the log-distribution will be symmetric, one would choose a log-normal, and a Pareto to model fat-tailed or extreme-value distributions.

Statisticians use such high-level semantic abstractions to construct their models, and hence we may lay out grammars that can abstract away such decisions to make a wide variety of models available for common use. For instance, software such as BRMS (Bürkner, 2017) acts as a compiler for a high-level probabilistic programming language (PPL, see van de Meent et al. (2021) for an introduction to the topic) with a syntax,

```
target_statistic {| hyperparameters = ...} ~ covar_a + s(covar_b)
```

that translates high-level parameter relationships to concrete probabilistic models within the

Stan ecosystem (Carpenter et al., 2017). Such a syntax completely abstracts away distributional choice and functional form for the relationship between statistics of the distribution to covariates, through a link function and parametric functions represented by `s`. Such grammars can describe a very large class of statistical models with a hierarchical structure, and are transpiled to lower level PPLs, where the specification of more fine-grained statements is possible, but a statistician would typically still rely on some level of abstraction to choose distributional assumptions for their data. We see PPLs (and other implementations) on at-least three levels, high-level grammars that allow for models to be specified by domain-experts, middle-level grammars that allow for finer specification of assumptions, but such that the probabilistic statements are still easy to communicate[9], and low-level implementations that cater for bespoke/complex modelling or inference needs.

As we describe the models presented in this thesis, we will aim to consider the **middle-layer of modelling semantics**: what real world assumptions do models implied by widely-used representation methods correspond to? We will show that, in the case of dimensionality reduction methods, these semantics correspond to covariance estimation using Wishart models or edge-detection using Bernoulli models, using logical parameterisations (in that a statistician would choose those models naturally, given the characteristics of the data at hand). We leave the exploration of higher-level semantics (e.g. what truly does a nearest neighbour graph estimated using zero-inflated data correspond to? Does it measure co-occurrence/co-activation rates?) for future work.

As we consider these questions, it is useful to keep in mind that models can be written as multiple equivalent sets of statements, with certain representations that may lend themselves to model extensions that are more intuitive given a problem.[10] Therefore, we argue, a description of multiple interpretations of one algorithm can be useful as different interpretations may lend themselves to different use-cases. In chapter 3, we show that t-SNE-like algorithms can have three different interpretations, a minimal Bernoulli interpretation that can be understood with ease, a Wishart interpretation that makes the model comparable to others (e.g. GPLVM), and a KL-minimisation view that lends itself to other use-cases, such as transformers as explored in

[9]For example, if a coin toss is modelled as a Bernoulli distribution with a probability parameter based on the mass distribution of a coin, it is easy to communicate why this model construction was chosen.

[10]Compound distributions are an example; a normal distribution with an exponentially distributed variance, for example, is equivalent to a Laplace distribution. A logistic regression can be interpreted to be a real-valued latent variable model, with an observation model that forces the latent variables to take categorical values. These models lend themselves more naturally to extensions, e.g. to ordinal regression.

chapter 4.

Abstracting from models to their semantic representations can make these models more accessible, especially to domain experts, to use statistical methods for their work. We motivate our search for a unified interpretation for dimensionality reduction methods and beyond, not only due to ease of transferability across implementation platforms (powered by PPLs at some level of abstraction), but also to work towards a semantic representation of the algorithms used in the field.

In this section, we have reviewed the mathematical basis of probabilistic modelling, how inference can be done, and we shared a view that portrays useful probabilistic models as ones that can correspond to natural language. In the next section, we show how probabilistic interpretations can come about, before reviewing projective and generative models for representation learning, and presenting a discussion of algorithms without any known probabilistic interpretations.

2.2 Formulating probabilistic interpretations

Having reviewed foundational ideas of probabilistic modelling in the last section, in this section, we describe some methods by which probabilistic interpretations, which make up a majority of this thesis, can be formulated. In the thesis, we interpret arbitrary objective functions that act on data as being related to the posteriors of probabilistic models, so that implicit modelling assumptions can be read off as motivated in the introduction. One way by which loss functions $\mathcal{L}$, minimised with respect to parameters $\boldsymbol{\theta}$, can be interpreted is by viewing the optimisation process as MAP-inference, with the objective interpreted as a negative log-posterior or an upper bound on it,

$$\mathcal{L}(\boldsymbol{\theta}) \geq -\log p(\boldsymbol{\theta}|\mathbf{x}) + k.$$

Another way of interpretation follows by viewing gradient components of unrolled optimisation (as we shall see corresponds to neural network architectures in chapter 4) as score functions of the negative log posterior in updates such as,

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \eta \nabla_{\theta} \underbrace{\mathcal{L}(\boldsymbol{\theta})}_{-\log p(\boldsymbol{\theta}|\mathbf{x})},$$

and update steps in the form of,

$$\boldsymbol{\theta} \longleftarrow f(\mathbf{x}),$$

are interpreted such that $f(\mathbf{x})$ calculates analytically the (arg-)maximum of $\log p(\boldsymbol{\theta}|\mathbf{x})$ (for example, f could represent the singular value decomposition of a matrix, which is known to result from the optimisation of a squared-error objective within the realm of matrix factorisation). Another example of such an interpretation involving discrete optimisation is given in appendix A.1.

There is one more case that is useful for the discussion in the thesis, which is how variational interpretations arise when working with objectives that seem "circular". Such objectives/cases occur widely in machine learning, as we see in section 3.4 and chapter 4. We now discuss an example of such a case in the upcoming subsection.

2.2.1 Variational interpretations and the Griffin-Lim algorithm

Some algorithms and objectives appear circular; this section shows how to interpret such methods—using variational ideas. This view will be important to understand our framework ProbDR from a different perspective in section 3.4, and self-supervised methods and transformers in chapter 4.

When a computational pipeline looks circular, e.g. in autoencoders $\mathbf{Y} \to \mathbf{X} \to \mathbf{Y}$ with $\mathbf{Y}$ representing data and $\mathbf{X}$ latent variables, a variational explanation typically exists. For example, Kingma and Welling (2014) introduce the variational autoencoder that makes inference in this setting statically sensible—with the (generative) model corresponding to edge $p : \mathbf{X} \to \mathbf{Y}$ and the variational approximation describing the unknowns using the data $q : \mathbf{Y} \to \mathbf{X}$, as posteriors are typically formulated.

In this section, we show that the Griffin-Lim algorithm (GLA, Griffin and Lim (1984)), a phase-reconstruction algorithm used widely for generating speech from spectrograms before the advent of neural vocoders, corresponds to an objective which does not have a simple MAP-interpretation. The squared-error objective that underpins the method involves the data, in a sense, on both "sides". We will show that a simple variational interpretation explains such an objective.

The complete details of the case-study are provided in appendix A.2. GLA finds unknown

phases $\boldsymbol{\theta}$ given the spectrogram $\mathbf{S}$. Let the audio sequence be represented as,

$$\tilde{\mathbf{a}} = \mathbf{D}_k^{\dagger} \operatorname{vec}(\mathbf{S} \odot \exp(\boldsymbol{\theta} i)),$$

where $\mathbf{D}_k^{\dagger}$ corresponds to the inverse Fourier transform. Then, GLA minimises a squared-error objective between the audio sequence and the real part of the audio (so that no imaginary part is recovered by $\boldsymbol{\theta}$ as expected of typical audio sequences; Masuyama et al. (2019)),

$$\mathcal{L} = \|\boldsymbol{\Pi}\Re\left(\tilde{\mathbf{a}}\right) - \tilde{\mathbf{a}}\|_F^2 = \Re(\tilde{\mathbf{a}})^T(\mathbf{I} - \boldsymbol{\Pi})\Re(\tilde{\mathbf{a}}) + \Im(\tilde{\mathbf{a}})^T\Im(\tilde{\mathbf{a}}).$$

The matrix $\boldsymbol{\Pi}$ describes how points in $\tilde{\mathbf{a}}$ are related due to redundancy in the spectrogram (the matrix $\mathbf{I} - \boldsymbol{\Pi}$ has a graph Laplacian interpretation). The loss effectively selects $\boldsymbol{\theta}$ that ensures that the audio recovered is real-valued and consistent (due to the action of $\boldsymbol{\Pi}$) given redundancies in the spectrogram. We show in appendix A.2 that this is the log-density (up to constants) of the model,

$$\begin{bmatrix} \Re(\tilde{\mathbf{a}}) \\ \Im(\tilde{\mathbf{a}}) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} ((1+\delta)\mathbf{I} - \boldsymbol{\Pi}) & \mathbf{0} \\ \mathbf{0} & (1+\delta)\mathbf{I} \end{bmatrix}^{-1}\right). \tag{2.1}$$

We add a small jitter matrix to the true precision for non-degeneracy reasons, $\delta\mathbf{I}$, which simply corresponds to adding a small amplitude term to the objective (and is independent to the phases being estimated). The implied covariance (calculated in appendix A.2) shows that the variance corresponding to the imaginary part of the audio is far smaller than the real part, and that there are correlations across audio points that are connected due to redundancies in the spectrogram, in line with what is expected of typical audio sequences. This example shows that probabilistic models can correspond to compact generative accounts of the random variables they model.

Due to the random variable of interest (the phases $\boldsymbol{\theta}$) appearing non-linearly on the left-hand side of the sampling statement however, this is a non-standard model statement. In such cases, objectives such as the above can be thought of as KL divergences—we will return to this

idea concretely in section 3.4. If a model and variational constraint are set up as the following,

$$p(\mathbf{z}) = \mathcal{N}\left(\begin{bmatrix}\Re(\tilde{\mathbf{a}})\\ \Im(\tilde{\mathbf{a}})\end{bmatrix}, \begin{bmatrix}(1+\delta)\mathbf{I} - \mathbf{\Pi} & \mathbf{0}\\ \mathbf{0} & (1+\delta)\mathbf{I}\end{bmatrix}^{-1}\right) \text{ and,}$$

$$q(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\mathbf{z}$ is an arbitrary random vector representing a hypothetical audio sequence. Then, the KL divergence $\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))$ results in the GLA objective (with $\delta \approx 0$) as,

$$\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})) \overset{+}{=} \frac{1}{2}\left[\underbrace{(\mu_q - \mu_p)\Sigma_p^{-1}(\mu_q - \mu_p)}_{\mathcal{L}} + \underbrace{\mathrm{tr}(\Sigma_p^{-1}\Sigma_q) - \log\det(\Sigma_p^{-1}\Sigma_q)}_{c}\right].$$

Such interpretations will form the basis of section 3.4 and chapter 4, in which we show that the variational constraint in many ways acts as a form of "data". The variational distribution q acts as a variational constraint and not as an estimator of the true posterior. This form of constraint is seen with de-noising diffusion models Ho et al. (2020).

Probabilistic interpretations can arise in other ways, e.g. through a Bayesian decision theoretic framework (where the objectives are related to utility, risk, or loss functions), approximate inference frameworks (where the objectives are interpreted as specific lower bounds on posteriors), or by studying algorithms as natural gradient descent algorithms, assuming a probabilistic model. The Bayesian learning rule Khan and Rue (2023) is an example of the latter, where a variety of algorithms (such as dropout, and the Adam optimiser introduced in Kingma and Ba (2017)) used in deep learning can be framed as natural-gradient-based minimisation algorithms, assuming a family of posterior-constraining variational forms. Such interpretations are useful for the study of the choice of architecture/algorithms as this have a large impact on inference, and example of which is described in appendix A.3. We leave the exploration of such ideas and their application to the methods studied in this thesis for future work.

We conclude the methodological background, where we have shown how models and probabilistic interpretations are constructed. In the upcoming sections, we will present a view of probabilistic representation learning that we believe summarises ideas of the field, to be able to consider where widely-used algorithms without probabilistic interpretations should sit.

2.3 Projective representation learning

Now that we have presented the mathematical background, we review existing probabilistic representation learning methods used in the field to contextualise our core work, by categorising the methods into two groups: **projective** and **generative** methods. Projective methods generally follow the rule, data → representations, as opposed to generative methods that act in the reverse direction. The split is not a rigid one, as equivalences exist between many projective methods and their generative counterparts (e.g. logistic regression and linear discriminant analysis). In practice, the difference between the two views reduces to whether a notion of non-statistical independence exists in one direction, i.e. whether the pair $p(y|x)$ and $p(x)$ is more reasonable to describe the joint density as opposed to the pair $p(x|y)$ and $p(y)$. In causal settings as an example, one of these directions may be easier to specify or be such that the quality of decomposition may be invariant to new data[11].

In this section, we also mention various case-studies of data analysis in science and show that in every case, **constraints** enable effective analysis. The section is categorised as: **fixed projections** that create representations as user-defined functions of the data, **learned projections** that have been fit to predict metadata, and **density estimators** that create a vectorised representation of data-points in order to estimate their log-density. The former being an example of supervised learning, and the latter unsupervised or self-supervised. To conclude the chapter, we provide ideas that bridge projective and generative methods—specifically how classifiers implicitly act as generative models and equivalences that exist between generative and projective methods. We present these equivalences to highlight two major properties of probabilistic models, that inspires the aspects of our framework we introduce later in the thesis. The properties are that probabilistic models preserve their logical properties under transformations, and that models intended for a use-case may find themselves being applicable in an entirely different setting, highlighting their use.

2.3.1 Fixed projections

This subsection highlights the first of our three views on projective representations. As a basic form of representation generation and dimensionality reduction, **random projections** map

[11]For more on the topic, see the principle of Independent Causal Mechanisms, Schoelkopf et al. (2012).

high dimensional vectors $\mathbf{y} \in \mathbb{R}^d$ to representations $\mathbf{x} \in \mathbb{R}^{d_q}$ linearly, i.e.,

$$\mathbf{X} = \mathbf{YZ},$$

where $\mathbf{Z} \in \mathbb{R}^{d \times d_q}; \mathbf{Z}_{ij} \sim \mathcal{N}(0, (1/\sqrt{q})^2)$ is a randomly generated matrix. The Johnson-Lindenstrauss lemma (Johnson et al., 1984) states that such projections approximately preserve distance matrices in the lower dimensional space. Furthermore, due to the central limit theorem, such projections can also make the distribution of the data Gaussian-like under regularity assumptions, which may make them more applicable for downstream methods that make strong distributional assumptions.

Other common projections include fixed featurisation functions that convert inputs to real-representations that are interpretable or are known to preserve properties of the input that depend on the domain. For example, in a case study involving call identification of Titi-monkeys for conservation efforts, we show in appendix A.4 that there exist human-engineered auditory representations of speech spectrograms. These are low-dimensional and effective despite being engineered for human speech. Such representations allow for small but effective models to be developed for tasks such as activity detection (where the problem is, given a short sequence of audio, the task involves whether or not it corresponds to a call).

Another example of fixed projections are feature vectors describing non-conventional data, such as graphs. One way to go about featurising such data is to consider a vector that stores information as to whether or not certain substructures are present in the graph, and such transformations can in many cases be interpreted as feature functions (Nikolentzos et al., 2022). Such interpretations are useful from a computational point of view as feature functions are modular and can slot into software, for example, to construct kernels for Gaussian processes, which we show in appendix A.5.

Featurisers are not commonly fixed however, and are learned to predict metadata or data density, as we show below.

2.3.2 Projections that learn metadata

This subsection highlights the second of our three views on projective representations. Many methods of representation learning involve a function acting on the data $f(\mathbf{y})$, which is learnt to predict the metadata related to a sample point, for example, a classifier using image data to

predict data-point type. Assume that f is set up in such a way that $f(\mathbf{y}) := \sigma(\mathbf{w}^T \mathbf{r}(\mathbf{y}))$, where $\mathbf{r}$ is a vector formed as part of the function f, and an optional non-linearity/inverse-link/activation function σ. The vector $\mathbf{r}(\mathbf{y})$ can then be used as a representation of the input $\mathbf{y}$. A reason to use representations "closer" to the linear predictor is if we expect that they are disentangled; however, for models in audio, Pasad et al. (2022) show that "deeper" layers of self-supervised learning models encode semantics while "shallower" layers encode signal-related information (acoustics). As an example of a trained projection model, BERT (Devlin et al., 2019) is trained to output probabilities corresponding to masked portions of inputs within text.

Constraints that lead to powerful representations within such function classes are highly studied, for example, the effect of constructing functions (or generative models) that are invariant or equivariant to classes of transformations reflecting real-world properties that the models are used to describe (Fuchs et al., 2020). Convolutions used for images and graphs are another example (Kipf and Welling, 2017; Krizhevsky et al., 2012), that constrain the search space by explicitly looking for learned filters useful for downstream processing of images.

Sometimes, there is no metadata available with a sample point, and unsupervised or self-supervised methods can be used to learn a featuriser used within density-estimation contexts, as we discuss next.

2.3.3 Projections that learn densities

Lastly, as an alternative to learning relationships between aspects of the data, fixed featurising functions f can be optimised instead to output the (log) probability density of observing a data point $\mathbf{y}$ in high dimensional space. **Noise contrastive estimation** (NCE) of Gutmann and Hyvärinen (2010) is an example of a framework for learning functions that estimate log densities of input vectors directly as a function of the data $f(\mathbf{y})$. With NCE, one transforms density estimation into a supervised learning problem by training a classifier to distinguish between samples from the data distribution and samples drawn from a known "noise" distribution. Assume that the data is sampled from a density $p_{\mathbf{x}}$, and that we seek to estimate it using an unnormalised density function $p^u_{\boldsymbol{\alpha}}$ with parameters $\theta = \{\boldsymbol{\alpha}, z\}$, where $\log \int p^u_{\boldsymbol{\alpha}}(\mathbf{v}) d\mathbf{v} = Z(\alpha)$, of which z will be an estimate. Density estimation with NCE is done by sampling $\mathbf{x}$ using the empirical data distribution, as well as a known noise distribution p_n. We treat the number of positive and negative samples as equal for ease of exposition, although this number can vary

in practice and constitutes a hyperparameter. In NCE, we then train a classifier to distinguish between the data and noise samples using the model,

$$\mathcal{I}(\mathbf{v} \sim p_{\mathbf{x}})|\mathbf{v}, \boldsymbol{\alpha}, z \sim \text{Bernoulli}(\sigma(\log p^u_{\boldsymbol{\alpha}}(\mathbf{v}) - z - \log p_n(\mathbf{v}))).$$

Gutmann and Hyvärinen (2010) show that the optimisation of this model's likelihood results in the recovery of the true parameters that describe the data log-density, under class and identifiability conditions. NCE forms the basis of many self-supervised learning methods, and many functions within such models, being density estimators or approximations thereof, can be used to **measure the out-of-domainness of input examples**. This contrasts with the intended use of some of these methods, which are motivated as models that construct representations as opposed to providing literal log-densities of inputs.

Explicitly, probabilistic classifiers *can* be more uncertain about outputs corresponding to out-of-domain inputs, thereby offering a way to interrogate out-of-domain-ness (when calibrated). This seems to be true for both general classifiers as well as explicit noise contrastive models. In appendix A.6 and appendix A.7 we show that contrastive model and large language model uncertainties can be used as proxies for out-of-domainness estimation, for speech and protein property prediction respectively, in line with literature. In these contexts however, we found that performing Monte-Carlo dropout (Gal and Ghahramani, 2016) leads to better performance even when the models were not trained with dropouts.

This concludes our presentation of projective methods, as fixed featurisers based on domain-knowledge, learned functions of metadata or data densities. Before we present the generative models however, we reiterate that the split introduced is not a rigid one and there are connections between generative and projective methods. Before concluding the section, we show one such connection.

2.3.4 Connecting projective and generative models

So far, we have seen projective methods corresponding to fixed and learned functions predicting metadata or data densities. Before our exposition of generative models, in this subsection, we briefly present an example highlighting that projections can act as posteriors of generative models, thereby bridging the two views.

Example 2.1 (LDA)**.** A classical example follows, linear discriminant analysis (LDA). Assume

a generative model for the data as a mixture of normal distributions that share a covariance (with the different clusters corresponding to different classes). Concretely, LDA assumes a generative model for a specific data point,

$$c \sim \text{Categorical}(\boldsymbol{\pi}) \text{ and } \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_c, \Sigma).$$

This describes the joint distribution $p(\mathbf{x}|c)p(c)$. The posterior over which class a data-point corresponds (i.e. the distribution $p(c|\mathbf{x})$) is a logistic regression (Murphy, 2022). Assuming two classes, $C = 2$, the posterior probability is calculated as,

$$\begin{aligned} p(c|\mathbf{x}) &= \frac{p(\mathbf{x}|c)p(c)}{\sum_i^C p(\mathbf{x}|i)p(i)} \\ &= \frac{1}{1 + p(\mathbf{x}|\not{c})p(\not{c})/p(\mathbf{x}|c)p(c)} \\ &= \sigma(\log p(\mathbf{x}|c)p(c) - \log p(\mathbf{x}|\not{c})p(\not{c})) := \sigma(\delta_c(\mathbf{x})); \end{aligned}$$

this matches the form of a logistic classifier, as the form of classification boundary is,

$$\begin{aligned} \implies \delta_c(\mathbf{x}) &= -0.5(\mathbf{x} - \boldsymbol{\mu}_c)^T\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) + 0.5(\mathbf{x} - \boldsymbol{\mu}_{c'})^T\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_{c'}) + (\log \pi_c - \log \pi_{c'}) \\ &= (\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c'})^T\Sigma^{-1}\mathbf{x} + (-0.5(\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c'})^T\Sigma^{-1}(\boldsymbol{\mu}_c + \boldsymbol{\mu}_{c'}) + \log \pi_c - \log \pi_{c'}), \end{aligned}$$

which is linear in $\mathbf{x}$.

LDA also pops up in connection to other widely-used algorithms too, for example, Otsu thresholding (Otsu, 1979) used for image segmentation shares a connection to k-means clustering (Liu and Yu, 2009) and therefore the generative model used in LDA. Appendix A.9 shows an example of the method for cell-background segmentation.

The idea that classifiers store information about the data's distribution is studied widely, for example, in Grathwohl et al. (2020). Another example of a connection between classifiers and generative models can be found in the context of density ratio estimation, when classifiers learn to predict between samples drawn from a true-data distribution and another. In such cases, the optimal classifier learns to estimate a log density ratio between the distribution of the true data, and that of the second distribution. NCE (Gutmann and Hyvärinen, 2010) and other density estimators (Sec. 14.2.4 of Hastie et al. (2009)) make use of this fact, and such ideas can explain the behaviour of discriminative classifiers of GANs (Mohamed and Lakshminarayanan,

2017).

In appendix A.8, we show another such idea, that, assuming the generative model for LDA, the classifier confidence is a direct proxy to how far we are from the mode of the class, leading to an explanation for the connection between confidence and data-point log-likelihood.[12] Those results, however, do make the strong assumption that the assumptions of LDA are valid.

The relationship demonstrated is not universal. Generative models can assign higher likelihoods to out-of-domain examples as (at least one reason being that) statistical OOD-ness is different to semantic OOD-ness (Nalisnick et al., 2019). Discriminative models can also assign high confidence to out-of-distribution inputs, an effect seen commonly through adversarial examples.

Having seen how projective methods construct representations directly as functions of the data (via fixed featurisations and learned functions for metadata or densities), in section 2.4, we will switch to a generative view of representation learning. Generative methods will be models that work with an explicit probabilistic data-generating mechanism that explain observations using latent variables, making data-level assumptions transparent and enabling uncertainty quantification. This presentation will then culminate in the question, where should classical methods for dimensionality reduction sit? By considering this question later in the thesis, we will also show what probabilistic principles underpin the methods that follow.

2.4 Generative latent variable models

In this section, we briefly describe some generative latent variable models used in science to provide examples for the form of their construction. These will be Gaussian mixture models which use discrete latents, hidden Markov models that use discrete latents but have a notion of temporal order, linear factor models with continuous latents and their non-linear extensions in the form of GPLVMs. The linear and non-linear latent factor models in this section are similar in form to the framework we develop in the thesis. We also explore some of their properties, which will be useful as a point of comparison to the models described later in the thesis.

Latent variable models relate observations in terms of underlying, unobserved, and in some

[12]The intuitive idea is that the log probability decreases towards the class boundary/linear separator faster than it increases as one moves away from the separator. Thinking of an isotropic sphere representing the data log density for a class, we can reason that on a contour of equal log-density there is a sharper fall off in the density towards the boundary than away from it.

cases, interpretable variables. These models are central to representation learning, as they can be used for discovering compact and meaningful representations from large, complex datasets. As before, we also mention scientific case studies which involve constraining the models using domain-specific information.

2.4.1 Clustering using Gaussian mixture models

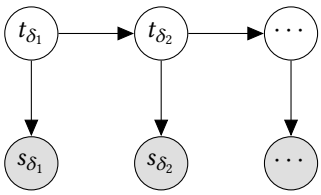
In this subsection, we describe the Gaussian mixture model with discrete latents, which performs *clustering* of data. The model is formulated as,

$$\begin{aligned} k &\sim \text{Categorical}(\boldsymbol{\pi}), \\ \mathbf{x}|k &\sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k), \end{aligned}$$

where data from unknown classes are assumed to follow a Gaussian distribution in the high-dimensional space. In the previous section, in example 2.1, we showed that posteriors of these models (under a shared covariance assumption) result in linear classifiers. These models are highly affected by model misspecification; in appendix A.10 we illustrate the results of Cai et al. (2021) who show that performing cluster enumeration when misspecifying the observation distribution (e.g. as a Gaussian when the data generating distribution is a Student-t) leads to an infinite number of clusters when the ground truth is finite. In appendix A.11, within a problem involving identification of cells with hypothesised irregular behaviour, we show how classification can be performed by the estimation of the Gaussian parameters through domain knowledge, and holding these fixed while performing one step of hard-assignment expectation maximisation. This is therefore another example of knowledge-driven constraints seen in science. Next, we show how such a model can be extended to temporal settings.

2.4.2 Latent time modelling using Hidden Markov models

A model akin to a GMM can be extended to account for temporal behaviour by imbuing the latent distribution with a distribution that evolves over time, leading to a hidden Markov model. They are a flexible model class used for a wide variety of applications, where the latents evolve according to a transition matrix that lists probabilities of moving from a state to another conditioned on the current state. The model graph is written as,



where observations s indexed by δ depend simply on the current state t_δ. For use-cases such as dynamic time-warping, we are interested in the discovery of latent "time" or an unseen temporal state (for example, where a stem cell is within its differentiation trajectory). Such models can be constrained by using upper-triangular banded transition matrices to ensure monotonicity of the latent time, thus imposing a hard constraint, necessary for the recovery of an interpretable latent variable. In appendix A.12, we show such an application within environmental science contexts, for dating ice-cores, and argue that probabilistic programming languages can be used to automate inference in such settings.

Turner and Sahani (2007) show that HMM-like models (linear dynamical systems, that use a continuous state space for their latents but follow a similar graph to the above) are a result of a probabilistic interpretation of a temporal dimensionality reduction method—slow-feature analysis (SFA). SFA uses the eigendecomposition of a covariance matrix derived from a "derivative" matrix $\dot{\mathbf{Y}} \approx \mathbf{Y}_{2:n} - \mathbf{Y}_{1:n-1}$ to rank features by small time derivatives. The probabilistic interpretation involves a latent variable with a temporally autoregressive AR(1) prior, and is related to the observed data in a linear manner.

Next, we explore non-temporal latent variable models, with continuous latent variables that have wide usefulness due to the relative ease of working with continuous latent variables.

2.4.3 Linear factor models

We now switch our attention to models for independent and identically distributed data that allow for continuous latents, which will be the focus of this thesis. Ideas from the next two subsections on how such models are constructed will appear frequently throughout the thesis. A number of algorithms, such as principal components analysis (**PCA**), factor analysis, Gaussian mixtures, non-negative matrix factorisation, latent Dirichlet allocation and independent components analysis (**ICA**) are known to have probabilistic interpretations or are formulated as probabilistic models (Murphy, 2023), wherein the generative model for n

independent high (d-)dimensional data points $\mathbf{Y} \equiv \begin{bmatrix} \mathbf{Y}_{1:} & \dots & \mathbf{Y}_{n:} \end{bmatrix}^T \in \mathbb{R}^{n \times d}$ is,

$$\begin{aligned} \mathbf{X}_{i:} &\sim p(.), \\ \mathbf{Y}_{i:} | \mathbf{X}_{i:} &\sim \text{ExponentialFamily}(g(\mathbf{X}_{i:}, \dots)) \end{aligned}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d_q}$ is a matrix-valued random variable of corresponding (typically low d_q-dimensional) latent variables. The inference process can be full-form (i.e. unamortised; as the posterior does not always factorise by data point) and inference can occur for the full matrix $\mathbf{X}$. Vanilla Gaussian process latent variable models (**GPLVMs**, Lawrence (2005)), generative topographic maps (GTMs, typically with a discrete latent grid; Bishop et al. (1998)) and variational models **VAEs** (Kingma and Welling, 2014) are also designed with such generative models, mapping the latents to the data distribution's parameters. In these models, the map f is described using a Gaussian process and a neural network respectively, rather than a linear function. In the case of VAEs, inference is not full form by design—the variational posterior is parameterised in these cases with a neural network, and factorises by data point, as the true posterior does.

The foundational linear latent variable models assume a linear relationship between data $\mathbf{Y}$, latent coordinates $\mathbf{X} \in \mathbb{R}^{n \times d_q}$ and a loading/factor matrix $\mathbf{W} \in \mathbb{R}^{d_q \times d}$,

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \boldsymbol{\epsilon}.$$

All three random variables on the RHS are unobserved, and there are known analytical solutions for some of the latents depending on which is marginalised. Probabilistic matrix factorisation (PMF, Mnih and Salakhutdinov (2007)) results in a (truncated) singular value decomposition (SVD) of the data describing the (rank-d_q) latents and the latent mapping,

$$\begin{aligned} \text{PMF:} \quad & \mathbf{Y} | \mathbf{X}, \mathbf{W} \sim \mathcal{MN}(\mathbf{X}\mathbf{W}, \sigma^2 \mathbf{I}_n, \mathbf{I}_d) \\ \implies & \widehat{\mathbf{X}\mathbf{W}} = (\mathbf{U}\mathbf{S})(\mathbf{V}^T) \qquad\qquad \text{SVD}(\mathbf{Y}), \end{aligned} \tag{2.2}$$

while probabilistic PCA of Tipping and Bishop (1999) involves the marginalisation of latent

coordinates,

$$
\begin{aligned}
&eq.\ (2.2) + \mathbf{X}_{ij} \sim \mathcal{N}(0,1):\\
&\text{PPCA:}\quad \mathbf{Y}|\mathbf{W} \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}_n, \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}_d) \\
&\qquad \Rightarrow \hat{\mathbf{W}}^T = \mathbf{U}(\mathbf{\Lambda} - \sigma^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}, \qquad\qquad \text{eigh}(\mathbf{Y}^T\mathbf{Y}/n)
\end{aligned}
\tag{2.3}
$$

and finally, dual-probabilistic PCA of Lawrence (2005), equivalently principal coordinates analysis, marginalises the linear map,

$$
\begin{aligned}
&eq.\ (2.2) + \mathbf{W}_{ij} \sim \mathcal{N}(0,1):\\
&\text{dual PPCA:}\quad \mathbf{Y}|\mathbf{X} \sim \mathcal{MN}(\mathbf{0}, \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}_n, \mathbf{I}_d) \\
&\qquad \Rightarrow \hat{\mathbf{X}} = \mathbf{U}(\mathbf{\Lambda} - \sigma^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}. \qquad\qquad \text{eigh}(\mathbf{Y}\mathbf{Y}^T/d)
\end{aligned}
\tag{2.4}
$$

When dealing with problems such as blind source separation when we need to "un-mix" signals from different sources (e.g. a speaker and background noise), a very similar model is specified—the linear model above with the noise level set to zero,

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^{-1} \iff \mathbf{X} = \mathbf{Y}\mathbf{W},$$

where $\mathbf{W}$ is invertible. Independent component analysis is an algorithm used to construct a matrix $\mathbf{W}$ that leads to "independent" sources in such a scenario. Hyvärinen and Oja (2000) show that it is an algorithm that can result from a number of equivalent views, including maximum likelihood estimation assuming a prior P_x on $\mathbf{X} \in \mathbb{R}^{n \times d_q}$ with $d_q = d$ for invertibility,

$$\forall i, j : \mathbf{X}_{ij} \sim P_x.$$

In words, probabilistic ICA makes the assumption that the latent sources $\mathbf{X}$ are independent and non-Gaussian (which is needed for identifiability). As we assume invertibility, specifying a prior is equivalent to specifying a generative model for $\mathbf{Y}|\mathbf{X}$. MLE within this model class

follows that each point $\mathbf{X}_{ij}$ follows a known distribution P_x with density f, hence,

$$\begin{aligned}
\hat{\mathbf{W}} &= \arg\max_{\mathbf{W}} \sum_{i}^{n} \log p_{\mathbf{Y}_i}(\mathbf{Y}_i|\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} \sum_{i}^{n} \log p_{\mathbf{X}_i}(\mathbf{W}^T\mathbf{Y}_i) + n \log|\det \mathbf{W}| \quad \text{change of variables} \\
&= \arg\max_{\mathbf{W}} \sum_{ij} \log f(\mathbf{W}_j^T\mathbf{Y}_i) + n \log|\det \mathbf{W}|, \quad \text{Pham and Garat (1997); Murphy (2023)}
\end{aligned}$$

which is the traditional ICA objective that is a function of the elements of the matrix $\mathbf{X}$ element-wise. Along with BSS, where interpretable latents are recovered, van Hateren and van der Schaaf (1998) found that interpretable maps (Gabor-like filters) are discovered when ICA is used with cell images.

Canonical correlation analysis (CCA) follows a similar generative model as the other linear factor models above, but with two datasets instead of one, following a model $\mathbf{Y}_a \leftarrow \mathbf{X} \rightarrow \mathbf{Y}_b$. CCA was originally defined to retrieve linear factors $\mathbf{w}_a, \mathbf{w}_b$ such that $\text{Cor}(\mathbf{w}_a^T\mathbf{Y}_a, \mathbf{w}_b^T\mathbf{Y}_b)$ is maximised (Hardoon et al., 2004), and this objective is a log-likelihood corresponding to the model graph above, with Gaussian conditionals Bach and Jordan (2005); Murphy (2023).

Many latent variable models presented thus far are used as dimensionality reduction models. In the case of dual-probabilistic PCA, as the latents are organised by decreasing variance, we are able to select only a few components that encode a large amount of variation of the data, thus describing the dataset with components with smaller dimensionality. In the case of ICA, if the latent factors are known not to correspond to sources of importance but to noise, then these sources are typically discarded, reducing the dimensionality of the data while reducing the amount of noise in the data. Of course, assuming that $\mathbf{X}$ is low dimensional in the first place leads to a dimensionality reduction, with $\mathbf{X}$ encoding properties of the data that depend on the model.

This concludes our presentation of models with discrete latents (GMMs, HMMs) and continuous linear latent variable models. Before concluding the section, we show how linear latent variable models are extended non-linearly, as many of our interpretations of t-SNE-like algorithms will have a very similar form as the upcoming models.

2.4.4 Gaussian process latent variable models

In the previous sections, we explored models where latents are related to observed outputs in a linear manner. On many occasions however, we expect causal factors to influence the data in a non-linear way. An example is found in the context of grid cells—certain brain cells called grid cells are known to have spatially tessellating firing fields, visualised in fig. 2.2 (Hafting et al., 2005). The activity spikes therefore show a non-linear transform of a causal variable, in this case location.

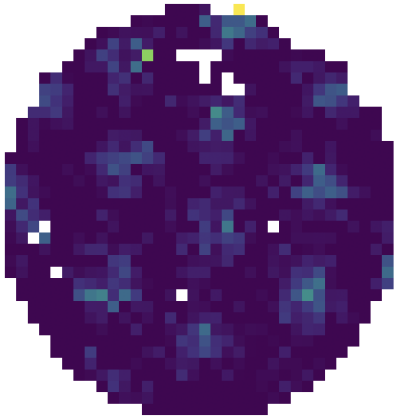


Figure 2.2: A spatial firing field measuring the number of firings of a rodent grid cell, plotted as a function of the rodent's location in space, showing that the firing field is spatially tessellated (the firing rates are a sinusoidal function of the location of the rodent in the circular container). Data from Hafting et al. (2005). This motivates non-linear latent factor models such as the GPLVM, as causal factors influence observations non-linearly.

Another example is that a latent corresponding to cell-division cycle can be related to gene expression in a cyclical manner.[13] To perform inference for such cell states, a strong constraint on the observation model is necessary to constrain latents enough such that interpretable latents can be recovered.

Observation models of the last section can be extended to discover such underlying latent variables by noticing that linear dimensionality reduction models use a **Gaussian process** with a linear covariance function (Lawrence, 2005).

Gaussian processes can be seen to represent distributions over functions (Rasmussen and Williams, 2005). For instance, consider a set of points chosen arbitrarily from an index set $\mathbf{X}$

[13]There are other situations where the behaviour of eigenvectors of empirical covariances is known to be smoothly varying across a variable of interest (yet another example is the yield curve, where an SVD of the dataset of bond rates by tenor yields the term structure).

distributed as,

$$\begin{bmatrix} y_a \\ y_b \end{bmatrix} \Big| \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & \exp(-\|\mathbf{x}_a - \mathbf{x}_b\|^2) \\ \exp(-\|\mathbf{x}_a - \mathbf{x}_b\|^2) & 1 \end{bmatrix}\right).$$

Samples of y can be seen as a finite set of points on a smooth line (which is enforced by the choice of covariance). If the covariance is changed to one that's periodic, e.g. $\text{cov}(y_a, y_b) = \exp(-\sin^2(\|\mathbf{x}_a - \mathbf{x}_b\|))$, samples y can be seen to be samples from a signal that's cyclical. The constructions are typically abstracted away, and we denote $\mathcal{GP}(0, k_{\text{smooth}}(\cdot,\cdot))$ or $\mathcal{GP}(0, k_{\text{periodic}}(\cdot,\cdot))$ to represent distributions over smooth/periodic functions[14]. Samples of paths from such GPs are illustrated below in fig. 2.3.

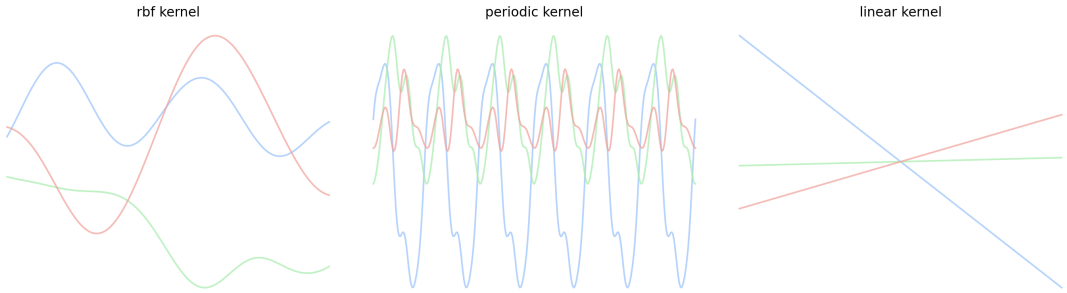


Figure 2.3: Samples obtained by smooth, periodic and linear kernels.

Replacing the linear kernel of the classical linear LVMs with a non-linear kernel leads to an extension known as the Gaussian process latent variable model (GPLVM, Lawrence (2005)),

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{MN}(0, K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n, \mathbf{I}_d), \quad \mathbf{X} \sim \ldots$$

We can recover features that are related to the output in a non-linear manner as $K(\mathbf{X},\mathbf{X}) \approx \Phi(\mathbf{X})\Lambda\Phi(\mathbf{X})^T$, where $\Phi(\mathbf{X})$ corresponds to the matrix of dominant eigenfunctions of the kernel evaluated at $\mathbf{X}$. The parametric form of the models in the previous section can suggest efficient inference ideas in such model classes, as demonstrated below by the parametric form of GPLVM.

Example 2.2. Parametric GPLVM: Kernels can be decomposed using random (Fourier) features (Rahimi and Recht, 2007) where $\mathbf{K}(\mathbf{X},\mathbf{X}) \approx \mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}(\mathbf{X})^T$, and we note that a matrix factorisation-type model,

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{MN}\left(\mathbf{\Phi}(\mathbf{X})\mathbf{W}, \sigma^2\mathbf{I}_n, \mathbf{I}_d\right),$$

[14]Posterior means of such processes, however, lie on different function spaces than their samples (Kanagawa et al., 2018).

with weights marginalised leads to the GPLVM. Therefore, the model above can be seen as a parametric precursor of the GPLVM, and it can be used to perform mini-batch parametric inference with stochastic gradient descent, due to factorisation of the log likelihood by data point (i.e. conditional independence of the embeddings given $\mathbf{W}$). Figure 2.4 shows MNIST digits clustered using the model, showing that clustering by digit is achieved, albeit with some amount of collapse into strands. A longer exploration of the use of RFFs with GPLVM in a fully

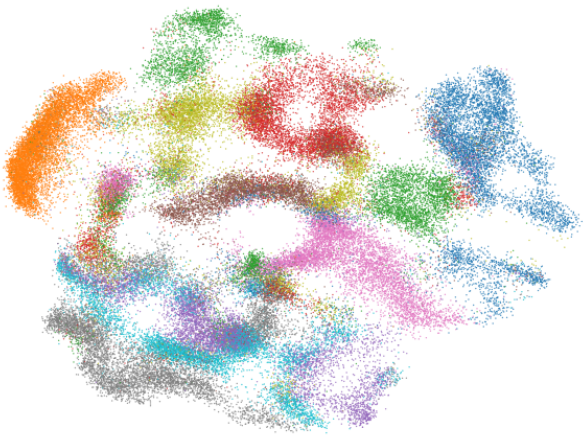


Figure 2.4: Embeddings generated using the parametric GPLVM model on the MNIST dataset (using mini-batch SGD for optimisation), showing clustering by digit, albeit with some collapse of clusters into strands, illustrating the ability of the model for dimensionality reduction.

Bayesian setting is presented in Gundersen et al. (2020). Similar ideas will be used with our interpretations of t-SNE-like algorithms to enable efficient inference through mini-batching.

Gaussian processes constrain the functional form between latents and data. In other models, such as VAEs for images, modelling constraints can similarly yield results. For example, Dorta et al. (2018) show that, a Gauss-Markov random field with adjacencies across pixels or areas of an image makes for a good generative model. Within generative models for proteins, representations are also commonly seen to respect physical symmetries, for example, in alphafold (Jumper et al., 2021).

GPLVMs are computationally challenging due to the calculation of the log-determinant and inverse of the kernel matrices needing to be computed at every iteration of optimisation. Following the ideas of Hensman et al. (2013), sparse-GPs can be used to construct GPLVMs that do not incur this cost. Appendix A.13 shows how such methods can be used within single-cell RNA-seq contexts; the main lessons are summarised as follows. Firstly, expert-driven or science-driven initialisations are necessary for interpretability. Secondly, "pre-training" other

hyperparameters of GPLVMs, when the latent initialisations are meaningful, is necessary to retain the inductive bias from initialisations. Thirdly, pseudo-inputs to the sparse-GPs can be constrained such that the resultant functions are additive, and finally that normalising the data, so that $\forall i : \sum_k \mathbf{Y}_{ik} = 10^5$, leads to vastly improved performance. These lessons reiterate that constraints are common in scientific modelling.

Before we conclude this section, we show that GPLVM-like models, which are similar to our models in chapter 3, can be resistant to likelihood-misspecification.

Proposition 2.1. Below, we show that GPLVMs which are used often in single-cell data analysis, are resistant to misspecification when the data is binary contrary to model assumptions.

Consider a basic model for data such as gene-expression data (which is typically zero-inflated) that is binarised, formulated as a latent Gaussian process model affected by thresholding,

$$\begin{aligned} \mathbf{z}|\mathbf{X} &\sim \mathcal{N}(\mathbf{0}, \Sigma(\mathbf{X})) \\ \forall i : y_i &= \begin{cases} \mathcal{I}(z_i > 0) & \text{w.p. } 1 - p_z \\ 0 & \text{w.p. } p_z \end{cases}. \end{aligned}$$

Then,

$$\begin{aligned} Cov(y_i, y_j) &= \mathbb{E}(y_i y_j) - \mathbb{E}(y_i)\mathbb{E}(y_j) \\ &= (1 - p_z)^2 \mathbb{P}(z_i, z_j > 0) - \frac{1}{4}(1 - p_z)^2 \\ &= \frac{(1 - p_z)^2}{2\pi} \arcsin(\hat{\Sigma}_{ij}) && \text{Gaussian orthant (Tong, 1990)} \\ &\approx \frac{(1 - p_z)^2}{2\pi} \hat{\Sigma}_{ij}, && 1^{\text{st}} \text{ order Taylor} \end{aligned}$$

where $\hat{\Sigma}_{ij} = \text{Cor}(z_i, z_j)$. This shows that using GPLVMs with binary data does not lead to a misspecification in terms of the covariance choice, as this is somewhat preserved (at least, the resulting covariance is monotone and nearly-linear in the assumed data-generating covariance). Moreover, if this is the generating process, the estimate of empirical covariance even in the presence of binarisation will lead to a sensible estimate due to the central limit theorem. The covariance arises as a sufficient statistic in a large class of models, and therefore, if its estimate is unharmed by misspecification, downstream inference results are saved, up to

their interpretation (as the interpretation of a covariance between binary random variables is different to that obtained between real-valued response variables).

We highlight this property for two reasons. Firstly, ProbDR models in chapter 3 are similar in form to GPLVMs, and therefore, some lessons can be transferred to those cases. Secondly, we will argue that GPLVM is a more natural specification (after adequate constraining) for general use-cases, and therefore, we are interested in its robustness.

This concludes our exposition of generative models. We have shown how they are constructed as models that use latents to explain aspects of the data, and provided examples of linear latent variable models, their non-linear extensions in the form of GPLVMs, and the interpretations of the corresponding inference algorithms as variance-maximising or source-separating algorithms. In the next section, we describe dimensionality reduction algorithms without known probabilistic interpretations, algorithms which will seem different to those presented thus far.

2.5 Dimensionality reduction and neighbour embedding algorithms

Before we conclude the chapter, we provide a brief exposition of classical dimensionality reduction methods (without existing probabilistic interpretations), to contrast them to the probabilistic methods presented thus far, which obtain (typically low-dimensional) vector representations of data. These do not have a trivial positioning within our taxonomy of models, and these algorithms will be the focus of our effort for the rest of the thesis.

Many of these algorithms use associations known between points to cluster them together in the latent space, thereby preserving the graph structure that implicitly or explicitly underpins the data. We organise the algorithms as follows. The first and second categories are algorithms that work with high-dimensional data points and act as dimensionality-reduction algorithms. Algorithms of the first category use an eigendecomposition of a matrix of data-data similarities to form a low-dimensional representation. Those of the second category create a graph of similarities using a (sometimes implicit) nearest neighbour algorithm and then match the distances in a latent graph to the distances implied by the data graph, using a graph matching objective that does not simply have a squared error form (which the eigendecomposition cases

correspond to). The third category we describe are relational embeddings, where one has access to pairs or triplets of points that are "similar", for example, words that are close together in text, or a triplet of words such that a combination of two words conveys a similar meaning to a third, e.g. "woman" and "ruler" conveying "queen". Another possibility is data points that can be related to others when the latter are defined by similarity-preserving transformations of the former, e.g. horizontal flips of natural-context images.

We describe dimensionality reduction as a **data analytical goal**—the majority of our interpretations for the DR algorithms below will be latent variable models that achieve the goal of dimensionality reduction. A "dimensionality reduction model" in this thesis is a latent variable model whose primary use is dimensionality reduction but can be used for other use-cases. As an example, dual-probabilistic PCA may be used for DR if only some latents that explain the highest variation in the data are kept, but it may also be used for rebalancing the variation in the data so that useful low-variance dimensions *have a larger say* (PCA-whitening is an example of this process).

With this in mind, we first describe dimensionality reduction methods before turning our attention briefly to relational embedding methods and ideas therein.

2.5.1 Dimensionality reduction algorithms

The first of our two categories, these algorithms, in their simplest formulation, aim to find embeddings $\mathbf{x} \in \mathbb{R}^{d_q}$ corresponding to data-points $\mathbf{y} \in \mathbb{R}^{d}$ with $d_q << d$.

In the field of single-cell transcriptomics (scRNA-seq) as an example, one measures the number of RNA molecules that are present in droplets of cytoplasm from single cells, that give us an idea of what genes are being expressed in the cell (Haque et al., 2017). This forms the high-dimensional dataset. It is hypothesised that there are typically a low number of causal factors that determine such high-dimensional observations, hence, a dimensionality reduction step is a central part of many pipelines to identify such factors (Luecken and Theis, 2019). For these reasons, it is said that these datasets lie near "low-dimensional manifolds", albeit with a lot of "noise" due to both observational/experimental reasons, and uncertainty from unobserved variables (representing aspects of aleatoric and epistemic uncertainty respectively).

Although disentangled and causally meaningful low-dimensional representations are the most desired outputs of dimensionality reduction methods, these can be difficult to find, and

often can only be achieved through strong scientific or modelling assumptions (constraints). Nonetheless, low-dimensional representations are still useful for downstream processing as they are generally more tractable (from computational and interpretability standpoints, but also as downstream models may be more robust to model misspecification with lower-dimensional inputs) and information dense when compared to their high-dimensional counterparts.

Many dimensionality reduction algorithms are used within single-cell data analysis pipelines (and beyond), but they lack explicit modelling semantics and there is no consensus as to what they truly *do*. A probabilistic framework would serve to explain such algorithms, as we show in chapter 3, and perhaps form the basis for future study of the methods.

We now describe our two views of DR algorithms: ones that use eigendecompositions of PSD matrices, and others that optimise a loss function based on a (sometimes implicit) nearest-neighbour graph. These will specifically be SNE, t-SNE and UMAP.

2.5.1.1 Eigendecomposition-based methods

A large class of widely-used DR methods involve obtaining a low dimensional representation of the data as the eigenvectors of a covariance-like matrix or a graph Laplacian matrix. As an example, Laplacian Eigenmaps of Belkin and Niyogi (2001) constructs a graph Laplacian matrix from a k-NN graph, and constructs a low-dimensional embedding as the non-trivial eigenvectors corresponding to the smallest eigenvalues of this matrix. Other examples of such methods include the dual formulation of PCA, introduced in the previous section. They are described in more detail in section 3.2.2 along with their probabilistic interpretations. In the coming sections, we detail the other class of DR algorithms which use a probabilistic graph matching objective to construct a low-dimensional embedding.

2.5.1.2 Stochastic neighbour embedding

The stochastic neighbour embedding (SNE) algorithm was introduced by Hinton and Roweis (2002) as an approach for dimensionality reduction. The approach was to minimise a KL divergence between a set of probabilities v_{ij}^S (corresponding to two data points i and j being neighbours) generated by a discrete distribution in a data space $\mathbf{Y}$ and a discrete distribution with probabilities w_{ij}^S generated by using a lower dimensional latent embedding $\mathbf{X}$. These

probabilities are defined as,

$$v_{ij}^S = \frac{\exp(-\|\mathbf{Y}_{i:} - \mathbf{Y}_{j:}\|^2/\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{Y}_{i:} - \mathbf{Y}_{k:}\|^2/\sigma_i^2)},$$
$$w_{ij}^S = \frac{\exp(-\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|^2)}{\sum_{k\neq i}\exp(-\|\mathbf{X}_{i:} - \mathbf{X}_{k:}\|^2)},$$

where $\mathbf{Y}_{i:}$ denotes the i^{th} row of $\mathbf{Y}$, and σ_i is a hyperparameter that is found by information-theoretic arguments. Probabilities w_{ij}^S are made close to probabilities v_{ij}^S by minimising the objective below with respect to $\mathbf{X}$,

$$\mathcal{L}_{SNE} = \sum_i \sum_{j\neq i} v_{ij}^S \log \frac{v_{ij}^S}{w_{ij}^S}.$$

The idea is that if probabilities defined in latent space are similar in terms of the KL divergence to probabilities defined in data space, then the latent dimensions of $\mathbf{X}$ are capturing some salient aspect of the data $\mathbf{Y}$. In all three algorithms, probabilities of association relating to the same point, v_{ii} and w_{ii}, are set to zero. This algorithm was succeeded by t-SNE, which follows.

2.5.1.3 t-distributed stochastic neighbour embedding

The t-SNE algorithm was introduced by van der Maaten and Hinton (2008) to improve optimisation and visualisation with respect to SNE. In the t-SNE algorithm, probabilities v_{ij}^t and w_{ij}^t are defined as,

$$v_{ij}^t = (v_{ij}^S + v_{ji}^S)/2n,$$
$$w_{ij}^t = \frac{(1 + \|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|^2)^{-1}}{\sum_{k\neq l}(1 + \|\mathbf{X}_{k:} - \mathbf{X}_{l:}\|^2)^{-1}},$$

which are then matched by minimising the cost function below with respect to $\mathbf{X}$,

$$\mathcal{L}_{t-SNE} = \sum_{i\neq j} v_{ij}^t \log \frac{v_{ij}^t}{w_{ij}^t}.$$

The normalization here, as opposed to SNE, is over the entire set of probabilities. Next, we review the UMAP algorithm.

2.5.1.4 Uniform manifold approximation and projection

The UMAP algorithm (McInnes et al., 2020) is used extensively in computational biology for visualising single-cell RNA-seq data due to decreased runtimes and a greater ability of recovering cell clusters as compared with t-SNE (Becht et al., 2019). The algorithm defines probabilities v^U_{ij} and w^U_{ij} as

$$\begin{aligned} v^U_{j|i} &= \exp((\rho_i - \text{distance}(\mathbf{Y}_{i:}, \mathbf{Y}_{j:}))/\sigma_i), \\ v^U_{ij} &= v^U_{i|j} + v^U_{j|i} - v^U_{i|j} * v^U_{j|i}, \\ w^U_{ij} &= (1 + a\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|^{2b})^{-1}, \end{aligned}$$

where ρ_i denotes the distance to the nearest neighbour of data point i. We match these by optimising the following cost function with respect to $\mathbf{X}$,

$$\mathcal{L}_{UMAP} = \sum_{i \neq j} v^U_{ij} \log \frac{v^U_{ij}}{w^U_{ij}} + (1 - v^U_{ij}) \log \frac{1 - v^U_{ij}}{1 - w^U_{ij}},$$

i.e. a cross-entropy type loss matching a latent graph to one created by using the high-dimensional data points.

We can also use, for example, a **Poincaré metric** or in general hyperbolic distance metrics within such algorithm (or model) constructions, instead of the Euclidean metric. Hyperbolic surfaces are well-suited to representing trees with low distortion, and so these geometries form natural real-valued spaces within which tree-structured data can be embedded to expose underlying similarity behaviour (Nickel and Kiela, 2017). Similarly, data embedded into hyperspheres can recover periodic/cyclical behaviour.

In the next subsection, we show similar algorithms to t-SNE and UMAP that are not strictly dimensionality reduction algorithms, but have a very similar form. We will use ideas from these methods in chapter 3, to show that approximate inference can be done in our framework that seeks to explain t-SNE-like algorithms.

2.5.2 Relational embeddings

This subsection presents the third set of algorithms, after eigendecompositions and t-SNE-like algorithms, that construct embeddings in contexts that are not quite as simple as reducing

the dimensionality of a dataset of i.i.d. points. They instead work with contexts where data points have temporal or relational connections, used widely in the context of word embeddings. Although we do not detail careful probabilistic interpretations of these algorithms in this thesis, they can be seen to be highly similar in construction to dimensionality reduction methods such as (t-)SNE and UMAP. Ideas relating to efficient inference within such settings are also reviewed briefly, as **they will be used to construct efficient inference options for our probabilistic interpretations**—thereby showing (in section 3.3) that within the realm of probabilistic models, insights from certain models can be translated easily to other cases. Relational embeddings learn representations directly from observed similarity relationships or co-occurrences.

Word2vec using SGNS: Given a word represented by an integer i and context j, skip-gram with negative sampling (SGNS) of Mikolov et al. (2013) maximises a logistic classifier's log-probability over edges vs sampled non-edges, with respect to latent embeddings $\mathbf{w}_i$ and $\mathbf{c}_j$,

$$\mathcal{E}_{ij} := \log \sigma(\mathbf{w}_i^T \mathbf{c}_j) + n_{\text{neg}} \mathbb{E}_{k \sim p_{\text{neg}}} \log \sigma(-\mathbf{w}_i^T \mathbf{c}_k).$$

The negative sampling was specifically inspired by noise contrastive estimation of Gutmann and Hyvärinen (2010). Levy and Goldberg (2014) show that optimising this objective is approximately equivalent to maximising the pointwise mutual information; in expectation, the stationarity condition gives $\mathbf{w}_i^T \mathbf{c}_j \approx \text{PMI}(w_i, c_j) - \log n_{\text{neg}}$ and hence, the argument follows that a singular value decomposition of the PMI matrix on the right-hand-side produces reasonable latents.

TransE: Given a triplet, $(\boldsymbol{u}, \boldsymbol{h}, \boldsymbol{v})$, TransE introduced by Bordes et al. (2013), minimises the objective,

$$\mathcal{L} = \sum_{\mathbf{u},\mathbf{h},\mathbf{v} \in S} \sum_{\mathbf{u}',\mathbf{v}' \sim p_{\text{neg}}} [\gamma + d(\boldsymbol{u} + \boldsymbol{h}, \boldsymbol{v}) - d(\boldsymbol{u}' + \boldsymbol{h}, \boldsymbol{v}')]_+$$

thereby learning embeddings such that, approximately, $\mathbf{u} + \mathbf{h} \approx \mathbf{v}$.

This concludes our presentation of algorithms that currently do not have interpretations as inference algorithms within probabilistic models. The methods presented in this section appear differently to probabilistic methods as there is no explicit construction of a model. We presented methods of dimensionality reduction through eigendecomposition and loss-minimisation on a graph. We then presented methods for obtaining word embeddings that follow similar ideas.

To recap the background, we have presented a review of probabilistic models and probabilistic models for representation learning, and in the next chapter, we formulate the probabilistic interpretations to the algorithms of this section, by showing how these objectives correspond to (lower bounds on) explicitly defined models' likelihoods. Our consideration of these ideas will lead to a framework that unifies ideas in scientific representation learning, to provide a framework with which one can compare, constrain and extend models based on use-case.

CHAPTER 3

ProbDR: A unifying framework for coordinate dimensionality reduction

The aim of the thesis is to show that explicit probabilistic models underpin methods of representation learning, and that often, they involve a minimal statistic of the data that limits degrees of freedom. We were motivated by a desire to find an interpretable framework that can enable comparison of the many methods used in scientific representation learning.

In the previous chapter, we provided a background on ideas in probabilistic representation learning to contextualise the field and explored real-world use-cases of latent variable models in science. We showed that constraints are commonly observed in scientific representation learning, that involve the usage of constrained estimators of known quantities. In this chapter, we focus on the large set of **dimensionality reduction** algorithms, reviewed briefly in the previous chapter (section 2.5), where we saw that they do not fall into the taxonomy of methods in probabilistic representation learning.

In this chapter, we provide the probabilistic interpretation for many dimensionality reduction methods used in science, thereby answering what they *do*: covariance estimation (or nearest neighbour prediction, which as we show, can be framed as covariance estimation). We also show the implicit assumptions that they make: they use specific linear/non-linear covariance kernels and use a non-standard covariance estimator. We show that the framework has many connections within its different views, and show that the various interpretations we present morph between one another, showing coherence of the models presented. The

following sections are all presented with the following roadmap in mind,

the interpretation → what are the model assumptions? → are the transforms coherent?

We will now summarise the major results of the chapter.

3.1 Overview of results

In this chapter, we unify prevalent dimensionality reduction methods, from a probabilistic perspective, draw comparisons to dual-probabilistic PCA and GPLVMs, and explain what such methods do. Through this process, we learn that GPLVMs are not as constrained as algorithms in the field, and learn what the ground truth underlying constraints that appear in scientific representation learning are. These constraints appear mainly through the construction and usage of atypical covariance estimators, and static variational constraints.

Concretely, we show that many dimensionality reduction algorithms are approximately inference algorithms within a specific modelling class, that we call **ProbDR**. Let $\hat{\mathbf{S}}$ be an estimated data covariance. Then, we write the model class as,

$$\gamma\hat{\mathbf{S}}|\mathbf{X} \sim (\text{Inv-})\mathcal{W}(\mathbf{X}\mathbf{X}^T + \beta\mathbf{H}\, K(\mathbf{X},\mathbf{X})\, \mathbf{H} + \kappa\mathbf{I}, \nu),$$

where $\mathbf{X}$ correspond to the latent variables (also known as coordinates), K is a Cauchy (rational quadratic) kernel matrix, and $\mathbf{H}$ is a centering matrix. All parameters apart from the latents $\mathbf{X}$ are known, and depend on the choice of algorithm being approximated. The primary difference between methods stems from the choice of estimator of $\hat{\mathbf{S}}$. Classical models also fit neatly into this model class, enabling comparability. This form sheds light into why many methods are similar: because they often estimate the same underlying statistics of the data, and assume similar models, with subtle differences arising from the choice of data covariance estimators or non-linear latent covariances. Although there is widespread effort in the community to understand these methods from "attraction" and "repulsion" terms used in the loss function, to the best of our knowledge, this is the first presentation of the methods studied in the thesis as maximum a-posteriori algorithms given a probabilistic model.

Our interpretation arises due to two main results, which are as follows. First, we observe that many DR algorithms, such as dual-probabilistic PCA, MDS, kernel-PCA, Laplacian Eigenmaps,

Locally Linear Embedding, and Isomap, output a low-dimensional representation by,

1. First estimating a PSD matrix which we interpret as a covariance $\hat{\mathbf{S}}$ or precision $\hat{\mathbf{\Gamma}}$ matrix. In PCA, for example, $\hat{\mathbf{S}}(\mathbf{Y}) = \mathbf{Y}\mathbf{Y}^T/d$, and in Laplacian Eigenmaps, $\hat{\mathbf{\Gamma}} = \mathbf{L}$ encodes a nearest neighbour adjacency matrix,

2. Then, setting the embedding $\mathbf{X}$ to the d_q scaled eigenvectors of the matrix corresponding to the largest or lowest eigenvalues (referred to as major & minor eigenvectors respectively).

The first result of the thesis arises due to a simple observation: that dual-probabilistic PCA involves an eigendecomposition. Therefore, methods that use an eigendecomposition to construct latent variables must be using simply a non-standard covariance estimator, as we show below.

Theorem 3.1. Assume a Wishart model for a covariance $\hat{\mathbf{S}}$ (or precision $\hat{\mathbf{\Gamma}}$) that uses a linear kernel (or its inverse, if a precision is being modelled) to describe the centrality parameter, using the latents $\mathbf{X}$. The MAP estimate of $\mathbf{X}$, with an improper uniform prior over $\mathbf{X}$, occurs at the d_q principal/major and minor scaled eigenvectors of $\hat{\mathbf{S}}$ & $\hat{\mathbf{\Gamma}}$ respectively,

$$\hat{\mathbf{S}} * \nu|\mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}_n, \nu\right) \quad \Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q\,\text{maj}}(\mathbf{\Lambda}_{d_q\,\text{maj}} - \hat{\sigma}^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T \tag{3.1}$$

$$\hat{\mathbf{\Gamma}} * \nu|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n)^{-1}, \nu\right) \quad \Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q\,\text{min}}(\mathbf{\Lambda}^{-1}_{d_q\,\text{min}} - \hat{\beta}\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T \tag{3.2}$$

where $\mathbf{U}_{d_q}, \mathbf{\Lambda}_{d_q}$ are matrices of d_q eigenvectors and corresponding eigenvalues, $\mathbf{R}$ is an arbitrary rotation matrix and ν are arbitrary degrees of freedom (i.e. their choice does not affect the embedding). $\hat{\mathbf{S}} = \mathbf{Y}\mathbf{Y}^T/d$ with $\nu = d$ recovers dual-probabilistic PCA.

Our second result, which interprets t-SNE-like algorithms in section 3.3, adds a non-linear component to the centrality parameter.

Theorem 3.2. UMAP and t-SNE-like algorithms correspond to MAP estimation of $\mathbf{X}$ assuming a Wishart distribution on a precision matrix, estimated by the graph Laplacian $\mathbf{L}$, that uses the inverse of a familiar non-linear kernel,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left(\left(\mathbf{X}\mathbf{X}^T + 0.5\mathbf{H}\mathbf{P}\mathbf{H} + \mathbf{I}/2\tilde{\epsilon}\right)^{-1}, n\right), \tag{3.3}$$

where $\tilde{\epsilon} \approx 4n_{\text{neg}}n_{\text{neigh}}/3n$ is defined at the outset by the choice of hyperparameters, $\mathbf{P}_{ij} = 1/(1 + \|\mathbf{X}_i - \mathbf{X}_j\|^2)$ is the Cauchy (Student-t or rational quadratic) kernel, and $\mathbf{H}$ is a centering matrix.

We will show that these interpretations are semantically consistent in accordance with the main claim of the thesis, that probabilistic models can correspond to semantic grammars over what they model. We show that the models are coherent even under transformations, and are the models that would be selected to model the statistic of interest. Efficient inference ideas in this framework will show that there is a correspondence between our linear and non-linear covariance interpretations.

The **what** that is modelled by many methods (the choice of covariance estimator), like many statistics estimated in the previous chapter, estimates a **particular characteristic** of the data. For example, we argue that a graph Laplacian is a "lossy" estimate of the data precision, and only some characteristics are retained. **What** exactly is retained, e.g. when a kNN graph is computed on data that is highly zero-inflated, as opposed to when it is computed using data that is more Gaussian-like, or when the data has extreme values, we leave to future research, but we point out exactly **how** these statistics are calculated for every algorithm.

In the last part of this chapter, we show that all algorithms studied also have variational interpretations. As an example, first represent the data covariance that is being modelled as $\mathbf{M}$. Then, MAP inference in our linear Wishart model (theorem 3.1, first case) is equivalent to finding $\arg\min_{\mathbf{X}} \mathrm{KL}(q(\mathbf{M})\|p(\mathbf{M}))$, assuming a model for the covariance,

$$p(\mathbf{M}|\mathbf{X}) = \mathcal{W}(\mathbf{M}|\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, \nu),$$

and a variational constraint for the covariance that uses the observed estimate of the covariance $\hat{\mathbf{S}}$ as the centrality parameter,

$$q(\mathbf{M}|\mathbf{Y}) = \mathcal{W}(\mathbf{M}|\hat{\mathbf{S}}(\mathbf{Y}), \nu).$$

The second case is similar, moreover, the t-SNE objective was originally given in van der Maaten and Hinton (2008) as a KL-divergence, also between a distribution that depends on the data with no learnable parameters and a distribution that depends on the latents (in that order, over

an adjacency matrix)[1].

A graphical summary of the ideas presented thus far is illustrated in fig. 3.1. The illustration shows that the likelihood interpretations, where the covariance is modelled by a Wishart using a covariance kernel, or a precision using the inverse of the kernel. This is followed by our variational framework, and finally, the specific choices of estimators that lead to the various algorithms is summarised.

The variational interpretations will be useful for the next chapter, where we use ideas presented in Yu et al. (2023); Nakamura et al. (2023); Hu et al. (2023) to understand transformers of Vaswani et al. (2017) as unrolled inference assuming our variational Laplacian Eigenmaps interpretation. We will show that by constructing this probabilistic connection, the performance of the architecture can be improved, highlighting the usefulness of our framework for a seemingly unrelated use-case in representation learning.

3.2 Likelihood interpretations: linear cases

In this section[2], we show how the first result of ProbDR arises, that MAP inference follows eigendecomposition assuming a Wishart latent variable model with a linear covariance kernel. We will also explain how many DR algorithms in practice fit into this framework. As part of enumerating the various connections, we show how the covariance-view and the precision-view recover the same solution when the traditional covariance estimator (corresponding to dual-probabilistic PCA) is used.

As summarised earlier, many DR algorithms compute embeddings in a two step process,

1. Estimate a PSD matrix which is interpreted as a covariance $\hat{\mathbf{S}}$ or precision $\hat{\mathbf{\Gamma}}$ matrix. This can be a straightforward function of the data, e.g. dual-probabilistic PCA, where $\hat{\mathbf{S}}(\mathbf{Y}) = \mathbf{Y}\mathbf{Y}^T/d$, or an estimator constructed in a hierarchical fashion, as in the case of LLE, which we present later in this chapter.

2. Set the latent embedding $\mathbf{X}$ to d_q scaled eigenvectors of the matrix above corresponding to

[1] In our work, all KL-divergences appear as $\mathrm{KL}(q||p)$. In the (t-)SNE papers, Hinton and Roweis (2002); van der Maaten and Hinton (2008) use notation $\mathrm{KL}(p||q)$; we flip this notation (and relabel the distributions) but otherwise keep calculations identical so that the functional objective remains unchanged. It is more natural, we argue, to denote the variational constraint that only acts on the data as q, and the model distribution as p, unlike how this is presented in the (t-)SNE papers.

[2] This section follows our results in **Ravuri et al. (2023)**.

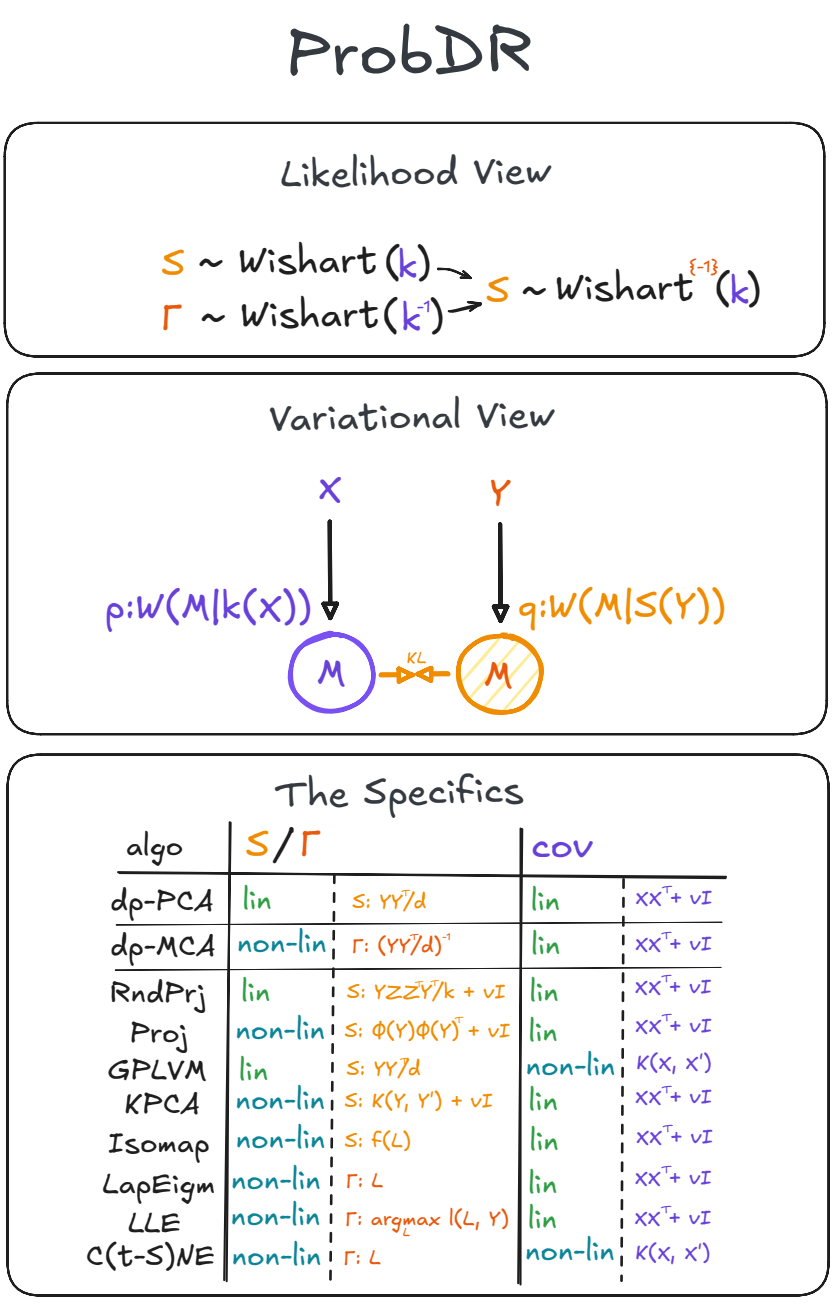


Figure 3.1: A graphical abstract summarising the main contributions of the chapter.
Top: A summary of the likelihood views of section 3.2 and section 3.3 showing that two main interpretations underpin ProbDR: a covariance $\mathbf{S}$ or **equivalently** a precision $\mathbf{\Gamma}$, is modelled by a covariance kernel k or its inverse respectively. The two views can instead be written in terms of a covariance $\mathbf{S}$, modelled by an either a Wishart or inverse-Wishart distribution with centrality matrix k.
Middle: The variational view of ProbDR showing that the likelihood interpretations have a KL-minimising view, where the model is defined on an arbitrary matrix $\mathbf{M}$ representing the covariance, with a variational constraint placed upon it. The constraint has the observed data matrix as the centrality parameter, and is a static constraint.
Bottom: The specific choices of $\mathbf{S}$ and $\mathbf{\Gamma}$ and k that lead to the various algorithms we consider.

the largest or lowest eigenvalues (referred to in this work as major and minor eigenvectors, respectively).

Firstly, we show that step 2 is MAP estimation for $\mathbf{X}$, in a quasi-maximum likelihood sense, meaning that we use an estimated statistic $\hat{\mathbf{S}}$ that may not be the standard maximum-likelihood estimator under a globally assumed model[3]. White (1982) showed that using a model family that is not the data-generating distribution for inference still results in a valid estimate, the **"quasi-maximum likelihood estimate"** in a KL sense; the QMLE minimises the KL-divergence between the true and assumed models.

Mathematically, inference for the latents given the observed statistic represented as the random variable $\mathbf{S}$, which is a function of the data $\mathbf{Y}$, follows,

$$\begin{aligned}\hat{\mathbf{X}} &= \arg\max_{\mathbf{X}} \log p_{\mathbf{X}}(\mathbf{X}|\mathbf{S}) \\ &= \arg\max_{\mathbf{X}} \log p_{\mathcal{M}}(\mathbf{S}|\mathbf{X}) \underbrace{p(\mathbf{X})}_{\propto 1} && \text{Bayes rule} \end{aligned} \tag{3.4}$$

We show the model that determines the model density $p_{\mathcal{M}}$ in theorem 3.1, restated below; the MAP estimate of $\mathbf{X}$, with an improper uniform prior over $\mathbf{X}$, given Wishart models with linear covariance kernels occurs at the d_q principal/major and minor scaled eigenvectors of the covariance estimate $\hat{\mathbf{S}}$ and the precision estimate $\hat{\mathbf{\Gamma}}$ respectively,

$$\begin{aligned}\hat{\mathbf{S}} * \nu|\mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}_n, \nu\right) \quad &\Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q\,\text{maj}}(\mathbf{\Lambda}_{d_q\,\text{maj}} - \hat{\sigma}^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T \\ \hat{\mathbf{\Gamma}} * \nu|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n)^{-1}, \nu\right) \quad &\Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q\,\text{min}}(\mathbf{\Lambda}^{-1}_{d_q\,\text{min}} - \hat{\beta}\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T\end{aligned}$$

where $\mathbf{U}_{d_q}, \mathbf{\Lambda}_{d_q}$ are matrices of d_q eigenvectors and corresponding eigenvalues, $\mathbf{R}$ is an arbitrary rotation matrix and ν are arbitrary degrees of freedom. $\hat{\mathbf{S}} = \mathbf{Y}\mathbf{Y}^T/d$ with $\nu = d$ recovers dual-probabilistic PCA. We will refer to the first of these results as the **covariance view** and the second as the **precision view**.

Therefore, we can interpret any DR algorithm first computing a PSD matrix, and then obtaining a representation through an eigendecomposition, as first estimating a covariance or a precision matrix using a non-standard estimator (e.g., a graph Laplacian), and then using the models above for inference.

[3] **for that statistic**—none of our models are generative models for the full dataset.

We will show later in the section, in section 3.2.2, how different algorithms can all be explained as inference methods in this framework. These include dual-probabilistic PCA, random projections, CMDS, LLE, LE, MVU, diffusion maps, kernel-PCA and Isomap.

Before proving the main results, we provide a brief explanation for the notation we use with Wishart distributions. In this section, we denote statements involving Wishart distributed random matrices as,

$$\mathbf{T} \sim \mathcal{W}(\hat{\mathbf{M}}, d).$$

The square random matrix $\mathbf{T}$ without an overset hat or tilde represents a matrix that is scaled in some way, whereas matrices with an overset hat or tilde represent widely-used statistics such as a covariance matrix. Using the model above as an example, $\mathbb{E}(\mathbf{T}) = \hat{\mathbf{M}} * d$. Therefore, we write $\hat{\mathbf{M}}$ as a matrix describing centrality of $\hat{\mathbf{T}}$, where $\hat{\mathbf{T}} = \mathbf{T}/d$. As an additional example, consider the case of dual-probabilistic PCA. The sample pairwise data covariance is calculated as $\hat{\mathbf{S}} = \mathbf{Y}\mathbf{Y}^T/d$, and is an unscaled quantity, in the sense that the centrality parameter of the Wishart $\hat{\mathbf{M}} = \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}$ estimates $\hat{\mathbf{S}}$. In this example, we denote by $\mathbf{S}$ the matrix $\mathbf{Y}\mathbf{Y}^T$, which, assuming that the columns of $\mathbf{Y}$ are independent multivariate normal samples, is Wishart distributed and **scaled by the degrees of freedom** i.e. $\mathrm{S} = d * \mathrm{Cov}(\mathrm{Y})$.

The rest of the section is dedicated to proving the covariance view in section 3.2.1.1 and the precision view in section 3.2.1.2, which immediately follow, and then showing how various algorithms connect to the framework in section 3.2.2.

3.2.1 Derivation of the linear cases

At the start of the section, we presented a view that algorithms involving the eigendecomposition of a similarity matrix are MAP inference algorithms given a Wishart model with a linear kernel used as a centrality parameter. In this subsection, we show how this arises.

The results are inspired by two main results; dual-probabilistic **principal component analysis** of Lawrence (2005), based on the work of Tipping and Bishop (1999), and dual-probabilistic **minor components analysis**, which is a novel perspective we introduce based on the results of Williams and Agakov (2002).

The idea of this section is simply that dual-probabilistic PCA constructs a low-dimensional embedding using the eigendecomposition of a covariance matrix. Any algorithm therefore using an eigendecomposition of a PSD matrix must be using the same model as dp-PCA, but

with a different estimate of the sufficient statistic (and therefore performing quasi-maximum likelihood estimation). Just one modification needs to be made: the Gaussian assumption needs to be written (equivalently) using a Wishart, so that we explicitly model the sufficient statistic—the covariance.

The outline for the derivation follows; we will first show that normal and Wishart statements for models such as ours lead to the same likelihood. Using this fact, we show that dp-PCA can be formulated using a Wishart statement. Finally, we show that describing (misspecifying) the data's precision matrix instead of a covariance with a Wishart distribution leads to the same embeddings (up to a small scaling factor), which leads to the precision view. For simplicity, we generally assume that $\mathbf{Y}$ has a zero mean (it is centred).

Lemma 3.1. Let $\mathbf{F} \in \mathbb{R}^{n \times d}$ and $\mathbf{T} := \tilde{\mathbf{T}} * d := \mathbf{F}\mathbf{F}^T$. The log likelihood of a multivariate normal with zero mean and unknown covariance $\hat{\mathbf{M}}$ is equal to the likelihood of a Wishart with centrality parameter $\hat{\mathbf{M}}$. Concretely, assuming,

$$\mathbf{F}|\hat{\mathbf{M}} \sim \mathcal{MN}(0, \hat{\mathbf{M}}, \mathbf{I}_d) \text{ and } \mathbf{T}|\hat{\mathbf{M}} \sim \mathcal{W}\left(\hat{\mathbf{M}}, d\right),$$
$$\log p_{\mathcal{MN}}(\mathbf{F}|\hat{\mathbf{M}}) \overset{+}{=} \log p_{\mathcal{W}}(\mathbf{T}|\hat{\mathbf{M}}) \overset{+}{=} -\frac{d}{2}\mathrm{tr}\left(\tilde{\mathbf{T}}\hat{\mathbf{M}}^{-1}\right) - \frac{d}{2}\log|\hat{\mathbf{M}}|.$$

Proof. of lemma 3.1. In the multivariate normal case,

$$\begin{aligned}
\mathcal{L}(\mathbf{F}) = \log p_{\mathcal{MN}}(\mathbf{F}|\hat{\mathbf{M}}) &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{I}_d\mathbf{F}^T\hat{\mathbf{M}}^{-1}\mathbf{F}\right) - \frac{d}{2}\log|\hat{\mathbf{M}}| - \frac{n}{2}\log|\mathbf{I}_d| - \frac{nd}{2}\log 2\pi \\
&= -\frac{d}{2}\mathrm{tr}\left(\frac{1}{d}\mathbf{F}\mathbf{F}^T\hat{\mathbf{M}}^{-1}\right) - \frac{d}{2}\log|\hat{\mathbf{M}}| + c, \qquad \text{(trace is cyclic)} \\
&= -\frac{d}{2}\mathrm{tr}\left(\tilde{\mathbf{T}}\hat{\mathbf{M}}^{-1}\right) - \frac{d}{2}\log|\hat{\mathbf{M}}| + c.
\end{aligned}$$

In the Wishart case when $d \geq n$, the sampling distribution of $\mathbf{F}\mathbf{F}^T$ is by definition Wishart, so the likelihood with respect to $\tilde{\mathbf{T}}$ is obtained easily,

$$\mathcal{L}(\mathbf{F}) = \log p_{\mathcal{W}}(\mathbf{F}\mathbf{F}^T|\hat{\mathbf{M}}) = \log p_{\mathcal{W}}(\tilde{\mathbf{T}} * d|\hat{\mathbf{M}}) = -\frac{d}{2}\mathrm{tr}\left(\hat{\mathbf{M}}^{-1}\tilde{\mathbf{T}}\right) - \frac{d}{2}\log|\hat{\mathbf{M}}| + c.$$

In the case when $d < n$, the distribution of $\mathbf{F}\mathbf{F}^T$ is a singular Wishart, a description of which is given by Uhlig (1994) (Theorem 6), who shows that the likelihood is identical to the statement above up to additive constants. □

Using this fact, we are now able to prove the covariance and precision views of the linear ProbDR cases (i.e. the cases that use a linear covariance kernel).

3.2.1.1 Dual-probabilistic PCA

We will now derive the covariance view of linear ProbDR. We use the equivalence derived between normal and Wishart modelling statements to show that dp-PCA can be written as a Wishart statement, which immediately gives us the first half of the main result, restated below,

$$\hat{\mathbf{S}} * \nu|\mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}_n, \nu\right) \quad \Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q \text{ maj}}(\mathbf{\Lambda}_{d_q \text{ maj}} - \hat{\sigma}^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T$$

of this section, theorem 3.1.

Lemma 3.2 (Dual-probabilistic principal components analysis dp-PCA)**.** The MAP estimate of $\mathbf{X}$ assuming a Wishart with a linear kernel (or a linear GP),

$$\mathcal{N}(\mathrm{Y}|0, \mathbf{X}\mathbf{X}^T + \sigma^2 I) \text{ or } \mathcal{W}(\mathrm{S}|\mathbf{X}\mathbf{X}^T + \sigma^2 I, d)$$

where $\mathrm{S} := \tilde{\mathrm{S}} * d = \mathrm{Y}\mathrm{Y}^T$ and $\mathbf{C} = \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}$ corresponds to optimisation of the likelihood,

$$\arg\max_{\mathrm{X}} -\frac{d}{2}\log|\mathbf{C}| - \frac{d}{2}\mathrm{tr}(\tilde{\mathrm{S}}\mathbf{C}^{-1}) + c,$$

the solution to which occurs at the major scaled eigenvectors of the estimator $\tilde{\mathbf{S}}$,

$$\hat{\mathbf{X}} = \mathbf{U}_{d_q}(\mathbf{\Lambda}_{d_q} - \hat{\sigma}^2\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T,$$

where $\hat{\sigma}^2 = \frac{\sum_{i=d_q+1}^{n}\lambda_i}{n-d_q}$ and $\mathbf{U}_{d_q}$ and $\mathbf{\Lambda}_{d_q}$ are the matrices of d_q major eigenvectors and eigenvalues of $\tilde{\mathrm{S}}$.

Proof of lemma 3.2. The Wishart model is equivalent to the normal case due to lemma 3.1. The main result is due to Lawrence (2005), which is based on Tipping and Bishop (1999). A sketch

proof using results from Petersen et al. (2008); Minka (1997):

$$\begin{aligned}
&\arg\max_{\mathbf{X}} -\frac{d}{2}\log|\mathbf{C}| - \frac{d}{2}\mathrm{tr}(\tilde{\mathbf{S}}\mathbf{C}^{-1}) \\
&= \arg\min_{\mathbf{X}} \log\det(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}) + \mathrm{tr}\left(\hat{\mathbf{S}}(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I})^{-1}\right) \\
&\implies 2\mathbf{C}^{-1}\mathbf{X} - 2\mathbf{C}^{-1}\hat{\mathbf{S}}\mathbf{C}^{-1}\mathbf{X} = 0 \\
&\implies \hat{\mathbf{S}}\mathbf{C}^{-1}\mathbf{X} = \mathbf{X}.
\end{aligned} \tag{3.5}$$

The following results were used as part of eq. (3.5),

$$d\log\det\mathbf{C} = \mathrm{tr}(\mathbf{C}^{-1}d\mathbf{C}) = \mathrm{tr}(\mathbf{C}^{-1}(d\mathbf{X}\mathbf{X}^T + \mathbf{X}d\mathbf{X}^T)) = 2\mathrm{tr}(\mathbf{X}^T\mathbf{C}^{-1}d\mathbf{X})$$

and,

$$\begin{aligned}
d\mathrm{tr}(\mathbf{S}\mathbf{C}^{-1}) &= \mathrm{tr}(\mathbf{S}d\mathbf{C}^{-1}) \\
&= -\mathrm{tr}(\mathbf{S}\mathbf{C}^{-1}d\mathbf{C}\mathbf{C}^{-1}) \\
&= -\mathrm{tr}(\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}(\mathbf{X}d\mathbf{X}^T + d\mathbf{X}\mathbf{X}^T)) \\
&= -2\mathrm{tr}(\mathbf{X}^T\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}d\mathbf{X}).
\end{aligned}$$

We notice that the number of degrees of freedom do not change the optimum, in the Wishart case. The multivariate normal case however is different, as the dimension of the data matrix leads to the number of degrees of freedom. Following Tipping and Bishop (1999), we decompose $\mathbf{X}$ using an SVD,

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{R}^T$$

and hence stationarity condition simplifies as,

$$\begin{aligned}
&\hat{\mathbf{S}}\mathbf{U}(\mathbf{\Lambda}^2 + \sigma^2\mathbf{I})^{-1}\mathbf{\Lambda}\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{R} \\
&\implies \hat{\mathbf{S}}\mathbf{U}(\mathbf{\Lambda}^2 + \sigma^2\mathbf{I})^{-1}\mathbf{\Lambda} = \mathbf{U}\mathbf{\Lambda} && \times\mathbf{R}^T \\
&\implies \hat{\mathbf{S}}\mathbf{U} = \mathbf{U}(\mathbf{\Lambda}^2 + \sigma^2\mathbf{I}).
\end{aligned}$$

The result follows as the conditions read $\hat{\mathbf{S}}\mathbf{u}_j = \lambda_j^s\mathbf{u}_j$, which is an eigenvalue definition. Letting λ_j^s be the j-th eigenvalue of $\hat{\mathbf{S}}$, we find $\lambda_j^2 + \sigma^2 = \lambda_j^s \Rightarrow \lambda_j = \sqrt{\lambda_j^s - \sigma^2}$. □

This proves our covariance view. There are many other ways to motivate the solution of (dual-probabilistic) PCA.[4] In our work, we look for a MAP interpretation to explicitly answer the question: "what is the generative model behind widely used dimensionality reduction methods?" We also choose our model family such that it unifies many DR algorithms **while bridging them with GPLVMs** as they themselves encompass many classical algorithms.

Having proved our covariance view that reinterprets dual-probabilistic PCA, we now prove our precision view, which introduces dual-probabilistic MCA.

3.2.1.2 Dual-probabilistic MCA

We will now prove the second Wishart statement of theorem 3.1,

$$\hat{\mathbf{\Gamma}} * \nu|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n)^{-1}, \nu\right) \quad \Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q\,\min}(\mathbf{\Lambda}^{-1}_{d_q\,\min} - \hat{\beta}\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T,$$

thereby completing the main theoretical claim of this section. This is a novel perspective that we term **dual-probabilistic minor components analysis**.

Theorem 3.3. Dual probabilistic minor components analysis (dual-MCA) We present a dimensionality reduction method using the result of probabilistic minor components analysis (Williams and Agakov, 2002). Using this algorithm, and given an estimated/empirical precision matrix $\tilde{\mathbf{\Gamma}}$, we find that a low dimensional embedding $\mathbf{X}$ by maximising objectives of the form,

$$\arg\max_{\mathbf{X}} \frac{\nu}{2}\log|\mathbf{P}^{-1}| - \frac{\nu}{2}\text{tr}(\mathbf{P}^{-1}\tilde{\mathbf{\Gamma}}) + c$$

with $\mathbf{P}^{-1} := \mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n$ is attained at the minor scaled eigenvectors of the estimator,

$$\hat{\mathbf{X}} = \mathbf{U}_{d_q}(\mathbf{\Lambda}^{-1}_{d_q} - \hat{\beta}\mathbf{I}_{d_q})^{1/2}\mathbf{R}^T,$$

[4] In the foundational work of Pearson (1901), the problem is motivated as finding best fitting lines to points in space, in Hotelling (1933) as directions of maximal variance. Lawrence (2005) show that a KL-minimisation between Gaussians can also lead to the result in addition to the maximum likelihood interpretation, and Van Assel et al. (2022) show that a KL-minimisation between posteriors over precisions assuming general Gaussian generative models also leads to a dp-PCA solution when one of the variational families is constrained to be low-rank.

where $\hat{\beta} = \frac{n-d_q}{\sum_{i=d_q+1}^{n} \lambda_i}$ and $\mathbf{U}_{d_q}$ and $\mathbf{\Lambda}_{d_q}$ are the matrices of d_q minor eigenvectors and eigenvalues of $\tilde{\mathbf{\Gamma}}$. Dual minor components analysis is maximum a-posteriori inference given the model,

$$\nu * \tilde{\mathbf{\Gamma}} | \mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta \mathbf{I}_n)^{-1}, \nu\right)$$

where $\tilde{\mathbf{\Gamma}}$ is an empirical precision matrix, for example, calculated[5] as $\tilde{\mathbf{\Gamma}} = (\mathbf{Y}\mathbf{Y}^T/d)^{-1}$.

Proof. of theorem 3.3. The result is based on the result of Williams and Agakov (2002), and follows directly from a flip in notation (for instance $\mathbf{X} \to \mathbf{X}^T$). The result can also be sketched out as follows.

$$\begin{aligned}
\mathcal{L} &\overset{+}{\propto} -\log|\mathbf{P}^{-1}| + \operatorname{tr}(\mathbf{P}^{-1}\tilde{\Gamma}) \\
\arg\min_{\mathbf{X}} \mathcal{L} \implies \frac{d\mathcal{L}}{d\mathbf{X}} &= -\frac{d}{d\mathbf{X}} \log\det(\mathbf{X}\mathbf{X}^T + \hat{\beta}\mathbf{I}) + \frac{d}{d\mathbf{X}} \operatorname{tr}(\tilde{\Gamma}(\mathbf{X}\mathbf{X}^T + \hat{\beta}\mathbf{I})) = 0 \\
&\implies (\mathbf{X}\mathbf{X}^T + \hat{\beta}\mathbf{I})^{-1}\mathbf{X} = \tilde{\mathbf{\Gamma}}\mathbf{X} \\
&\implies \hat{\mathbf{S}}(\mathbf{X}\mathbf{X}^T + \hat{\beta}\mathbf{I})^{-1}\mathbf{X} = \mathbf{X}.
\end{aligned}$$

This is the same stationarity condition as dp-PCA with $\hat{\mathbf{S}} = \tilde{\mathbf{\Gamma}}^{-1}$, and the solution follows. The probabilistic interpretation of the objective in theorem 3.3 follows trivially as it is the likelihood of the models in lemma 3.1 with $\tilde{\mathbf{T}} = \tilde{\mathbf{\Gamma}}$. □

Dual probabilistic minor components analysis is the second statement of theorem 3.1, hence completing the proof of our main statement.

Unlike in the case of dual-probabilistic PCA, where the model statement $\mathbf{S} \sim \mathcal{W}(.)$ arises naturally as the sampling distribution of the covariance ($\hat{\mathbf{S}} = \mathbf{Y}\mathbf{Y}^T/d$, where $\mathbf{Y} \sim \mathcal{MN}(\mathbf{0}, \Sigma, \mathbf{I}_d)$), the sample precision matrix of such a random matrix does not in general follow a Wishart distribution.

Nevertheless, before we close our subsection on proofs, we show below that if we use the inverse of a sample covariance to estimate the precision matrix, the embeddings found by the covariance and precision views are effectively identical, regardless of which method is chosen. This shows that even when a model is misspecified, one can expect to obtain semantically consistent results within this model class, and that there is no strange edge-behaviour.

[5] If the covariance matrix $\hat{\mathbf{S}}$ is low rank, then $\tilde{\mathbf{\Gamma}}$ can be set to its pseudo-inverse.

3.2.1.3 Dual-probabilistic PCA via dual-probabilistic MCA

Below, we show that dual-probabilistic PCA and dual-probabilistic MCA obtain embeddings that are similar, differing only by a multiplicative factor (they are in fact equivalent when the noise level tends to zero), even though they are not equivalent probabilistic statements[6]. This forms a proof of our framework being "correct" under transformation (inversion) of the statistic. The idea below is simple; the precision and covariance share major and minor eigenvectors respectively, when they are related by a (pseudo-)inverse. Dp-PCA estimates the covariance, and dp-MCA estimates the precision, therefore, the major eigenvectors of the covariance recovered by dp-PCA are the minor eigenvectors of the precision recovered by dp-MCA.

Lemma 3.3 (Equivalence of probabilistic PCA and probabilistic MCA)**.** Let $\tilde{\mathbf{S}} \coloneqq \mathbf{S}/d \coloneqq \mathbf{Y}\mathbf{Y}^T/d$ and $\tilde{\mathbf{\Gamma}} \coloneqq \mathbf{\Gamma}/d \coloneqq \tilde{\mathbf{S}}^{-1}$. The matrices $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{\Gamma}}$ share eigenvectors, represented by the matrix $\mathbf{U}_{d_q}$ and their diagonal eigenvalue matrices are $\mathbf{\Lambda}_{\tilde{\mathbf{S}}}$ and $\mathbf{\Lambda}_{\tilde{\mathbf{S}}}^{-1}$ respectively. Then, the estimated embeddings of the dp-PCA and dp-MCA models are related by a diagonal multiplicative factor, and the covariance estimates (i.e. the matrix $\mathbf{X}\mathbf{X}^T + \text{coef} \cdot \mathbf{I}$) found are identical in form,

$$\mathbf{S}|\mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, d\right), \quad \mathbf{\Gamma}|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, d\right)$$

Proof. of lemma 3.3. The proof is due to lemma 3.2 and theorem 3.3. In dp-PCA,

$$\hat{\mathbf{S}}_{\text{PCA}} = \hat{\mathbf{X}}\hat{\mathbf{X}}^T + \hat{\sigma}^2\mathbf{I}_n = \mathbf{U}_{d_q}(\mathbf{\Lambda}_{\hat{\mathbf{S}}} - \hat{\sigma}^2\mathbf{I})\mathbf{U}_{d_q}^T + \sigma^2\mathbf{I}_n = \mathbf{U}_{d_q}\mathbf{\Lambda}_{\hat{\mathbf{S}}}\mathbf{U}_{d_q}^T + \hat{\sigma}^2(\mathbf{I}_n - \mathbf{P}),$$

where $\mathbf{P} = \mathbf{U}_{d_q}\mathbf{U}_{d_q}^T$ is a projector and in dp-MCA,

$$\hat{\mathbf{S}}_{\text{MCA}} = \hat{\mathbf{X}}\hat{\mathbf{X}}^T + \hat{\beta}\mathbf{I} = \mathbf{U}_{d_q}(\mathbf{\Lambda}_{\hat{\mathbf{\Gamma}}}^{-1} - \hat{\beta}\mathbf{I}_{d_q})\mathbf{U}_{d_q}^T + \hat{\beta}\mathbf{I}_n = \mathbf{U}_{d_q}\mathbf{\Lambda}_{\hat{\mathbf{S}}}\mathbf{U}_{d_q}^T + \hat{\beta}(\mathbf{I}_n - \mathbf{P}),$$

which differs from the dp-PCA solution by $(\hat{\sigma}^2 - \hat{\beta})(\mathbf{I} - \mathbf{P})$, which will be close to a diagonal matrix with large n, as the elements of the eigenvectors scale as $1/\sqrt{n}$ (for them to be orthonormal),

[6] The equivalent statement for dp-PCA would involve describing the precision using an inverse-Wishart,

$$\underbrace{\mathbf{\Gamma}|\mathbf{X} \sim \mathcal{W}^{-1}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, d\right)}_{\text{equivalent to dp-PCA}} \text{ and not } \underbrace{\mathbf{\Gamma}|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, d\right)}_{\text{dp-MCA}}.$$

and the elements of the projector therefore scale as $1/n$. □

Although the embedding estimates are the same, the noise levels are not, and therefore, using the misspecified model (i.e. the Wishart placed on the precision matrix) results in a biased estimator of the unexplained variance.

Theorem 3.4 (Estimated noise level in dp-PCA and dp-MCA)**.** The estimated noise level in dp-MCA $\hat{\beta}$ is lower than its counterpart in dp-PCA $\hat{\sigma}^2$,

$$\hat{\beta} \leq \hat{\sigma}^2.$$

Proved in appendix B.1.

Therefore, even when using a misspecified model (dp-MCA[7] used in place of dp-PCA), we have shown the embeddings to be at least proportional to those of dp-PCA, ensuring a form of logical consistency.

This concludes our subsection on proofs of dp-PCA and dp-MCA that make up the core theoretical claim of the linear ProbDR cases. In the next and final subsection, we show how various DR algorithms used in practice fit into our framework.

3.2.2 Explaining eigendecomposing algorithms as linear ProbDR

In the previous section, we proved that ideas of dp-PCA and dp-MCA underpin the core claim of the section, that embeddings found through eigendecomposition may be related to the models,

$$\begin{aligned}
\hat{\mathbf{S}} * \nu | \mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2 \mathbf{I}_n, \nu\right) \quad &\Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q \text{ maj}} (\mathbf{\Lambda}_{d_q \text{ maj}} - \hat{\sigma}^2 \mathbf{I}_{d_q})^{1/2} \mathbf{R}^T \\
\hat{\mathbf{\Gamma}} * \nu | \mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta \mathbf{I}_n)^{-1}, \nu\right) \quad &\Rightarrow \quad \hat{\mathbf{X}}_{\text{MAP}} = \mathbf{U}_{d_q \text{ min}} (\mathbf{\Lambda}^{-1}_{d_q \text{ min}} - \hat{\beta} \mathbf{I}_{d_q})^{1/2} \mathbf{R}^T.
\end{aligned}$$

In this section, we show how many dimensionality reduction algorithms arise as inference algorithms within ProbDR or as trivial extensions thereof. By doing so, we exemplify that many algorithms used in practice have probabilistic interpretations within the same model framework, the main claim of the thesis. Many algorithms use unscaled eigenvectors of the PSD matrices they form, whereas our interpretations yield latents scaled by a diagonal eigenvalue-related

[7] Dp-MCA can have a normal factorisation and therefore be correctly specified, when a factor of the precision matrix is assumed to be normally distributed. If, $\text{In}(\mathbf{Y}) \sim \mathcal{MN}(0, \Sigma^{-1}, \mathbf{I}_d)$, where $\mathbf{\Gamma} := \text{In}\text{In}^T$, then $\mathbf{\Gamma} \sim \mathcal{W}(\Sigma^{-1}, d)$. In can represent an incidence matrix.

matrix. This difference is insignificant for most use-cases, and our results are thus approximate in light of this difference.

We categorise the results that follow as,

- methods that define a covariance-like matrix and obtain major eigenvectors as an embedding, such as dp-PCA, GPLVM CMDS, Isomap, kPCA and MVU.
- methods that define a graph Laplacian and obtain minor eigenvectors as an embedding, such as Laplacian Eigenmaps and LLE.
- Diffusion maps, that uses eigenvectors of a matrix that is not PSD.

We start with methods that estimate covariance-like matrices.

3.2.2.1 Algorithms that eigendecompose covariance-like matrices

Here, we show that the CMDS, Isomap, kernel PCA and MVU algorithms first calculate a similarity matrix $\mathbf{K}$, which is either PSD or nearly PSD, and obtain an embedding through eigendecomposition, retaining the eigenvectors corresponding to the largest eigenvalues. The connection to our framework is then clear, as we described our linear ProbDR cases as models for algorithms that use eigendecomposition for inference. The only difference between algorithms stems from how the covariance statistic is computed. We also show that GPLVMs and projective methods fall into this view.

Classical multidimensional scaling CMDS was introduced by Torgerson (1952) and described in Gower (1966) as the dual of classical principal component analysis. CMDS is motivated by the fact that a double-centred negative distance matrix,

$$\mathbf{K} = -0.5\mathbf{H}\mathbf{D}^2\mathbf{H},$$

if PSD of rank r, can be considered a Gram matrix; in other words, an isometric Euclidean embedding $\mathbf{Y}$ exists, with $\mathbf{Y} \in \mathbb{R}^{n\times r}$, such that $\mathbf{K} = \mathbf{Y}\mathbf{Y}^T$ (Young and Householder, 1938). Using a loss function that minimises squared residual distances between this distance matrix and one constructed using latent variables $\mathbf{X} \in \mathbb{R}^{n\times d_q}$,

$$\mathcal{L}_{ij} = \left(\mathbf{D}_{ij}^2 - \|\mathbf{X}_i - \mathbf{X}_j\|^2\right)^2$$

results in a solution that sets $\mathbf{X}$ to the top d_q eigenvectors of $\mathbf{K}$ (Tenenbaum et al., 2000).

Isomap Other methods, such as the Isomap algorithm introduced by Tenenbaum et al. (2000), use distance matrices constructed using non-Euclidean distance metrics. The Isomap algorithm $\mathbf{D}$ to be shortest path distances on a nearest-neighbour graph constructed using the high-dimensional data $\mathbf{Y}$. Then, an embedding is obtained using the method of MDS as above; by obtaining d_q major eigenvectors of the matrix $\mathbf{K} = -0.5\mathbf{H}\mathbf{D}^2\mathbf{H}$.

Algorithms such as Isomap have natural scientific use-cases, shown by Pietal et al. (2015) who introduce the algorithm as GDFuzz3d. Within the context of protein structure prediction, we may have access to a *contact map*—a symmetric adjacency matrix showing which two amino acids of a protein sequence are "in contact".[8] Using the process above, we can calculate $\mathbf{K}$ using graph distances, to find three-dimensional coordinates corresponding to the protein backbone. Unlike many real-world cases, this is an example of a case-study where the underlying latent dimensionality is known a priori.

Although the similarity matrices $\mathbf{K}$ that use non-Euclidean distance metrics are not guaranteed to be PSD, we restrict our search to eigenvectors corresponding to non-negative eigenvalues, which gives us the "nearest" Euclidean embedding is found that reconstructs the distance matrix in a lowest squared-error sense.

Kernel PCA kPCA, introduced by Schölkopf et al. (1997), replaces the Gram matrix obtained with dual-PCA $\mathbf{K} = \mathbf{Y}\mathbf{Y}^T/d$ with a kernel acting on data points i and j, and typically centres the resulting matrix so that the features implied by the kernel are centred;

$$\tilde{\mathbf{K}}_{ij} = k(\mathbf{Y}_i, \mathbf{Y}_j), \quad \mathbf{K} = \mathbf{H}\tilde{\mathbf{K}}\mathbf{H},$$

with the idea that the kernel introduces a non-linear feature map of a much higher dimensionality than the data. The embedding is found as the major eigenvectors of $\mathbf{K}$ as above. Naively setting $\tilde{\mathbf{K}}_{ij} = \exp(-d_{ij}^2)$ for example, where d is a non-Euclidean distance can also result in a non-PSD similarity matrix (Feragen and Hauberg, 2016).[9] Therefore, the restriction of the

[8] This is typically an indicator function activated if the distance between two Cα atoms is less than a thresholding value. Classically, contact maps were based on mutual-information based studies, or calculating sparse-precisions between mutations based on multiple sequence alignments (using GLASSO; Jones et al. (2011)). More recently, their estimation has followed studying attention matrices in neural models, or fitting logistic regressions to true contact maps using representations of protein language models (e.g. see Lin et al. (2023)).

[9] Covariance kernels can be defined on specific manifolds in question (Borovitskiy et al., 2020).

embedding to eigenvectors corresponding just to non-negative eigenvalues applies here too.

Maximum variance unfolding MVU introduced by Weinberger et al. (2004) estimates $\mathbf{K}$ by maximising $\mathrm{tr}(\mathbf{K})$ under PSD, centring and local isometry constraints. The embedding is then found as the eigenvectors of $\mathbf{K}$ corresponding to the largest eigenvalues.

We now propose that these methods are all examples of the covariance view of linear ProbDR.

Proposition 3.1. CMDS, Isomap, kPCA and MVU all construct a (near-)PSD matrix $\mathbf{K}$ and obtain an embedding as the major eigenvectors of $\mathbf{K}$. Therefore, they correspond to inference within our covariance view,

$$\nu * \mathbf{K} \sim \mathcal{W}(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, \nu),$$

with the specification of $\mathbf{K}$ depending on the algorithm being interpreted.

Before closing these results, we remark that GPLVMs and projective methods also fit into this view.

GPLVM Linear factor methods described in section 2.4 that perform full-form inference for the latents $\mathbf{X}$, such as dual-probabilistic PCA, in addition to non-linear latent factor models GPLVMs, assume a model,

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{MN}(0, K(\mathbf{X}, \mathbf{X}), \mathbf{I}_d),$$

where the covariance $\hat{\mathbf{S}}$ is a sufficient statistic, and with additional priors that constrain the nature of the embedding. The sampling distribution of the covariance is Wishart, and hence, inference follows maximum a-posteriori for the latents $\mathbf{X}$ assuming the model,

$$\mathbf{S}|\mathbf{X} \sim \mathcal{W}(K(\mathbf{X}, \mathbf{X}), d),$$

with $\mathbf{S} = \mathbf{Y}\mathbf{Y}^T$ along with the same priors over $\mathbf{X}$ as assumed initially. This is the covariance view of the ProbDR framework, with a linear or non-linear kernel depending on the choice of covariance kernel K.

Projective methods Consider a mapping $\mathbf{X} = \Phi(\mathbf{Y})$. We show that this can be seen a mode given the model,

$$\nu * \hat{\mathbf{S}}|\mathbf{X} \sim \mathcal{W}(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}, \nu),$$

where $\hat{\mathbf{S}} = \Phi(\mathbf{Y})\Phi(\mathbf{Y})^T + \beta\mathbf{I}$. Consider the stationarity condition that arises assuming the dual-probabilistic PCA and MCA models,

$$\hat{\mathbf{S}}^{-1}\mathbf{X} = (\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n)^{-1}\mathbf{X} \Rightarrow (\Phi\Phi^T + \beta\mathbf{I}_n)^{-1}\mathbf{X} = (\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}_n)^{-1}\mathbf{X}$$

This is attained if $\mathbf{X} = \Phi(\mathbf{Y})$, showing that projective methods can also be seen through our framework.

Next, we turn our attention to algorithms that obtain embeddings as **minor eigenvectors of a graph Laplacian**, instead of major eigenvalues of a covariance-like matrix as the algorithms thus far.

3.2.2.2 Algorithms that eigendecompose precision-like matrices

Here, we detail the Laplacian Eigenmaps and Locally Linear Embedding algorithms, and show that they estimate an embedding through the minor eigenvectors of a precision-like matrix. As before, the connection to ProbDR will then become clear. We will also describe the properties of the graph Laplacian that makes it a suitable estimator for a precision matrix.

Laplacian Eigenmaps Laplacian Eigenmaps introduced by Belkin and Niyogi (2001) generates a (potentially weighted) graph Laplacian matrix and sets the embedding to be the d_q eigenvectors corresponding to the smallest eigenvalues, that are a result of the generalised eigenvalue problem,

$$\mathbf{L}\mathbf{v} = \lambda\mathbf{D}\mathbf{v} \quad \text{equivalently,} \quad \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{x} = \lambda\mathbf{x}.$$

The method of Laplacian Eigenmaps is justified due to a least-squares view, to find vectors $\mathbf{x}$ such that $\sum_{ij}\mathbf{A}_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2$ is minimised but such that $\|\mathbf{x}\|^2 = 1$. This leads to an objective $\mathcal{L} = \text{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$ with orthonormality constraints on $\mathbf{X}$, (which can immediately be read off as the data dependent term of the dp-MCA likelihood, theorem 3.3).

Locally Linear Embedding Lawrence (2012) interprets the LLE algorithm (Roweis and Saul, 2000) to first perform inference on "reconstruction weights" $\mathbf{W}$ via pseudolikelihood

optimisation with a Gaussian Markov random field (GMRF) model,

$$\forall i : \mathbf{Y}_{i:} | \mathbf{Y}_{-i} \sim \mathcal{N}\Big(- \mathbf{W}_{ii}^{-1} \sum_{j \in \mathcal{N}(i)} \mathbf{W}_{ji} \mathbf{Y}_{j:}, \mathbf{W}_{ii}^{-2} \Big),$$

where $\mathbf{W}_{ii} = -\sum_{j \in \mathcal{N}(i)} \mathbf{W}_{ji}$ (although in LLE, this is constrained to be 1) and $\forall j \notin \mathcal{N}(i)$: $\mathbf{W}_{ji} = 0$ and we expect that $\forall j \in \mathcal{N}(i), \mathbf{W}_{ji} < 0$. This leads to an objective of the form $\mathcal{L}(\mathbf{W}_i^T) = \|\mathbf{Y}_i - \sum_{j \in \mathcal{N}(i)} \mathbf{Y}_j \mathbf{W}_{ji}\|^2$. Once these weights have been found, an embedding $\mathbf{X}$ is found that optimises a similar objective, $\mathcal{L} = \sum_i \|\mathbf{X}_i - \sum_{j \in \mathcal{N}(i)} \mathbf{X}_j \mathbf{W}_{ji}\|^2$, which is solved using an eigenvalue problem. Therefore, setting the graph Laplacian $\mathbf{L} = \hat{\mathbf{\Gamma}}(\mathbf{Y}) = \mathbf{W}\mathbf{W}^T$ as in Lawrence (2012) and Belkin and Niyogi (2001) recovers LLE as a case of Laplacian Eigenmaps.

Proposition 3.2. Laplacian Eigenmaps and LLE construct a graph Laplacian matrix $\mathbf{L}$ that may be weighted and/or normalised, and obtain an embedding as the minor eigenvectors of $\mathbf{L}$. Therefore, they correspond to inference within our precision view,

$$\nu * \mathbf{L} | \mathbf{X} \sim \mathcal{W}\left(\left(\mathbf{X}\mathbf{X}^T + \beta \mathbf{I} \right)^{-1}, \nu \right),$$

with the specification of $\mathbf{K}$ depending on the algorithm being interpreted.

We will now review the implications of the model, by studying what problem Laplacian Eigenmaps solves, and the role of a graph Laplacian as an estimator of a precision matrix.

Embeddings used for spectral clustering, e.g. those described in Shi and Malik (2000); von Luxburg (2007) which are motivated by finding an approximate solution to the minimal graph-cut problem, are nearly identical to Laplacian Eigenmaps. Given a graph, the graph-cut problem aims to find a partition that splits the graph into two, with each group, as an example, having maximum interconnected-ness. These ideas provide a semantic statement for what the embeddings recovered from such an algorithm/model correspond to; as dp-PCA can be thought to recover directions of maximal variation (or possibly slow modes of variation), and as ICA can be thought to recover disentangled signals, the model underpinning Laplacian Eigenmaps can be thought to recover coordinates that best separate the data graph by latent clusters.

Next, we study the Laplacian matrix as (part of) a precision matrix. Graph Laplacians (or matrices with their sparsity structure) commonly appear as precision matrices within models.[10]

[10]Such as, in intrinsic conditional autoregressive models (ICAR models, Besag (1974)). Besag (1974) show that the Gibbs-like specification of marginals does not necessarily lead to a consistent joint distribution, but does in

The graph Laplacian satisfies many of the expected properties of a precision matrix, for example,

1. It is positive semidefinite.

2. When an off-diagonal element is non-zero and negative, the two corresponding random variables are known to be conditionally dependent with a positive correlation. Concretely, assume a random vector $\mathbf{x}$ indexed by Ω and distributed as $\mathbf{x}_\Omega \sim \mathcal{N}(0, \mathbf{\Gamma}^{-1})$.

   (a) The normalised precision matrix $\bar{\mathbf{\Gamma}}$ is such that $\bar{\mathbf{\Gamma}}_{ij} = 0 \Leftrightarrow i \perp j | \{\Omega \setminus ij\}$. In words, $\mathbf{x}_i$ and $\mathbf{x}_j$ are conditionally independent given the other part of the random vector.

   (b) $\bar{\mathbf{\Gamma}}_{ij} < 0 \Leftrightarrow \text{Cor}(\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_{\Omega \setminus ij}) > 0$ (Lauritzen (1996), Section 5.1.3).

   The graph Laplacian also shares these properties, as when two points are adjacent, its off diagonal elements are negative, and zero otherwise.

3. A Bayesian network's precision matrix shares the same sparsity pattern as the associated moralised graph Laplacian (Loh and Bühlmann, 2013).

Therefore, the graph Laplacian functions as an estimator of a precision matrix. The degrees of the graph Laplacian also have ties to data density.[11]

An implication of using the GL as a precision matrix is that it is a function of *just* the data's (nearest-)neighbour graph, we posit that it may be a lossy estimate of certain data characteristics. Research into understanding what features are extracted at a high-level by the various covariance estimators in this chapter will shed light on the best circumstances for the usage of each of the methods discussed. A question that arises is, to what extent nearest-neighbour graphs of highly zero inflated distributions (often seen in science), are solely functions of zero-occurrence rates. We note this in light of the observation that visualisation algorithms based on the neighbour graph can be somewhat robust to binarisation of the data, illustrated in fig. 3.2. We see that for a particular scRNA-seq dataset, binarising the data matrix does not have a catastrophic effect on the resultant embeddings. Conversely, there are cases (such as the case-study in appendix A.11) where clustering in the data is not found in its

the case of ICAR-like models. Furthermore, precision matrices can be built (obeying a discretisation of a manifold) which lead to the resulting covariance being Matérn (Lindgren et al., 2011).

[11] For random walks on the graph, the stationary distribution is proportional to the degree vector (von Luxburg, 2007), and if the graph Laplacian is built using a kernel as in diffusion maps, then the degree vector encodes the data points' kernel density estimate Coifman and Lafon (2006).

dominant-variance components. Therefore an open question remains as to how similarity measures for such cases should be constructed.[12]

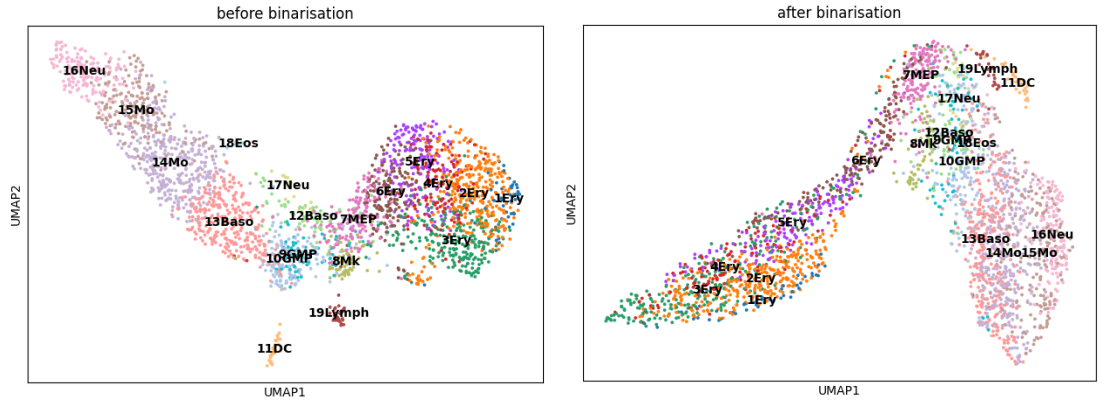


Figure 3.2: Cell-type embedding positions are largely retained even when the underlying data is binarised, reflecting simply that the zero-occurrence rates explain a large part of the visualisation, which is based on just the data's nearest neighbour graph.

Before concluding these results, we point out that the model can, due to the equivalence of dp-PCA and dp-MCA, be written in terms of a covariance. In fact, Lawrence (2012) presents an algorithm similar to Laplacian Eigenmaps that instead chooses the top eigenvectors of a covariance-like matrix, $\hat{\mathbf{S}}(\mathbf{Y}) = \mathbf{H}(\mathbf{L} + \gamma\mathbf{I})^{-1}\mathbf{H}$. In part, this can be motivated by the fact that Laplacian Eigenmaps discards the constant eigenvector corresponding to zero eigenvalues, **which is not done** in our precision view. Choosing this definition of covariance yields the following result,

$$\hat{\mathbf{S}} = \mathbf{H}(\mathbf{L} + \gamma\mathbf{I})^{-1}\mathbf{H} \overset{\text{eig}}{=} \mathbf{H}\mathbf{U}(\gamma\mathbf{I} + \mathbf{\Lambda})^{-1}\mathbf{U}\mathbf{H} \overset{\gamma\to 0}{\to} \underbrace{\mathbf{H}\mathbf{1}\mathbf{1}^T\mathbf{H}/n\gamma}_{\mathbf{0}} + \mathbf{L}^+ = \mathbf{L}^+$$

where $\mathbf{L}^+$ is formed by removing columns corresponding to the zero eigenvalue(s). Therefore, Laplacian Eigenmaps can also be interpreted using our dp-PCA view, employing the estimator $\hat{\mathbf{S}}(\mathbf{Y}) = \mathbf{L}^+$, which does discard the constant eigenvector.

We have shown that Laplacian Eigenmaps and its extensions fall into the ProbDR framework, and that the graph Laplacian can play a meaningful role as an estimator of a precision matrix. We end our subsection by considering one last algorithm, that uses an eigendecomposition of a matrix that is not PSD, unlike the cases we presented thus far.

[12] A line of work in this direction can be found in the work of Skinnider et al. (2019), who show that measures of proportionality (a statistically non-standard similarity estimator) yield gene–gene networks that better recover known biological structure than the sample correlation in scRNA-seq datasets.

3.2.2.3 Algorithms that eigendecompose non-PSD matrices

This section mainly reviews the diffusion maps algorithm of Coifman and Lafon (2006), which uses the eigendecomposition of a transition matrix to find an embedding.

Diffusion maps The algorithm estimates a transition matrix assuming that each data point is a node of a graph,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{K},$$

where $\mathbf{K}_{ij} = k(\mathbf{Y}_i, \mathbf{Y}_j)$ is a kernel matrix acting on the data $\mathbf{Y}$, and $\mathbf{D}_{ii} = \sum_k \mathbf{K}_{ik}$ left-normalises the matrix. An embedding is found such that Euclidean distances approximate diffusion distances on the graph, and hence, the embeddings are computed as the major eigenvectors of $\mathbf{P}^t$. Set $t = 1$.

Proposition 3.3. The major eigenvectors of $\mathbf{P}$ are the minor eigenvectors of $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$ (as the implied adjacency is a *similar* matrix to the transition matrix), we see that the diffusion maps algorithm can be seen as a case of MAP inference within ProbDR's precision view,

$$v * \mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, v\right).$$

We will briefly review the interpretation of the graph Laplacian used in this context. A slightly different construction presented of the transition matrix, that is also presented by Coifman and Lafon (2006) is as follows: first, a normalised kernel matrix is constructed as $\mathbf{W} = \mathbf{D}^{-1}\mathbf{K}_\epsilon\mathbf{D}^{-1}$, where $\mathbf{K}_\epsilon$ is calculated using distances scaled with a factor of $1/\epsilon$. Then, a transition matrix is constructed as $\tilde{\mathbf{P}} = \tilde{\mathbf{D}}^{-1}\mathbf{W}$, where $\tilde{\mathbf{D}}_{ii} = \sum_k \mathbf{W}_{ik}$. This transition matrix approximates a symmetric positive definite matrix, the heat kernel $\exp(-t\Delta)$, where Δ is the Laplace-Beltrami operator, approximated by $\mathbf{L}_\epsilon^{rw} = (\mathbf{I} - \tilde{\mathbf{P}})/\epsilon$.[13] Therefore, no matter which matrix is used for eigendecomposition, the statistic used estimates the manifold that underpins the data. Other interpretations of diffusion maps may be possible, but we detail the Wishart due to its comparability with other methods[14].

To conclude the section, we have shown that there are two main model classes, dp-PCA

[13] With this construction, the graph Laplacian is an analogue of the Markov chain generator.

[14] For example, a transition matrix $\mathbf{P}$ could be modelled as a matrix normal, or more appropriately, its rows could be modelled using a Dirichlet, or Von Mises distribution with centrality parameters $\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}_n$, all of which would lead to a functional form within the log-likelihood, $\mathrm{tr}(\mathbf{P}\mathbf{X}\mathbf{X}^T)$.

and dp-MCA, that we term the linear ProbDR views,

$$\text{dp-PCA:} \quad \nu * \hat{\mathbf{S}}|\mathbf{X} \sim \mathcal{W}\left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, \nu\right) \text{ and,}$$
$$\text{dp-MCA:} \quad \nu * \mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, \nu\right),$$

that underpin many methods of dimensionality reduction that use eigendecomposition of a matrix as an embedding. We also show an equivalence between the views in the sense of the solutions they find. In the next section, we will show that t-SNE-like algorithms follow a similar generative model as our Laplacian Eigenmaps model, but with a non-linear covariance function used in place of the linear kernel.

3.3 Likelihood interpretations: non-linear cases

In the previous section, we showed how the linear ProbDR model,

$$\nu\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left(\left(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}\right)^{-1}, \nu\right)$$

where $\mathbf{L}$ is a graph Laplacian corresponding to the data's k-NN graph, explains algorithms such as Laplacian Eigenmaps. This leads to an eigendecomposition of the estimated data covariance/precision to obtain the latent variable $\mathbf{X}$. In this chapter, we show that t-SNE-like algorithms are a non-linear extension of this model.[15]

Concretely, in this section, we show that dimensionality reduction methods that are neighbour embedding algorithms, such as UMAP and t-SNE, can be recast approximately, in the large-n limit, as MAP inference methods corresponding to an almost identical model. The key modification that arises from our study of t-SNE-like algorithms is the introduction of a non-linear covariance function to the model that underpins Laplacian Eigenmaps. We assume an improper prior on the latents, $p(\mathbf{X}) \propto 1$, as before. The model now becomes,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left(\left(\mathbf{X}\mathbf{X}^T + 0.5\mathbf{H}K(\mathbf{X},\mathbf{X})\mathbf{H} + \gamma\mathbf{I}\right)^{-1}, n\right), \tag{3.6}$$

where $\mathbf{L}$ is an estimate of the graph Laplacian generated using the high-dimensional data $\mathbf{Y} \in \mathbb{R}^{n\times d}$, $\mathbf{X} \in \mathbb{R}^{n\times d_q}$ corresponds to the set of low (d_q-)dimensional latent variables, and K

[15] This section is based on **Ravuri and Lawrence (2024)**.

is a covariance function (the Cauchy/Student-t/rational quadratic kernel) used to construct a positive-definite matrix using latent variables.

This enables a direct comparison between algorithms such as Laplacian Eigenmaps, t-SNE and UMAP. Our interpretation offers deeper theoretical and semantic insights into such algorithms and forges a connection to Gaussian process latent variable models by showing that well-known kernels can be used to describe covariances implied by the graph Laplacian.

We also show that a precursor interpretation of our methods is a simple edge detection model,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right),$$

where the adjacencies of the kNN graph are described directly with a Bernoulli distribution, using distances between the latent variables to describe the probabilities of adjacency.

After presenting our main interpretation, we show that the models correspond to meaningful assumptions, and that they are coherent under transformations (i.e. when we switch from the Bernoulli to the Wishart interpretations). To close the section, we will show how ideas from word2vec can be used for approximate but efficient inference in t-SNE-like algorithms.

We now review a key result of the last section (our Laplacian Eigenmaps interpretation) and Contrastive Neighbour Embedding, which forms the foundation of our work.

3.3.1 Background

As a background to the main results of the section, we recap the Laplacian Eigenmaps interpretation of ProbDR as this, in a way, forms a "linear edge-case"/limiting result of our interpretation of t-SNE-like algorithms. Then we present a key result that we use to build our interpretation; the **Contrastive Neighbour Embedding** introduced by Damrich et al. (2022), which simplifies t-SNE and UMAP. It provides the main objective function that we use to interpret t-SNE-like algorithms as models over the adjacency matrix and then, the graph Laplacian.

3.3.1.1 A recap of probabilistic Laplacian Eigenmaps

We briefly recap Laplacian Eigenmaps' model from the previous section. The algorithm involves the calculation of a nearest-neighbour graph using high-dimensional data points $\mathbf{Y}$, which can be represented using a graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$. $\mathbf{A}$ is the corresponding adjacency matrix, and $\mathbf{D}$ is the diagonal degree matrix, $\mathbf{D}_{ii} = \sum_k \mathbf{A}_{ik}$). Then, the embedding $\mathbf{X}$ is set to the

eigenvectors of $\mathbf{L}$ corresponding to the lowest eigenvalues. Although Laplacian Eigenmaps uses the symmetrically normalised graph Laplacian, in this section, we use the ordinary Laplacian for ease of computation and linearity in the adjacency. In section 3.2.2.2, we showed that this corresponds to inference for $\mathbf{X}$ by maximising the likelihood,

$$\log \mathcal{W}\left(\nu * \mathbf{L} | (\mathbf{X}\mathbf{X}^T + \beta \mathbf{I})^{-1}, \nu\right), \tag{3.7}$$

where $\mathbf{L}$ can, as before, be interpreted as an estimate of a precision matrix. The model results in the implied covariance ($\mathbf{L}^+$) being modelled using a linear covariance function acting on the latents $\mathbf{X}$, which is familiar in models such as dual probabilistic PCA and GPLVM.

As an example of visualisations that are obtained from using Laplacian Eigenmaps, fig. 3.3 shows embeddings of three datasets; the first corresponding to 10k vectorised MNIST digits (Deng, 2012), and the rest corresponding to embeddings of transcriptomics datasets[16] from Macosko et al. (2015) and Zheng et al. (2017). The figure illustrates that clustering is achieved by data-point type, and that the embeddings are "sharp" or "pointed".

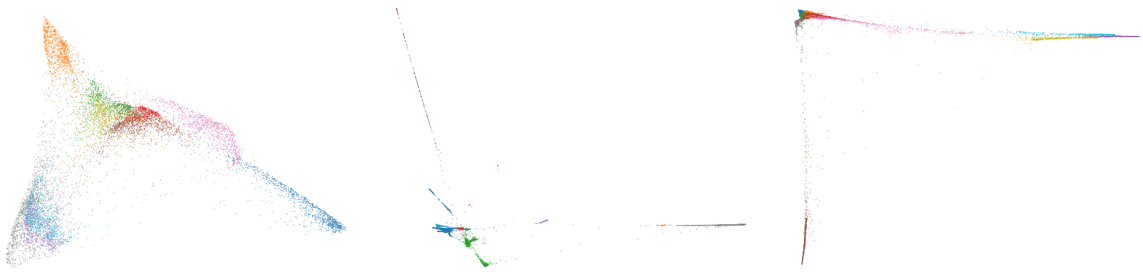


Figure 3.3: Comparison between embeddings obtained using our Laplacian Eigenmaps, across different datasets, showing clustering by data type. Tasks, left to right: clustering of MNIST digits, and cells from two transcriptomic datasets.

Next, we review the contrastive neighbour embedding framework, which provides an objective that forms the basis of our interpretations.

3.3.1.2 Contrastive neighbour embedding

We now present the contrastive neighbour embedding (CNE) framework, which provides a clear objective, when optimised, leads to t-SNE and UMAP-like behaviour. This provides the

[16]The datasets were accessed using the openTSNE repository (Poličar et al., 2024). We resample each dataset so that it has exactly 10 groups to aid visualisation, with up to 10k points in total. The optimisation was done using an A100 GPU with 80 GiB of GPU memory, using pytorch's Adam optimiser (Kingma and Ba, 2017), with an initial learning rate set to 1.0. Each experiment was run for 100 epochs, with linear rate decay.

central objective which we interpret in this section, allowing for links to be drawn between t-SNE-like algorithms, GPLVMs and ProbDR.

UMAP and t-SNE are defined as KL-minimising algorithms and can easily be interpreted in a variational framework (described briefly later in the chapter), acting on a binary adjacency matrix $\mathbf{A}'$. If the variational probabilities are one or zero, signifying whether two points are nearest neighbours or not, the interpretation becomes equivalent to MAP estimation[17] (Damrich et al., 2022; Ravuri et al., 2023).

This simplification can be made due to the findings of Damrich and Hamprecht (2021), who show that the relatively complex calculation of the variational probabilities in t-SNE and UMAP can be replaced with simply the adjacency matrices without significant loss of performance. Damrich et al. (2022), however, show that the optimisation process is equally important. As part of an extensive study on the nature of the t-SNE and UMAP loss functions, they show how the stochastic optimisation of t-SNE and UMAP is contrastive estimation with the objective function (which is maximised)[18],

$$\mathcal{E}(\mathbf{X}) \propto \underbrace{\sum_{ij} \mathbf{A}_{ij} \log\left(\frac{1}{d_{ij}(\mathbf{X})^2+1}\right)}_{\mathcal{T}_a} + \underbrace{\frac{4n_{\text{neg}}n_{\text{neigh}}}{3n} \sum_{ij}(1-\mathbf{A}_{ij}) \log\left(1-\frac{1}{d_{ij}(\mathbf{X})^2+1}\right)}_{\mathcal{T}_b}, \tag{3.8}$$

where $\mathbf{A}_{ij}$ represents whether data points $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are neighbours, and $d_{ij}^2(\mathbf{X}) = \|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|^2$. We will refer to this as the **CNE objective**. The hyperparameter n_{neg} sets the number of contrastive negatives (set to be five) that affects the strength of repulsion, and n_{neigh} corresponds to the number of neighbours set for a point (fifteen in this work).

In fig. 3.4, we visualise embeddings obtained using CNE on the MNIST and transcriptomics dataset used previously. We see that the embeddings are more compact, diffuse and easier to visualise.

We will work with this objective and aim to interpret it as a likelihood, but over the latents $\mathbf{X}$. The coming subsections first interpret this objective as an approximate Bernoulli likelihood describing the distribution of the k-nearest neighbour graph as an intermediary step. Then, using these results, we interpret the objective as a Wishart likelihood over the

[17] A sketch: $\text{KL}_{\text{categorical}}(q\|p) = \sum_i q_i \log\left(\frac{q_i}{p_i}\right) \stackrel{+}{=} -\sum_i q_i \log p_i = -\sum_i a_i \log p_i = -\log \text{Categorical}(\mathbf{a}|\mathbf{p})$.

[18] We modify the presentation of the statement, and approximate certain quantities; a derivation of the form of the CNE objective we present from what appears in Damrich et al. (2022) can be found in appendix B.2.

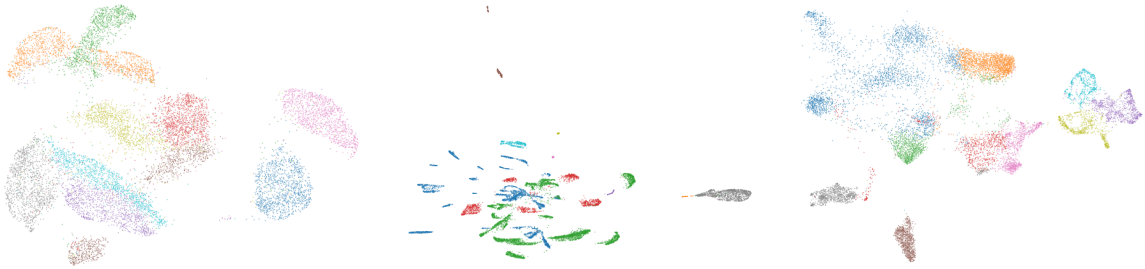


Figure 3.4: Comparison between embeddings obtained using CNE, across different datasets, showing clustering that is more diffuse than Laplacian Eigenmaps, and is easier to visualise. Tasks, left to right: clustering of MNIST digits, and cells from two transcriptomic datasets.

graph Laplacian to relate t-SNE-like methods to the earlier interpretations of ProbDR. This will therefore tie major dimensionality reduction methods as inference algorithms in one framework. Finally, to close the section, we will argue that the interpretations are semantically consistent, reinforcing the claim made in the introduction that the framework behaves correctly under certain transformations of the observed sufficient statistic.

3.3.2 Towards a distribution of the knn-graph

As an intermediary step towards the Wishart distribution that will describe t-SNE-like algorithms, in this subsection, we prove the following result.

Theorem 3.5 (Bernoulli interpretation of CNE)**.** A Bernoulli distribution over the edges of a graph, that uses a kernel to describe the probabilities as an inverse monotonic function of the latent distances, explains UMAP and t-SNE-like algorithms

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right), \tag{3.9}$$

where $\tilde{\epsilon} = 4n_{\text{neg}}n_{\text{neigh}}/3n$.

Proof. The objective we derived in eq. (3.8) is not a likelihood due to the multiplicative constant weighting the contributions of points that are not adjacent. The second term, $\mathcal{T}_b$, can be expressed as follows,

$$\begin{aligned}\mathcal{T}_b &= \tilde{\epsilon}\sum_{ij}(1-\mathbf{A}_{ij})\log\left(\left[1-\frac{1}{d_{ij}(\mathbf{X})^2+1}\right]\right)\\ &= \sum_{ij}(1-\mathbf{A}_{ij})\log\left(\left[1-\frac{1}{d_{ij}(\mathbf{X})^2+1}\right]^{\tilde{\epsilon}}\right).\end{aligned}$$

Assume a hypothetical Bernoulli distribution's likelihood over the adjacency matrix,

$$\log \text{Bernoulli}(\mathbf{A}_{ij}|\tilde{\mathbf{p}}_{ij}) = \underbrace{\mathbf{A}_{ij} \log \tilde{\mathbf{p}}_{ij}}_{\mathcal{R}_a} + \underbrace{(1 - \mathbf{A}_{ij}) \log(1 - \tilde{\mathbf{p}}_{ij})}_{\mathcal{R}_b}.$$

Setting $\mathcal{R}_b = [\mathcal{T}_b]_{ij}$, the implied probability of adjacency $\tilde{\mathbf{p}}_{ij}$ is,

$$\begin{aligned}
\tilde{\mathbf{p}}_{ij} &= 1 - \left[1 - \frac{1}{d_{ij}(\mathbf{X})^2 + 1}\right]^{\tilde{\epsilon}} \\
&= 1 - \exp\left[\tilde{\epsilon} \log\left(1 - \frac{1}{d_{ij}(\mathbf{X})^2 + 1}\right)\right] \\
&= 1 - \exp\left[-\tilde{\epsilon} \log\left(1 + \frac{1}{d_{ij}(\mathbf{X})^2}\right)\right] && 1 - \frac{1}{x+1} = \frac{x}{x+1} = 1/\left(1 + \frac{1}{x}\right) \\
&\approx 1 - 1 + \tilde{\epsilon} \log\left(1 + \frac{1}{d_{ij}(\mathbf{X})^2}\right) && \text{large n} \Rightarrow \exp(-\epsilon x) \approx 1 - \epsilon x.
\end{aligned}$$

Now, we lower bound these probabilities as follows, so that the form of our new probabilities $\mathbf{p}_{ij}$ match the first term $\mathcal{T}_a$,

$$\begin{aligned}
\tilde{\mathbf{p}}_{ij} &= \tilde{\epsilon} \log\left(1 + \frac{1}{d_{ij}(\mathbf{X})^2}\right) \\
&\geq \frac{\tilde{\epsilon}}{1 + d_{ij}(\mathbf{X})^2} := \mathbf{p}_{ij}, && \log x + 1 \geq \frac{x}{x+1} \Rightarrow \log\left(1 + \frac{1}{x}\right) \geq \frac{1}{1+x}
\end{aligned}$$

where the last identity is a commonly used log-identity. We see that the first term of the CNE objective (eq. (3.8)), $\mathcal{T}_a$, is preserved up to constants,

$$\mathcal{T}_a = \sum_{ij} \mathbf{A}_{ij} \log\left(\frac{1}{d_{ij}(\mathbf{X})^2 + 1}\right) \stackrel{+}{=} \sum_{ij} \mathbf{A}_{ij} \log\left(\frac{\tilde{\epsilon}}{d_{ij}(\mathbf{X})^2 + 1}\right) = \sum_{ij} \mathbf{A}_{ij} \log \mathbf{p}_{ij}.$$

With the choice of probabilities $\mathbf{p}_{ij}$,

$$\begin{aligned}
\mathcal{T}_b &\approx \sum_{ij} (1 - \mathbf{A}_{ij}) \log\left(1 - \tilde{\mathbf{p}}_{ij}\right) \\
&\leq \sum_{ij} (1 - \mathbf{A}_{ij}) \log\left(1 - \tilde{\epsilon} \frac{1}{1 + d_{ij}(\mathbf{X})^2}\right) \\
&= \sum_{ij} (1 - \mathbf{A}_{ij}) \log\left(1 - \mathbf{p}_{ij}\right).
\end{aligned}$$

Therefore,

$$\mathcal{E}(\mathbf{X}) \leq \sum_{ij} \mathbf{A}_{ij} \log\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) + \sum_{ij}(1-\mathbf{A}_{ij})\log\left(1-\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) + c.$$

We conclude that the CNE objective lower bounds the Bernoulli likelihood implied by the model,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\mathbf{p}_{ij} = \frac{\tilde{\epsilon}}{1+d_{ij}(\mathbf{X})^2}\right).$$

□

In words, the edge between two data points can be described by a Bernoulli distribution whose probability is inversely proportional to the latent distance between the points.[19]

In fig. 3.5, we show embeddings that use optimisation of our Bernoulli likelihood on the MNIST and transcriptomics datasets, and see that the quality of the embeddings is visually similar to the embeddings obtained by CNE.

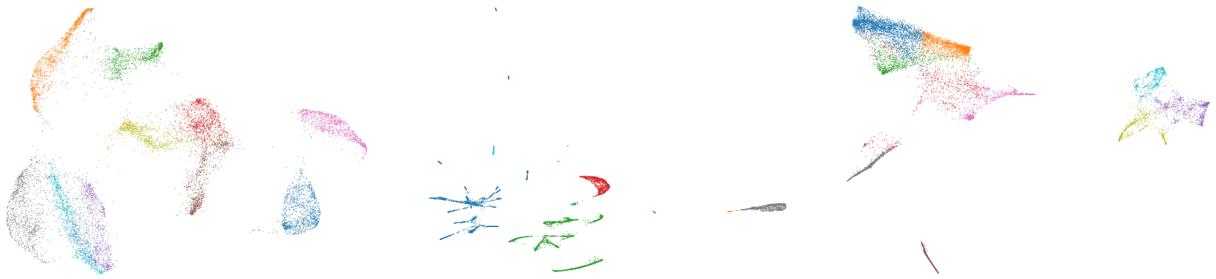


Figure 3.5: Comparison between embeddings obtained using our Bernoulli interpretation, across different datasets, showing that the quality of the embeddings is visually similar to the embeddings obtained by CNE. Tasks, left to right: clustering of MNIST digits, and cells from two transcriptomic datasets.

Our Bernoulli model for the graph adjacency is statistically meaningful. Consider the data structure—the adjacency matrix used in t-SNE and UMAP is based on a k-nearest neighbour graph, and is such that on average, n_{neigh} neighbours exist per data point. Therefore, the probability that a point is adjacent to another scales as n_{neigh}/n. Contrast this with the model implied probability,

$$\mathbb{E}_{\mathbf{X}}(\mathbf{p}_{ij}) = \frac{n_{\text{neigh}}}{n}\frac{4}{3} \cdot n_{\text{neg}}\mathbb{E}_{\mathbf{X}}\left(\frac{1}{1+d_{ij}^2}\right),$$

[19]Our Bernoulli model can be used to suggest alternative quasi-likelihood inference methods; for example, we found that optimising the quasi-likelihood $\mathcal{E} = -\sum_{ij}(\mathbf{A}_{ij} - \tilde{\epsilon}/1+d_{\mathbf{X}_{ij}^2})^2$ achieves similar visual results.

which is in line with the expected behaviour of the probability. The only difference to the expected probability of adjacency is the appearance of a kernel and the number of negatives. It may be that in full-form models, without the effect of stochasticity in mini-batch gradient descent, the number of negatives makes the model better-specified at initialisation.

To recap, we showed above how UMAP can be seen as inference in a Bernoulli model, which will be an intermediate step to deriving our Wishart interpretations over the graph Laplacian. In the coming subsection, we take a short detour from our exposition that the result above can be transformed into the ProbDR modelling statement, and answer, **how are t-SNE and UMAP different**?

3.3.2.1 From t-SNE to UMAP

In the previous subsection, we showed that CNE, interpreted using the UMAP settings, corresponds to inference within the model,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\mathbf{p}_{ij} = \frac{\tilde{\epsilon}}{1 + d_{ij}(\mathbf{X})^2}\right).$$

In this subsection, we make a short digression from showing how this fits into the ProbDR framework to show what the main difference between our probabilistic versions of t-SNE and UMAP is—the answer will be latent **scale**, which affects optimisation dynamics.

The general form of the kernel in CNE that interpolates between their versions of t-SNE and UMAP[20] is,

$$\frac{1}{1 + \tilde{s}(1 + d_{ij}(\mathbf{X})^2)} \text{ as opposed to the kernel used previously, } \frac{1}{1 + d_{ij}(\mathbf{X})^2}.$$

where $\tilde{s} = 100n_{\text{neg}}/n$ corresponding to the t-SNE setting. As we work in the large-n limit, this scale hyperparameter is small, and therefore, approximate the kernel leading to t-SNE as, $1/(1 + \tilde{s} + \tilde{s}d_{ij}^2) \approx 1/(1 + \tilde{s}d_{ij}^2)$. Replacing the distances d_{ij}^2 in the derivation above with $\tilde{s}d_{ij}^2$, we

[20]This corresponds to the "NEG" setting in CNE, as opposed to the "UMAP" setting used for the previous derivation. Although $\tilde{s} = 1$ produces a UMAP-like embedding with NEG, we do not provide an interpretation that interpolates between various settings of the "NEG" class, and provide one that interpolates between embeddings of the "UMAP" class to the "NEG" class with $\tilde{s} << 1$.

arrive at the modified interpretation,

$$\mathbf{A}_{ij} \sim \text{Bernoulli}\left(\tilde{\epsilon}\frac{1}{1+\tilde{s}d_{ij}^2(\mathbf{X})}\right), \tag{3.10}$$

noting that $\tilde{s} = 1$ leads to UMAP-like embeddings as in the last subsection, and $\tilde{s} << 1$ leads to t-SNE like embeddings.

We do not perform early exaggeration; we noticed that scaling the initialisation with the inverse of the scale parameter ($\tilde{s}$) results in qualitatively similarly embeddings as when we perform early exaggeration.

Figure 3.6 shows the resulting embeddings evolving from t-SNE to UMAP as the scale is increased. We work with the MNIST dataset, and vary the scale parameter $\tilde{s}$ from 0.05 to 10 (past the UMAP setting). We divide the initialisations by the same value, so that all runs start with the same initialisation, and optimise for a hundred epochs (the illustrated effects are stronger for lower epochs). We see that the clusters drift farther with increasing scale.

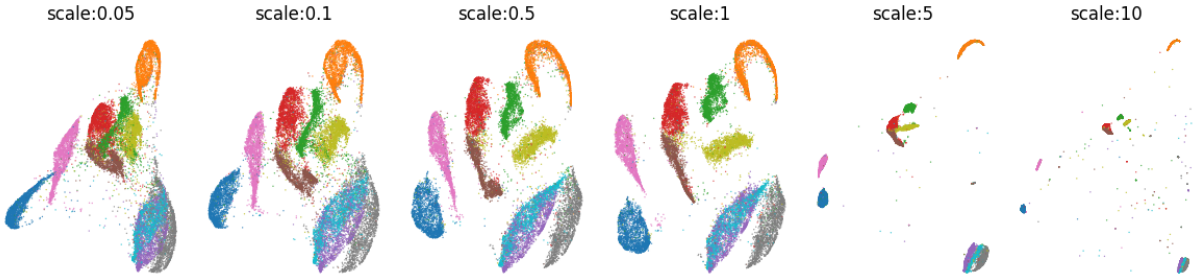


Figure 3.6: A series of embeddings of MNIST digits showing that t-SNE-like embeddings tend to UMAP-like embeddings with increasing scale ($\tilde{s} \in [0.05, 10]$) with our modified model, $\mathbf{A}_{ij} \sim \text{Bern.}(\tilde{\epsilon}/1 + \tilde{s}d_{ij}^2)$. The illustration shows that lower scale corresponds to embeddings that are more packed, and that the clusters drift apart with increasing scale.

In the model, apart from the implicit weighting of the data and regularising terms of the likelihood, the scale parameter enforces rate of convergence through implicitly lowering the learning rate of the algorithm. At smaller scales than the minimum scale visualised, there is little change from the initialisation.

This concludes our presentation of how t-SNE and UMAP differ within our Bernoulli interpretation. The interpretation of CNE as a Bernoulli distribution over edges, $\mathbf{A}_{ij} \sim \text{Bern.}(\tilde{\epsilon}/1 + d_{\mathbf{X}ij}^2)$ (eq. (3.9)), is a key result due to its simplicity. In the coming subsections however, we will go further, and show that our Bernoulli interpretation can be used to describe the approximate distribution over the graph Laplacian to enable comparability to ProbDR and GPLVMs. We

will return to our Bernoulli interpretation to show that it is statistically coherent later in the section. We now present the extension of our Bernoulli interpretation to a Wishart distribution over a graph Laplacian (therefore, a ProbDR model).

3.3.3 Describing the graph Laplacian with a Wishart distribution

In the previous subsection, we showed that UMAP corresponds approximately to inference assuming a model for the adjacencies of a nearest neighbour graph,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\mathbf{p}_{ij} = \frac{\tilde{\epsilon}}{1 + d_{ij}(\mathbf{X})^2}\right).$$

In the following subsection, we finally detail how this model can be transformed into a Wishart distribution over the graph Laplacian, to place it into the ProbDR framework.

Although the Bernoulli interpretation is a valid probabilistic interpretation of the CNE objective, we outline the following argument to make it comparable to the other models of ProbDR.[21] As Wishart distributions have supports over positive definite matrices, we will try to consider them as a model for the graph Laplacian $\mathbf{L}$ as the observed statistic.

Theorem 3.6. UMAP as non-linear ProbDR UMAP corresponds approximately to inference assuming our model for Laplacian Eigenmaps, with a simple non-linear extension to the covariance matrix using a Cauchy kernel,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((\mathbf{X}\mathbf{X}^T + 0.5\mathbf{H}\mathbf{P}^u\mathbf{H} + 0.5\tilde{\epsilon}^{-1}\mathbf{I})^{-1}, n\right), \tag{3.11}$$

where $\mathbf{P}^u_{ij} = 1/(1 + d^2_{ij}(\mathbf{X}_i, \mathbf{X}_j))$ is the unscaled Cauchy (Student-t/rational quadratic) kernel, and $\tilde{\epsilon} = 4n_{\text{neg}}n_{\text{neigh}}/3n$ as before.

Proof. The likelihood for $\mathbf{X}$ implied by the Bernoulli model, $\mathbf{A}_{ij} \sim \text{Bern.}(\tilde{\epsilon}/1+d_{ij}(\mathbf{X})^2)$ (eq. (3.9))

[21] This can be done as exponential families share similar likelihood forms, and a Wishart interpretation, despite being over discrete matrices, may correspond to a similar misspecification as performing classification using linear regression (i.e. using the L^2 norm separator).

is,

$$\begin{aligned}
\log p(\mathbf{A}|\mathbf{X}) &= \sum_{ij} \mathbf{A}_{ij} \log\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) + \sum_{ij}(1-\mathbf{A}_{ij})\log\left(1-\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) \\
&\approx \sum_{ij} \mathbf{A}_{ij} \log\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) + \sum_{ij}\log\left(1-\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) && \text{large n} \\
&\approx \sum_{ij} \mathbf{A}_{ij} \log\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) - \sum_{ij}\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2} && \text{small } \tilde{\epsilon} \quad (3.12)
\end{aligned}$$

Equation (3.12) is a Poisson likelihood,

$$\log p(\mathbf{A}|\mathbf{X}) = \sum_{ij} \log \text{Poisson}\left(\mathbf{A}_{ij}\middle|\mu = \tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right). \tag{3.13}$$

This likelihood is noteworthy as it is a minimal form (an ablation) of the CNE objective, as it only keeps terms that are crucial for the optimisation—the adjacency attraction term and a diffusing regularisation term that's a non-linear function of the distances.

In passing, we illustrate in fig. 3.7 that optimising embeddings using just the regularising term ($\mathcal{L} = \sum_{ij} 1/(1+d_{ij}^2(\mathbf{X}))$) leads to a purely diffusive behaviour.

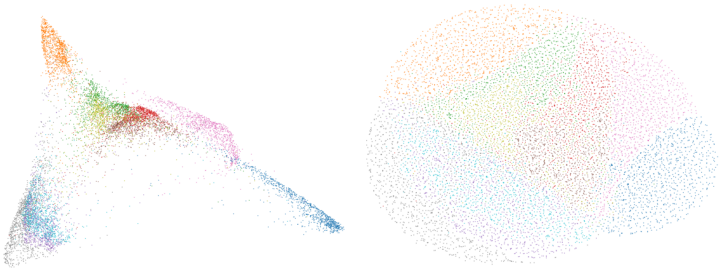


Figure 3.7: Embeddings of the MNIST data using Laplacian Eigenmaps (left) used for initialisation and using just the repulsive diffusion term (right) as part of optimisation. This shows that the action of the second regularising term is mainly diffusive.

Going back to the objective,

$$\log p(\mathbf{A}|\mathbf{X}) = \sum_{ij} \mathbf{A}_{ij} \log\left(\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2}\right) - \sum_{ij}\tilde{\epsilon}\frac{1}{1+d_{ij}(\mathbf{X})^2},$$

define for convenience,

$$\underbrace{\mathbf{P}^u_{ij} = 1/(1+d^2_{ij})}_{\text{(unscaled)}}, \quad \underbrace{\mathbf{P}^s_{ij} = \tilde{\epsilon}\mathbf{P}^u_{ij}}_{\text{(scaled)}}, \quad \underbrace{\mathbf{P}^{ls}_{ij} = \log \mathbf{P}^s_{ij}}_{\text{(log-scaled)}}.$$

Then the objective simplifies as[22],

$$\begin{aligned}
\log p(\mathbf{A}|\mathbf{X}) &\approx \operatorname{tr}(\mathbf{A}\mathbf{P}^{ls}) - \sum_{ij} \tilde{\epsilon} \frac{1}{1 + d_{ij}(\mathbf{X})^2} && \operatorname{tr}(AB) = \sum_{ij} A_{ij} B_{ji} \\
&\stackrel{+}{=} \operatorname{tr}((\mathbf{A} - \mathbf{D})\mathbf{P}^{ls}) - \sum_{ij} \tilde{\epsilon} \frac{1}{1 + d_{ij}(\mathbf{X})^2} && \operatorname{tr}(\mathbf{D}\mathbf{P}^{ls}) = \sum_{ij} \mathbf{D}_{ij} \mathbf{P}^{ls}_{ij} = \underbrace{\sum_{i} \mathbf{D}_{ii} \log \tilde{\epsilon}}_{\text{const.}}
\end{aligned}$$

and we can now define the likelihood in terms of the graph Laplacian $\mathbf{L}$,

$$\Rightarrow \log p(\mathbf{L}|\mathbf{X}) = -\operatorname{tr}(\mathbf{L}\mathbf{H}\mathbf{P}^{ls}\mathbf{H}) - \sum_{ij} \mathbf{P}^{s}_{ij}, \qquad \mathbf{H}\mathbf{L}\mathbf{H} = \mathbf{L}.$$

We simplify further; firstly,

$$\begin{aligned}
\operatorname{tr}(\mathbf{H}\mathbf{P}^{s}\mathbf{H}) &= \operatorname{tr}\left(\left(\mathbf{P}^{s} - \frac{1}{n}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s}\right)\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^{T}\right)\right) \\
&= \operatorname{tr}\left(\mathbf{P}^{s} - \frac{2}{n}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s} + \frac{1}{n^2}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s}\mathbf{1}\mathbf{1}^{T}\right) && \text{tr cyclic} \\
&= \operatorname{tr}\left(\mathbf{P}^{s} - \frac{1}{n}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s}\right) && \operatorname{tr}\left(\frac{1}{n^2}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s}\mathbf{1}\mathbf{1}^{T}\right) = \operatorname{tr}\left(\frac{1}{n}\mathbf{1}\mathbf{1}^{T}\mathbf{P}^{s}\right) \\
&\stackrel{+}{=} -\sum_{ij} \mathbf{P}^{s}_{ij}/n. && \operatorname{tr}(\mathbf{P}^{s}) = n\tilde{\epsilon} \qquad (3.14)
\end{aligned}$$

Secondly,

$$\operatorname{tr}(\mathbf{P}^{s}) = \log\det\operatorname{expm}\mathbf{P}^{s} \approx \log\det(\mathbf{I} + \mathbf{P}^{s}). \qquad \mathbf{P}^{s} \approx \mathbf{0} \qquad (3.15)$$

We can now simplify the previously found likelihood as,

$$\begin{aligned}
\log p(\mathbf{L}|\mathbf{X}) &\stackrel{+}{=} -\operatorname{tr}(\mathbf{L}\mathbf{H}\mathbf{P}^{ls}\mathbf{H}) + n\operatorname{tr}(\mathbf{H}\mathbf{P}\mathbf{H}) + k && \text{using eq. (3.14)} \\
&\approx -\operatorname{tr}(\mathbf{L}\mathbf{H}\mathbf{P}^{ls}\mathbf{H}) + n\log|\mathbf{I} + \mathbf{H}\mathbf{P}\mathbf{H}| + k. && \text{using eq. (3.15); small } \tilde{\epsilon}
\end{aligned}$$

[22] Our derivation also produces a similar objective to the DK-LLE objective of Draganov and Dohn (2023) (Lemma 4), which has gradients that are similar to those of UMAP.

The matrix $\mathbf{HP}^{ls}\mathbf{H}$ is PSD.[23][24] We will now approximate $\log \mathbf{P}^s_{ij}$ within the vicinity[25] of $\tilde{\epsilon}$ by matching the gradient and function value using an ansatz,

$$\log x \approx ax + \frac{b}{x} + c \tag{3.16}$$
$$\text{eq. (3.16)} \implies \log \tilde{\epsilon} = a\tilde{\epsilon} + \frac{b}{\tilde{\epsilon}} + c,\ d\cdot/dx|_{\tilde{\epsilon}} \implies \frac{1}{\tilde{\epsilon}} = a - \frac{b}{\tilde{\epsilon}^2},\ d^2\cdot/dx^2|_{\tilde{\epsilon}} \implies -\frac{1}{\tilde{\epsilon}^2} = \frac{2b}{\tilde{\epsilon}^3}$$
$$\therefore b = -\frac{\tilde{\epsilon}}{2},\ a = \frac{1}{2\tilde{\epsilon}},\ c = \log\tilde{\epsilon}.$$

We use this ansatz because it results in a familiar kernel form;

$$\log \mathbf{P}^s_{ij} \overset{+}{\approx} \frac{\mathbf{P}^u_{ij}}{2} - \frac{1}{2\mathbf{P}^u_{ij}} \overset{+}{=} \frac{1}{2}\left(\mathbf{P}^u_{ij} - \|\mathbf{X}_i - \mathbf{X}_j\|^2\right),$$

and we use the well-known result $-0.5\mathbf{HD}^2\mathbf{H} = \mathbf{XX}^T$ (assuming/enforcing centered $\mathbf{X}$) to get,

$$\mathbf{HP}^{ls}\mathbf{H} \overset{+}{\approx} \frac{1}{2}\mathbf{HP}^u\mathbf{H} + \mathbf{XX}^T.$$

This leads finally to the Wishart likelihood over the graph Laplacian,

$$\begin{aligned}
\implies \log p(\mathbf{L}|\mathbf{X}) &\overset{+}{\approx} -\text{tr}(\mathbf{L}(0.5\mathbf{HP}^u\mathbf{H} + \mathbf{XX}^T)) + n\log|\mathbf{I} + \mathbf{HPH}| \\
&\overset{+}{=} -\text{tr}(\mathbf{L}(0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{HP}^u\mathbf{H} + \mathbf{XX}^T)) + n\log|0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{HP}^u\mathbf{H}| \\
&\leq \log \mathcal{W}(\mathbf{L}|(0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{HP}^u\mathbf{H} + \mathbf{XX}^T)^{-1}, n). \qquad \log|A+B| \geq \log|A|
\end{aligned}$$

Therefore, the Wishart model that underpins Laplacian Eigenmaps, extended with a non-linear kernel, approximates inference UMAP,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{HP}^u\mathbf{H} + \mathbf{XX}^T)^{-1}, n\right). \tag{3.11}$$

This is a restatement of eq. (3.11), completing our proof. □

This concludes the major result of this section, showing that t-SNE-like algorithms fit into the

[23] $\mathbf{P}^{ls}$ is conditionally positive semidefinite (CPSD; double centering a CPSD matrix makes it PSD) as $\mathbf{P}_{ij}$ defines a kernel, which enforces positive definiteness and an element-wise log of a PD matrix with all positive elements (and infinite divisibility) will at least be conditionally positive definite (Ex. 5.6.15, Bhatia (2007)).

[24] When are element-wise functions of PSD matrices PSD? Due to the results of Schur, any polynomial with non-negative coefficients will result in the retention of positivity. Schoenberg's work showed converse results.

[25] This neighbourhood was chosen as $\tilde{\epsilon}$ is the maximal value of $\mathbf{P}^s_{ij}$.

ProbDR model form.

The result above provides a direct connection to the model behind Laplacian Eigenmaps as our model for UMAP is a non-linear extension of the previous section's Laplacian Eigenmaps interpretation. The Wishart model of eq. (3.11) also connects Gaussian processes to ProbDR, as the model implies that the implicit data covariance is modelled by a non-linear Gaussian process covariance function.

The implied covariance of our model (the inverse of the Wishart's centrality parameter) is non-stationary and can be justified by the fact that the adjacency probabilities of t-SNE/UMAP-like algorithms approach zero as a function of distance, unlike what would be expected from stationary Gaussian process-like generative models.

Our Wishart model uses a covariance based on a double-centred non-linear kernel, differing from the linear kernel used in the previous section. Although we retain the double centring as a result of our manipulations, removing the double centring does not lead to any significant changes to the visualised embeddings.

Figure 3.8 illustrates embeddings of MNIST and transcriptomics datasets obtained using our Wishart interpretation. This is the main experimental validation of our interpretations: we see a t-SNE-like clustering when inference is performed using our probabilistic model. We see that the embeddings found are more diffuse than the embeddings found previously using CNE. The Wishart interpretations, though based on CNE's version of UMAP, are qualitatively similar to CNE's version of t-SNE, illustrated side-by-side in fig. 3.9.[26]

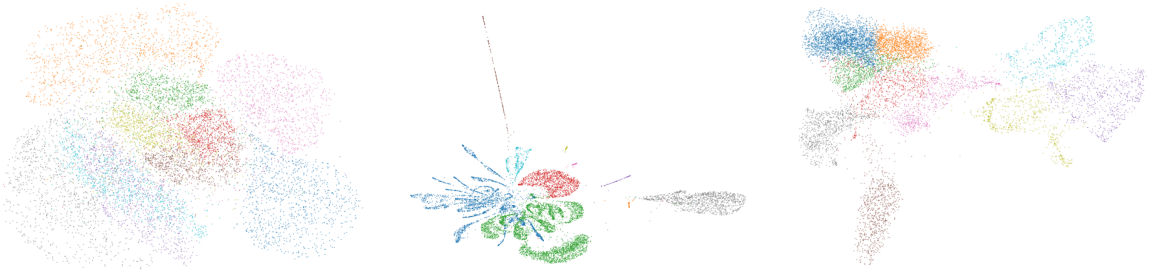


Figure 3.8: Comparison between embeddings obtained using our Wishart interpretations, across the MNIST and transcriptomics datasets as before. The plots show that embeddings obtained using our Wishart interpretations are more diffuse than embeddings corresponding to CNE with UMAP settings (and are more t-SNE-like than UMAP-like).

Although the kernel that appears in our derivations involves the double centered kernel,

[26]We posit that due to the relative coarseness of the approximations made to derive the Wishart interpretations, the differences between CNE's t-SNE and UMAP interpretations are lost, and the resulting behaviour of our Wishart model is difficult to analyse within the CNE framework.

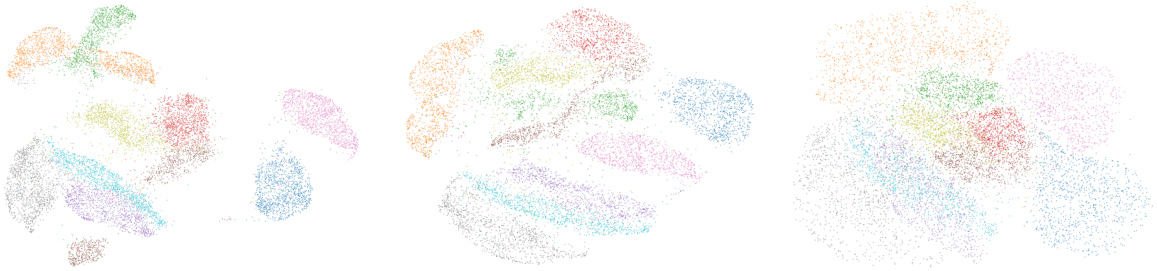


Figure 3.9: An illustration showing that embeddings obtained our Wishart likelihood (right) are more comparable to CNE's t-SNE settings (center) than its UMAP settings (left).

removing the kernel did not affect visualisations. Figure 3.10 shows MNIST embeddings recovered with and without centering, showing no noticeable differences.

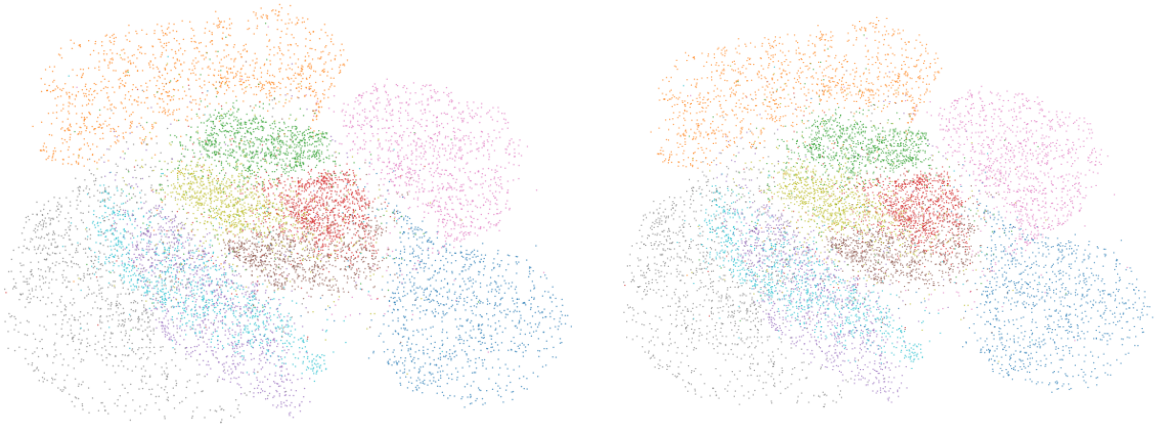


Figure 3.10: MNIST embeddings obtained using our Wishart interpretation, (left) with double centering of the kernel and (right) without, showing no visual differences, suggesting that the double centering is safe to drop.

For purposes of illustration, we show MNIST embeddings obtained by dp-PCA and GPLVM in fig. 3.11, which are visually distinct from all nearest-neighbour based methods discussed thus far, presumably due to the graph Laplacian encoding different data statistics than the sample covariance that GPLVM and dp-PCA are based on.[27]

In appendix B.2, we show that the assumptions that are read off from our Wishart statements are valid (and introduce a novel approximation to a GP precision in the process). Specifically, $\mathbb{E}(\mathbf{L}_{ij}|\mathbf{X}) \sim -1/n$, i.e. the expected value of the off-diagonal elements of the graph Laplacian under our Wishart model scales inversely with the number of data, as expected from the data structure. In appendix B.2, we also note that, using our interpretation, the implied data

[27] Our GPLVM experiments were run using a `linear + constant + t + noise` kernels to match our Wishart interpretations, using a dp-PCA initialisation. In each case, the GPLVM hyperparameters were first "pre-trained" using the for 10 epochs, and the embeddings were trained for a further 40, matching the optimisation that was used in **Ravuri et al. (2022b)**.

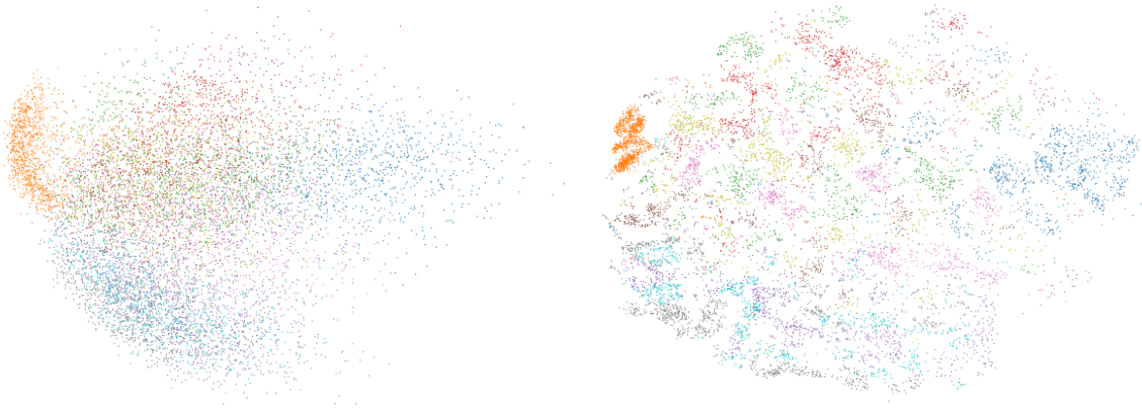


Figure 3.11: An illustration showing that MNIST embeddings found using dp-PCA (left) and GPLVM (right) are visually distinct to the other nearest neighbour-based methods of dimensionality reduction presented thus far.

covariance is described by an inverse-Wishart distribution,

$$\mathbf{L}^{+} \sim \mathcal{W}^{-1}(\mathbf{X}\mathbf{X}^{T} + \mathbf{H}K(\mathbf{X}, \mathbf{X})\mathbf{H} + 0.5\tilde{\epsilon}^{-1}, n),$$

and consider if there is a Wishart distribution that can be placed on the covariance, as in the case of a GPLVM,

$$\mathbf{L}^{+} \sim \mathcal{W}(.^{?}.).$$

This turns out to be non-trivial, as the model misspecification makes it so that the inverse of the statistic and the inverse of the model behave in different ways. We observe, for the first time in our interpretations, that simple approaches to translate the model via moment-matching fail, and that model specification should follow what we know to be true about the data structure.

This concludes our dialogue on our interpretations for t-SNE-like algorithms. We have shown that a simple Bernoulli model for graph adjacencies, and a Wishart model for the graph Laplacian underpins t-SNE-like algorithms. We show that, visually, they recover similar embeddings to t-SNE-like algorithms, and we have shown that they correspond to valid statistical assumptions about the data structure (and hence are models that arise naturally when we try to model these data structures). Before we close the section, we show next what our interpretations bring: a potential for finding analytical inference algorithms.

3.3.4 Efficient inference ideas using non-linear ProbDR

In the previous subsection, we showed that UMAP and t-SNE-like algorithms correspond to maximum-likelihood assuming a model for the adjacencies given latents $\mathbf{X}$,

$$\mathbf{A}_{ij} \sim \text{Bernoulli}\left(\frac{\tilde{\epsilon}}{1 + d_{ij}^2(\mathbf{X}_i, \mathbf{X}_j)}\right),$$

where $\tilde{\epsilon} = 4n_{\text{neg}}n_{\text{neigh}}/3n$, and a model for a graph Laplacian that is approximately equivalent,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{H}\mathbf{P}^u\mathbf{H} + \mathbf{X}\mathbf{X}^T)^{-1}, n\right).$$

We conclude this section by describing ideas on how our framework can be used to suggest efficient inference within our unified model class. This will reveal more connections between different interpretations of ProbDR.

We first look for solutions within the eigenbasis obtained by the graph Laplacian, based on visual evidence. We then explore the usage of efficient inference algorithms suggested by Levy and Goldberg (2014) within SGNS/word2vec contexts that can be applied directly to our model class. This idea will connect our linear and non-linear ProbDR models, and show how non-linearity from the "model side" can be translated to a non-linearity on the "statistic side". Overall, the subsection shows how our framework suggests new ideas for building efficient algorithms.

3.3.4.1 Parametric t-SNE using the graph Laplacian eigenbasis

Here, we argue that embeddings that share visual characteristics of t-SNE-like algorithms can be found using scaled minor eigenvectors of the graph Laplacian. This enables a form of parametric inference by making data points conditionally independent given parameters that we introduce.

Recall the log-likelihood implied by our Wishart interpretation (eq. 3.11); an optimum is

achieved if the modelled precision (that uses the latents $\mathbf{X}$) is equal to $\mathbf{L}/n$[28],

$$\begin{aligned} \mathcal{E} &= -\text{tr}(\mathbf{L}\Sigma) + n \log\det(\Sigma) \\ \Rightarrow \frac{d\mathcal{E}}{d\Sigma} &= -\mathbf{L} + n\Sigma^{-1} = 0 \qquad \text{at optimum} \\ \Rightarrow \mathbf{L} &= n\Sigma^{-1} \end{aligned}$$

Therefore, near an optimum of the log-likelihood, we expect that the graph Laplacian $\mathbf{L}$ and the kernel $k(\mathbf{X}_i, \mathbf{X}_j) = \langle \mathbf{X}_i, \mathbf{X}_j \rangle + (1 + \|\mathbf{X}_i - \mathbf{X}_j\|^2)^{-1}$ share eigenvectors. Typically, for kernels like the RBF that are smoothly decreasing as a function of distance and stationary, the kernel eigenfunctions are organised in frequency[29]. Assume that the addition of the linear kernel simply increases the magnitude of the initial low-frequency modes of the Cauchy kernel. Therefore, the eigenvectors of the graph Laplacian must be either linear in the latents, or are low-frequency transforms of the latents.

Although the argument for this claim is coarse, fig. 3.12 illustrates that, using the embedding found with our Wishart interpretation, the eigenvectors of the graph Laplacian are smooth functions (with some of them that are mostly linear) of the solution found by optimising the likelihood, *and vice versa.*

Therefore, we may attempt to parameterise the embedding using a non-linear function, e.g. a neural network on the eigenvectors of $\mathbf{L}$, and optimise just the parameters of this function using the Wishart likelihood (or the Bernoulli likelihood for efficiency, as the likelihood factorises by data point). Figure 3.13 illustrates embeddings found when the embeddings are parameterised as a **linear function** of the graph Laplacian eigenvectors (the Laplacian Eigenmaps initialisation).[30], showing that the visual quality of the embeddings found is similar to the embeddings found using our Wishart interpretation. Future work can explore whether such a solution can be found without direct optimisation of the objective.[31]

Next, we will briefly show how ideas of efficient inference for word2vec/SGNS can be applied to our models, linking our linear and non-linear interpretations.

[28] As is the case with exponential families, this corresponds to a moment matching.

[29] For RBF kernels, $\phi_k(x) \propto \exp(x^2/4\sigma^2) \cdot \exp(-cx^2) H_k(\sqrt{2c}x)$ (Rasmussen and Williams, 2005), where the second part behaves as $\cos(2\sqrt{ck}x - k\pi/2)$ (DLMF: §18.15(v)).

[30] We use just eight dimensions of the GL, leading to just eight parameters that are optimised.

[31] Although this approach "uses the data twice", the approach can be justified as variational inference.

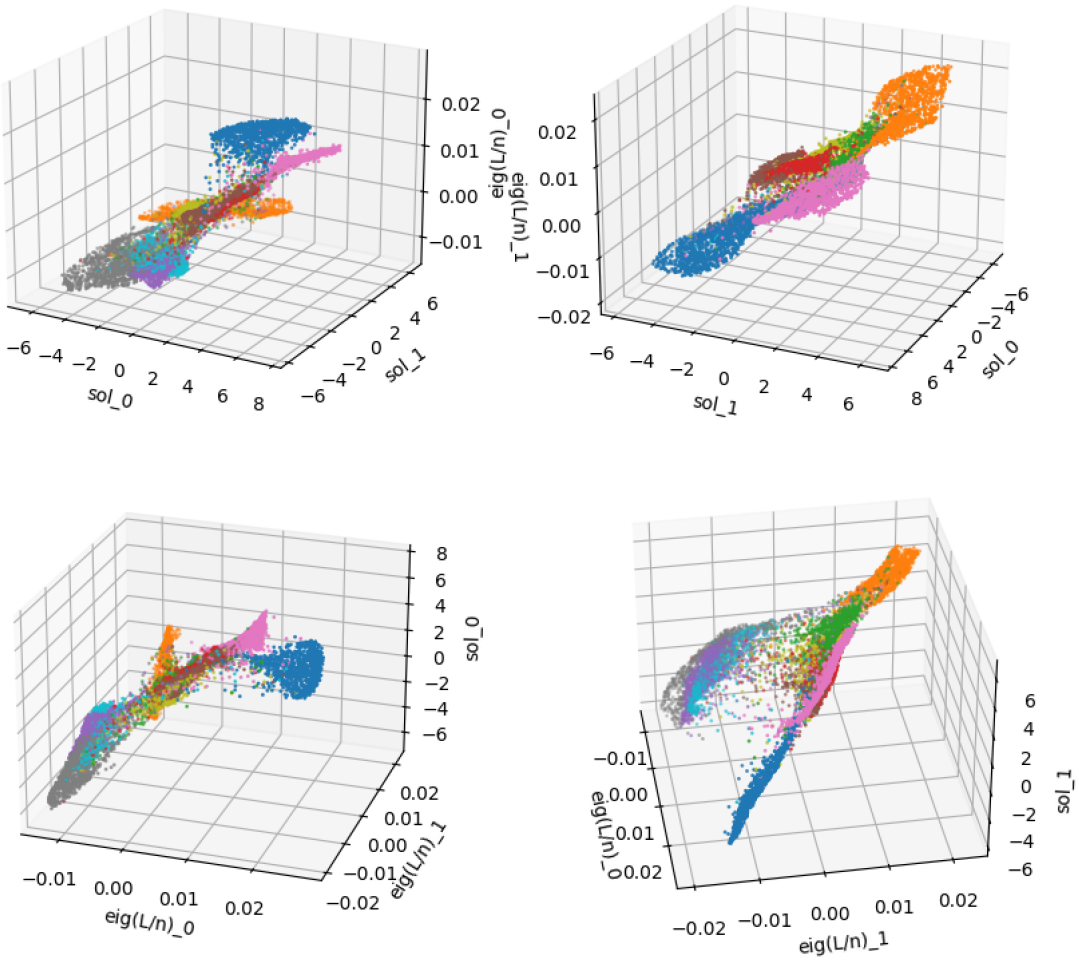


Figure 3.12: Illustrations showing the two dimensions of the non-linear ProbDR solution found using our Wishart interpretation (eq. (3.11)) as a function of the first two eigenvectors of the graph Laplacian **L** (bottom), and the two minor graph Laplacian eigenvectors as functions of the solution (top). These plots show that optimised embeddings are smooth functions of the GL eigenvectors and vice versa.

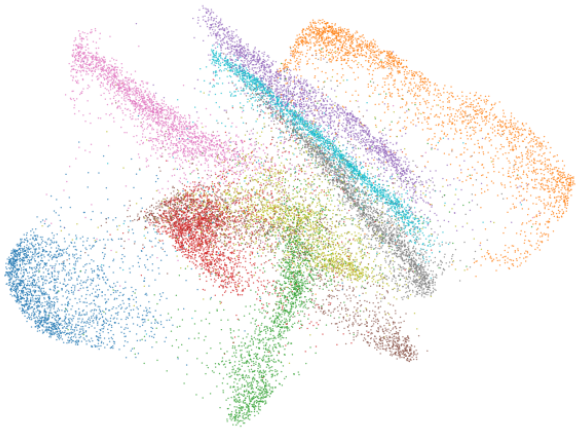


Figure 3.13: MNIST embeddings found by parameterising our Wishart interpretation solutions as a linear function of the graph Laplacian, showing a similar clustering quality as our full-form Wishart models.

3.3.4.2 Approximate Bernoulli inference using SVD

Below, we borrow the main idea of Levy and Goldberg (2014), who show that approximate inference within word2vec models can be done using singular value decomposition. The idea shows how element-wise functions of our matrix-valued statistics arise within inference contexts (and bear resemblance to kernel-PCA). However, we note that further refinement of the ideas is needed for large-scale use.

Within SGNS contexts, the work above uses a similar GLM-type model as our Bernoulli interpretation, which leads to a loss of the form $\sum_{ij} \mathcal{L}_{ij}$. They then consider, what is the stationarity condition implied by a single data point pair? Setting $d\mathcal{L}_{ij}/dp_{ij} = 0$, where p_{ij} is their probability of adjacency, they find that the stationarity condition implies,

$$\mathbf{w}_i \mathbf{c}_j^T = \mathbf{M}_{ij},$$

where $\mathbf{w}_i, \mathbf{c}_j$ are latent variables representing word and context embeddings. They then argue that such a problem is naturally solved using SVD of the matrix $\mathbf{M}$.

We will follow a similar argument. Consider the CNE objective with a minor modification to the kernel,

$$\mathbf{p}_{ij} = 2/(1 + \exp(\mathbf{D}_{ij}^2)),$$

as such a parameterisation will introduce the log function, making our estimator better behaved. This function has a similar behaviour to the Cauchy-kernel in that the probability decreases monotonically with distance (between the latents) and attains a maximal value of one when the distance is zero. Consider the objective with respect to a single pair of data points i and j,

$$\begin{aligned}
\mathcal{E}_{ij} = \mathbf{A}_{ij} \log \mathbf{p}_{ij} + \tilde{\epsilon} \log(1 - \mathbf{p}_{ij}) \implies \frac{\partial \mathcal{E}_{ij}}{\partial \mathbf{p}_{ij}} = \frac{\mathbf{A}_{ij}}{\mathbf{p}_{ij}} - \frac{\tilde{\epsilon}}{1 - \mathbf{p}_{ij}} = 0 \qquad \text{at optimum} \\
\implies \frac{\mathbf{p}_{ij}}{1 - \mathbf{p}_{ij}} = \frac{\mathbf{A}_{ij}}{\tilde{\epsilon}} \implies \mathbf{p}_{ij} = \frac{\mathbf{A}_{ij}}{\mathbf{A}_{ij} + \tilde{\epsilon}} \\
\implies \mathbf{D}_{ij}^2 = -\text{logit} \frac{0.5\mathbf{A}_{ij}}{\mathbf{A}_{ij} + \tilde{\epsilon}} = \log\left(1 + \frac{2\tilde{\epsilon}}{\mathbf{A}_{ij}}\right)
\end{aligned}$$

The distance matrix $\mathbf{D}^2 = \log(1 + 2\tilde{\epsilon}/\mathbf{A})$, implies a corresponding inner product matrix,

$$\mathbf{X}\mathbf{X}^T = -0.5\mathbf{H}\mathbf{D}^2\mathbf{H} = -0.5\mathbf{H} \log\left(1 + \frac{2\tilde{\epsilon}}{\mathbf{A}}\right)\mathbf{H},$$

and we our embedding through the eigendecomposition of the RHS.

In practice, the element-wise inverse of $\mathbf{A}$ is ill-posed. In appendix A.14, we attempt a naive fix to the problem, and we show that embeddings from such an approach are indeed more diffuse than Laplacian Eigenmaps. It is the case however, that further refinement of these ideas is necessary to recover embeddings of a visual quality compared to those corresponding to CNE.

Nevertheless, our arguments in this section show that there exists a rough equivalence between linear and non-linear ProbDR models. The interpretation of the algorithm above with a linear ProbDR model follows,

$$\nu\hat{\mathbf{S}}|\mathbf{X} \sim \mathcal{W}(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, \nu).$$

with an elementwise function applied to the data,

$$\hat{\mathbf{S}} = -0.5\mathbf{H}\log(1 + 2\tilde{\epsilon}/\mathbf{A})\mathbf{H},$$

as algorithms involving an eigendecomposition can be interpreted in this manner. The non-linear ProbDR models that interpret t-SNE-like algorithms use a **non-linear kernel** but a straightforward estimate of the data statistic. Future work can explore whether more accurate approximations exist for obtaining CNE embeddings analytically.

This concludes our presentation of our interpretations of UMAP and t-SNE-like algorithms. We showed that UMAP ($\tilde{s} = 1$) and t-SNE ($\tilde{s} = 100n_{\text{neg}}/n << 1$) correspond to inference assuming the model for nearest neighbour adjacencies,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\frac{\tilde{\epsilon}}{1 + \tilde{s}d_{ij}^2(\mathbf{X})}\right).$$

We then show that this model can be approximated by a Wishart distribution on the corresponding graph Laplacian,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{H}\mathbf{P}^u\mathbf{H} + \mathbf{X}\mathbf{X}^T)^{-1}, n\right),$$

where $\tilde{\epsilon} = 4n_{\text{neg}}n_{\text{neigh}}/3n$. We showed that the model make valid assumptions given the data statistic, and that they recover visually similar embeddings to those obtained by CNE (the

simplification of UMAP/t-SNE that laid the foundations to the section). We also briefly explore ideas on efficient inference suggested by the framework, specifically that the solution may be found in the eigenbasis of the graph Laplacian and that SGNS-style approximate inference can be done in our framework. We leave a complete exploration of the ideas, and an exploration of natural gradient descent in our framework[32] to future work.

In the last section of this chapter, we present variational interpretations for ProbDR, which will enable comparisons to wider representation learning, specifically SSL methods and transformers, presented as chapter 4.

3.4 Variational interpretations

In this final section, we show that all methods with maximum a-posteriori interpretations presented so far have a dual KL-minimisation view, i.e. that the MAP inference described so far can also be framed as a KL-minimisation in a variational framework. This view will be useful for chapter 4, to draw connections between ProbDR and wider representation learning.

We demonstrate in this section that the graph presented in fig. 3.14, showing a model estimating a statistic (covariance/adjacency or precision matrix) of interest and a static variational constraint on the same statistic that uses the data, explains all methods discussed thus far. This alternative presentation enables these methods to be used in variational frameworks such as optimal transport and in frameworks where inference seems to be circular, as explored in the next chapter (chapter 4) to explain the behaviour of transformers and SSL methods.

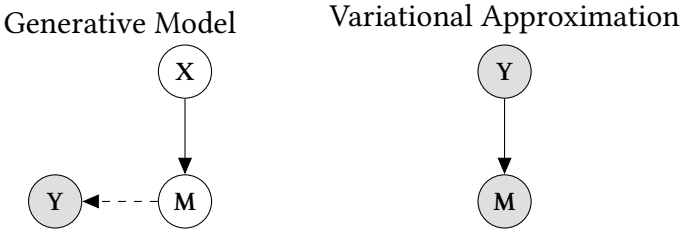


Figure 3.14: A simplified graphical model that summarises the ProbDR class of models under a variational view. The framework estimates a statistic **M**, which corresponds to a covariance or precision matrix. This is explained using the latents **X**, and a variational constraint is put on **M** using the data **Y**. The dotted line from **M** to **Y** is an optional inclusion to make the model generative for the data, that does not contribute any terms to objectives with respect to the latents **X**.

Concretely, ProbDR can be presented as a variational framework in which low dimensional latents **X** describe a statistic of the data **M** (e.g., a covariance), which **optionally** can be used

[32] As just one step of ordinary gradient descent produces well-defined clusters using our Poisson interpretation of eq. (3.13).

to construct a generative model on the data $\mathbf{Y}$. The moment $\mathbf{M}$ has a variational distribution associated with it, that uses the data $\mathbf{Y}$.

Inference in the framework is done by maximising a lower bound on the evidence (and the likelihood), the evidence lower bound (ELBO, Jordan et al. (1999); Blei et al. (2017)), with respect to $\mathbf{X}$ and model parameters,

$$\arg\max_{\mathbf{X},\theta} \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}[\log p_\theta(\mathbf{Y}|\mathbf{M})] - \mathrm{KL}(q(\mathbf{M}|\mathbf{Y})||p(\mathbf{M}|\mathbf{X})). \tag{3.17}$$

The ELBO is derived as follows.

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{M}|\mathbf{Y})||p_\theta(\mathbf{M}|\mathbf{X},\mathbf{Y})) &= \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}\left[\log \frac{q(\mathbf{M}|\mathbf{Y})}{p_\theta(\mathbf{M}|\mathbf{X},\mathbf{Y})}\right] \\
&= \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}[\log q(\mathbf{M}|\mathbf{Y})] - \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}[\log p_\theta(\mathbf{Y}|\mathbf{M})p(\mathbf{M}|\mathbf{X})] + \log p(\mathbf{Y}|\mathbf{X}) \\
&= \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}\left[\log \frac{q(\mathbf{M}|\mathbf{Y})}{p(\mathbf{M}|\mathbf{X})}\right] - \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}[\log p_\theta(\mathbf{Y}|\mathbf{M})] + \log p(\mathbf{Y}|\mathbf{X}) \\
&= \mathrm{KL}(q(\mathbf{M}|\mathbf{Y})\|p(\mathbf{M}|\mathbf{X})) - \mathbb{E}_{q(\mathbf{M}|\mathbf{Y})}[\log p_\theta(\mathbf{Y}|\mathbf{M})] + \log p(\mathbf{Y}|\mathbf{X}) \\
&= \log p(\mathbf{Y}|\mathbf{X}) - \mathrm{ELBO}(\mathbf{X},\theta) \geq 0 \\
\implies \mathrm{ELBO}(\mathbf{X},\theta) &\leq \log p(\mathbf{Y}|\mathbf{X}),
\end{aligned}$$

which shows that the function we derive lower bounds the model evidence, and in a similar fashion to variational inference, we seek to maximise it, with respect to latent variables that describe $\mathbf{M}$, $\mathbf{X}$, and the model parameters θ that describe the data $\mathbf{Y}$ given the statistic $\mathbf{M}$. As before, we have assumed an improper uniform prior over $\mathbf{X}$, i.e. $p(\mathbf{X}) \propto 1$.

Optimising the ELBO with respect to $\mathbf{X}$ leads to the minimisation problem becoming,

$$\arg\max_{\mathbf{X}} \mathrm{ELBO}(\mathbf{X},\theta) = \arg\min_{\mathbf{X}} \mathrm{KL}(q(\mathbf{M}|\mathbf{Y})||p(\mathbf{M}|\mathbf{X})), \tag{3.18}$$

as the data fit term of the ELBO (the first term in eq. (3.17)) is independent of $\mathbf{X}$ and the KL above is independent of θ.

In our framework, the variational distribution q does not have any parameters that are optimised, much like the case of denoising diffusion models (Ho et al., 2020), and unlike traditional variational inference (Blei et al., 2017) used by model frameworks such as VAEs and "back-constrained" GPLVMs (Bui and Turner, 2015; Lawrence and Quiñonero Candela, 2006).

The objective above has two terms. The second term (the KL divergence) corresponds to the objective/cost function that is minimised in each of the respective DR algorithms. The first term, $\log p_\theta(\mathbf{Y}|\mathbf{M})$, corresponds to the generative model applied using the moment $\mathbf{M}$ on data $\mathbf{Y}$ and has no dependence on latents $\mathbf{X}$. Therefore, the generative model is a "free" addition, as its presence adds a constant to the objective with respect to the latents.[33]

In the coming subsections, we show how the Wishart models of ProbDR and additionally, t-SNE-like algorithms in their original formulation, which are **defined** as KL-minimising algorithms, fit into our variational framework.

3.4.1 Explaining the Wishart cases

In this subsection, to establish a connection to ProbDR, we show that the 2-step process of estimating a covariance/precision-like matrix $\hat{\mathbf{G}}$ using the data $\mathbf{Y}$, and performing MAP estimation for latents in Wishart models of the form used with our dp-PCA, dp-MCA and our non-linear Wishart models that explain t-SNE-like algorithms models fit into our variational view of ProbDR.

Theorem 3.7 (Variational ProbDR and MAP Equivalence: Wishart Cases)**.** The maximum a-posteriori estimate for $\mathbf{X}$ in our Wishart model after first estimating a covariance/precision-like matrix $\hat{\mathbf{G}}$, i.e.,

$$\arg\max_{\mathbf{X}} \log p(\nu * \hat{\mathbf{G}}|\mathbf{X}) \text{ assuming } p(\mathbf{M}|g(\mathbf{X})) = \mathcal{W}(\mathbf{M}|g(\mathbf{X}), \nu)$$

with an improper uniform prior on the latents $\mathbf{X}$, is equivalent to minimising the ProbDR KL-divergence (eq. 3.17), with the same model, and with a variational constraint that uses the data statistic as its sufficient statistic,

$$\arg\min_{\mathbf{X}} \mathrm{KL}(q(\mathbf{M}|\hat{\mathbf{G}})||p(\mathbf{M}|g(\mathbf{X}))).$$

[33] This shows how a two-step process that first obtains latents and then uses them as part of a regression arises, as the inference for θ follows after the inference of $\mathbf{X}$. An example of such a regress-after-DR algorithm is principal component regression, where a dimensionality reduction is first done on the covariates, and a regression then performed using simply the top principal components in the data.

Written using model notation, the variational setup is,

$$\text{model (law of p)} : \mathbf{M}|g(\mathbf{X}) \sim \mathcal{W}(g(\mathbf{X}), v),$$
$$\text{variational approx (law of q)} : \mathbf{M}|\hat{\mathbf{G}}(\mathbf{Y}) \sim \mathcal{W}(\hat{\mathbf{G}}(\mathbf{Y}), v).$$

Proof of theorem 3.7. In the maximum a-posteriori setup, the negative log-likelihood is as follows,

$$-\log p(\hat{\mathbf{G}}(\mathbf{Y}) * v|\mathbf{X}) = \frac{v}{2}\text{tr}(g(\mathbf{X})^{-1}\hat{\mathbf{G}}(\mathbf{Y})) + \frac{v}{2}\log|g(\mathbf{X})|.$$

The variational bound can be written as,

$$\text{KL}(q(\mathbf{M}|\hat{\mathbf{G}}(\mathbf{Y}))\|p(\mathbf{M}|\mathbf{X}))) = \frac{v}{2}\left(\log|g(\mathbf{X})| - \text{const.}\right) + \frac{v}{2}\text{tr}(g(\mathbf{X})^{-1}\hat{\mathbf{G}}(\mathbf{Y})) + c.$$

The objectives are equal up to additive constants. □

Such a result is true for many exponential family distributions of the form,

$$p(\mathbf{x}) = h(\mathbf{x})\exp(\boldsymbol{\eta}^T T(\mathbf{x}) - A(\boldsymbol{\eta}))$$

as for exponential family densities p and q, where q has no parameters of interest (i.e. where its contributions to the objective are constant),

$$\begin{aligned}\mathcal{E} = -\text{KL}(q\|p) &= -\mathbb{E}_q(\log q(\mathbf{x})/p(\mathbf{x})) \\ &= -[\eta(\theta_q) - \eta(\theta_p)]^T \cdot \mathbb{E}_q(\mathbf{T}(\mathbf{x})) + [A(\eta_q) - A(\eta_p)] \\ &= \eta(\theta_p)^T \cdot \mathbb{E}_q(\mathbf{T}(\mathbf{x})) - A(\eta_p) + c, \end{aligned} \tag{3.19}$$

which is the log likelihood of the exponential family distribution p, up to a constant, with the expectation of the sufficient statistic under the variational distribution being set to the observed sufficient statistic. The Gaussian version of our Wishart statement was proved in Lawrence (2005). Concretely, inference in classical GPLVMs occurs by maximising the log-likelihood,

$$\log \mathcal{MN}(\mathbf{Y}|\mathbf{0}, K_\theta(\mathbf{X}, \mathbf{X}), \mathbf{I}),$$

which is equivalent, due to theorem 3.7, to $\mathrm{KL}(q(\mathrm{S}|\hat{\mathrm{S}})\|p(\mathrm{S}|K(\mathbf{X},\mathbf{X})))$, where $\hat{\mathrm{S}} = \mathrm{Y}\mathrm{Y}^T/d$ assuming,

$$p(\mathrm{S}|K_\theta(\mathbf{X},\mathbf{X})) = \mathcal{W}(\mathrm{S}|K_\theta(\mathbf{X},\mathbf{X}),d) \text{ and } q(\mathrm{S}|\hat{\mathrm{S}}) = \mathcal{W}(\mathrm{S}|\hat{\mathrm{S}},d).$$

As another example, consider the model that explains Laplacian Eigenmaps, as we show earlier in the section; the model is formulated as,

$$\mathbf{L} * \nu|\mathbf{X} \sim \mathcal{W}\left(\left(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}\right)^{-1}, \nu\right),$$

with the MAP solution for $\mathbf{X}$ occurring at the minor eigenvectors of $\mathbf{L}$. This is equivalent, due to theorem 3.7, to KL-minimisation assuming the variational view,

$$p(\mathbf{\Gamma}|\mathbf{X}) = \mathcal{W}(\mathbf{\Gamma}|(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, \nu) \text{ and } q(\mathbf{\Gamma}|\mathbf{L}) = \mathcal{W}(\mathbf{\Gamma}|\mathbf{L}, \nu), \tag{3.20}$$

concluding the claim that the MAP views presented have a dual KL-minimising view. We will use the result corresponding to Laplacian Eigenmaps in the next chapter, to explain the behaviour of transformers. A natural question that comes up is, what happens if the variational constraint is dropped? We show in appendix B.4, that with certain generative models, dp-PCA can be recovered by marginalising the covariance/precision $\mathbf{M}$. Before concluding the section, we show that t-SNE and UMAP, in their original formulation, also fit easily into this framework, showing that our variational framework can explain a variety of constructions.

3.4.2 Explaining the neighbour embedding cases

We conclude the section by showing that (t-)SNE and UMAP minimise the ProbDR KL-divergence. As these will not be directly relevant to the exposition of the next chapter (chapter 4), we keep our discussion brief and point the reader to additional results in the appendix.

Many neighbour embedding algorithms, such as (t-)SNE and UMAP are **defined** as KL-minimising algorithms—in this section, we explicitly define the random variables on which they place distributional assumptions, and using which, the resulting objectives arise immediately. However, Damrich and Hamprecht (2021); Damrich et al. (2022) showed that optimisation details of UMAP and t-SNE play a large part in obtaining the embeddings (the objective does not

fully characterise their behaviour). Therefore, we construct simple variational interpretations for these algorithms here for completeness, but we did not use them in their original formulation in the previous section for the interpretation of t-SNE-like algorithms as MAP algorithms.

Our interpretations are based on a random adjacency matrix $\mathbf{A}' \in \{0,1\}^{n\times n}$, which represents a data-data similarity matrix. (t-)SNE and UMAP define probabilities of data similarity v_{ij} that depend on distances between the high-dimensional data points $\mathbf{Y}_{i:}$ and $\mathbf{Y}_{j:}$, and w_{ij} that depend on the distances between the low-dimensional latents $\mathbf{X}_{i:}$ and $\mathbf{X}_{j:}$. As a reminder of the definitions in section 2.5, the form of the latent probabilities for UMAP is,

$$w_{ij}^{U}(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{1 + \|\mathbf{X}_i - \mathbf{X}_j\|^2},$$

with the data-based probabilities $v_{ij}(\mathbf{Y}_i, \mathbf{Y}_j)$ following similar constructions that use the high-dimensional distances $\|\mathbf{Y}_i - \mathbf{Y}_j\|^2$.

Theorem 3.8. (t-)SNE and UMAP objectives are recovered as the ProbDR KL divergence of eq. (3.17) when model & variational distributions on an auxiliary adjacency matrix $\mathbf{A}'$ are set as Bernoulli/Categorical distributions, tabulated in table 3.1.

algo	$q(\mathbf{A}'\|\mathbf{Y})$	$p(\mathbf{A}'\|\mathbf{X})$	$\mathrm{KL}(q\|\|p)$
UMAP	$\prod_{i\neq j}^{n} \mathrm{Bernoulli}(\mathbf{A}'_{ij}\|v_{ij}^{U}(\mathbf{Y}))$	$\prod_{i\neq j}^{n} \mathrm{Bernoulli}(\mathbf{A}'_{ij}\|w_{ij}^{U}(\mathbf{X}_{i:,j:}))$	C_{UMAP}
SNE	$\prod_{i}^{n} \mathrm{Categorical}(\mathbf{A}'_{i:}\|v_{i:}^{S}(\mathbf{Y}))$	$\prod_{i}^{n} \mathrm{Categorical}(\mathbf{A}'_{i:}\|w_{ij}^{S}(\mathbf{X}_{i:,j:}))$	C_{SNE}
t-SNE	$\mathrm{Categorical}(\mathrm{vec}(\mathbf{A}')\|v_{::}^{t}(\mathbf{Y}))$	$\mathrm{Categorical}(\mathrm{vec}(\mathbf{A}')\|w_{::}^{t}(\mathbf{X}))$	$C_{\text{t-SNE}}$

Table 3.1: ProbDR assumptions that result in (t-)SNE & UMAP objectives in their original formulation (i.e. disregarding optimisation dynamics).

Proofs are provided in appendix B.5.[34]

We reiterate that although the variational ideas presented thus far can lead to a unified class of models, explicit modelling of covariances and edges through normal and Bernoulli assumptions is recommended for traditional statistical modelling, due to the simplicity of reading off their assumptions. In other words, the usage of such variational frameworks can obfuscate exactly what is being modelled and with what constraints; their use lies in non-standard use-cases as we shall see in chapter 4.

[34]This results in a KL-divergence of the form $\mathrm{KL}(q||p)$ and not $\mathrm{KL}(p||q)$ as it is written in the (t-)SNE papers—this is simply a difference in notation, as it is more natural to set the data-based probabilities to q, as explained in appendix B.6.

Before we close the subsection, we revisit our approximate inference ideas in section 3.3.3 and show that the ideas of variational constraints encoding the data above and variational inference give rise to a model graph that will be similar to SSL methods studied in chapter 4.

3.4.2.1 Revisiting approximate inference in non-linear ProbDR

In this short subsection, before we close the chapter, we show how variational inference leads to a model graph that we will see in chapter 4 when describing SSL methods, specifically one that looks like,

$$\text{“model”:}\mathbf{Y} \xrightarrow{\boldsymbol{\theta}} \mathbf{\Gamma} \qquad \text{static variational constraint} : \mathbf{Y} \rightarrow \mathbf{\Gamma},$$

where $\boldsymbol{\theta}$ corresponds to optimised parameters.

In section 3.3.3, we approximated the MAP solution for $\mathbf{X}$ given the model,

$$\mathbf{A}_{ij} \sim \text{Bernoulli}\left(\frac{\tilde{\epsilon}}{1+\tilde{s}d^2_{ij}(\mathbf{X})}\right), \tag{3.21}$$

with the following construction (where $\boldsymbol{\theta}$ is a learned parameter),

$$\mathbf{X} = \mathbf{U}_{d_q}(\mathbf{L}(\mathbf{Y}))\text{Diag}(\boldsymbol{\theta}).$$

This is ill-specified due to circularity, but it is easy to see that this process is variational inference; i.e. minimisation of,

$$\text{ELBO}(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{X}|\mathbf{Y})}(\log p(\mathbf{A}|\mathbf{X})) - \underbrace{\text{KL}(q(\mathbf{X}|\mathbf{Y})||p(\mathbf{X}))}_{\text{ignored}},$$

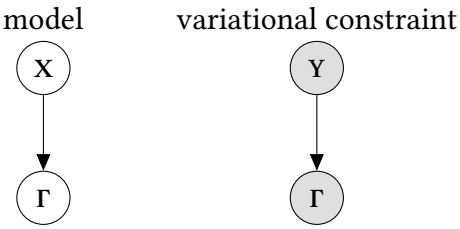
with $p(\mathbf{A}_{ij}|\mathbf{X}_i, \mathbf{X}_j)$ as in eq. (3.21), and a variational posterior, $q(\mathbf{X}|\mathbf{Y}) = \delta\left[\mathbf{X}|\mathbf{U}\cdot\text{Diag}(\boldsymbol{\theta})\right]$. The KL-divergence acts as a constant, and so it is ignored.[35]

Setting aside the approximate inference, notice that, due to the equivalence between MAP

[35]The KL is ignored due to the following behaviour. Assume $p(\text{vec}(\mathbf{X})) = \mathcal{N}(0, \sigma_p^2\mathbf{I})$ and $q(\text{vec}(\mathbf{X})) = \mathcal{N}(\mu_q, \sigma_q^2\mathbf{I})$. Then,

$$\lim_{\sigma_p^2,\sigma_q^2\rightarrow\infty,0}\text{KL}(q(\mathbf{X})||p(\mathbf{X})) = \underbrace{\frac{\|\mu_q\|^2}{2\sigma_p^2} + \frac{\sigma_q^2}{2\sigma_p^2}}_{\rightarrow 0} - \underbrace{\frac{n}{2}\log\frac{\sigma_q^2}{\sigma_p^2}}_{\text{dominates, independent of } \mu_q}.$$

estimation and the ProbDR KL-minimisation we show in this section, MAP inference for $\mathbf{X}$ assuming the model in eq. (3.21) is equivalent to KL-minimisation assuming a graph as below, where $p(\mathbf{\Gamma}_{ij}|\mathbf{X})$ is the Bernoulli model in eq. (3.21) and $q(\mathbf{\Gamma}_{ij}|\mathbf{A}_{ij}) = \text{Bernoulli}(\mathbf{\Gamma}_{ij}|\mathbf{A}_{ij})$,



Therefore, with our approximate inference ideas added back in, we see that the graph describing the framework is seemingly,

$$\text{“model”:} \qquad \mathbf{X} = \mathbf{U}(\mathbf{Y})\text{Diag}(\boldsymbol{\theta}) \rightarrow \mathbf{\Gamma},$$
$$\text{variational constraint:} \qquad \mathbf{Y} \rightarrow \mathbf{\Gamma}.$$

We will see in chapter 4 that SSL methods follow a similar process. This shows that variational distributions that appear in machine learning perform two key activities: they can form a description of data when static, and they can correspond to approximate posteriors when they are optimised.

This concludes our chapter on the interpretations of dimensionality reduction methods as probabilistic inference algorithms. We have shown that every algorithm considered performs MAP inference assuming the model class,

$$\mathbf{S}|\mathbf{X} \sim \mathcal{W}^{\{-1\}}\left(\mathbf{X}\mathbf{X}^T + \beta K_{\text{cauchy}}(\mathbf{X}) + \gamma\mathbf{I}, \nu\right),$$

and that an intermediate interpretation in the case of UMAP and t-SNE appears as,

$$\mathbf{A}_{ij}|\mathbf{X} \sim \text{Bernoulli}\left(\frac{\tilde{\epsilon}}{1 + \tilde{s}d_{ij}^2(\mathbf{X})}\right).$$

We showed that the models are statistically valid, and show semantic outcomes and efficient inference ideas. We also present a variational view, which explains our MAP algorithms of the form,

$$\mathbf{G} \sim \text{ExpFam}(g),$$

can be interpreted to be $\mathrm{KL}(q||p)$ minimisation assuming,

$$q(\mathbf{M}|\mathbf{G}) = \mathrm{ExpFam}(\mathbf{G}) \text{ and } p(\mathbf{M}|g) = \mathrm{ExpFam}(g),$$

which will be useful in the next chapter, to understand SSL methods and transformers, and suggest architectural modifications.

CHAPTER 4

Connecting ProbDR to SSL and Transformers

The core claim of the thesis is that representation learning methods are inference methods of probabilistic models which can be framed as variational frameworks with a fixed variational target representing observations. In the previous chapters, we have seen that classical dimensionality reduction methods can be explained as inference algorithms corresponding to a probabilistic model. In this chapter[1], we show that transformers perform unrolled inference in the ProbDR's Laplacian Eigenmaps model introduced in the last section, and that our interpretations suggest principled architecture modifications that lead to better performance using nanoGPT. The chapter serves as a core validation of the ProbDR framework, due to its ability to explain architectures beyond classical DR.

Concretely, we will show that single-head transformers at initialisation can be thought of as unrolled optimisation of (the ProbDR) KL-divergence via gradient descent assuming a probabilistic Laplacian Eigenmaps model, which suggests that there should be a **graph Laplacian term in the place of an adjacency matrix**—our proposed change of architecture. A graphical abstract of the major idea is presented in fig. 4.1, illustrating that an operation involving a skip connection in the architecture is interpreted to be the first term of gradient descent in $\mathbf{X} \leftarrow \mathbf{X} - \eta \nabla_{\mathbf{X}} \mathcal{L}$, with the second operation corresponding to the gradient of a term of a probabilistic objective.

The chapter is structured as follows. In the background, we present the transformer

[1] This chapter is based on **Ravuri and Lawrence (2025b)**.

architecture of Vaswani et al. (2017), and the work of Yu et al. (2023), a view presenting **(white-box) transformers** as unrolled inference within probabilistic models that is instrumental to our work. Their results show that, assuming a probabilistic model on the representations, performing gradient descent on a probabilistically-inspired (the sparse rate reduction) objective leads to each gradient descent step mirroring the actions of a transformer block. Our main idea is that the softmax operation can be derived assuming our ProbDR model behind Laplacian Eigenmaps. We also present the interpretations of Nakamura et al. (2023) that frame some self-supervised learning methods as KL-minimising algorithms assuming explicit probabilistic models, and we show how their interpretations follow ideas of ProbDR. Then, we use the ideas presented in the background to modify the white-box transformer model and show another interpretation that uses our variational form of probabilistic Laplacian Eigenmaps from the previous section, that provides an alternative explanation for how the softmax operation arises within the transformer. Finally, we show that an architecture modification, simply subtracting an identity matrix from the attention matrix (thereby performing graph diffusion or Laplacian smoothing in the attention step), arises naturally from this view. We show that this architectural change can achieve higher performance on language models and a vision transformer fit on the tiny Shakespeare (Karpathy, 2015), OpenWebText Gokaslan et al. (2019), and the downsampled Imagenet datasets (Russakovsky et al., 2015; Chrabaszcz et al., 2017) respectively.

This chapter validates the main thesis claim by showing that our interpretations are useful in a different use-case to their original motivation, demonstrating the power of probabilistic models.

4.1 Background

In this section, we present ideas that are necessary for the exposition of our main idea. We present a brief introduction to the transformer architecture of Vaswani et al. (2017), and an overview of the Laplacian Eigenmaps interpretation of the ProbDR framework which will give rise to the attention block. Then, we present the white-box transformer work of Yu et al. (2023), which forms the basis of our interpretation, and finally, we present the work of Nakamura et al. (2023) that show the correspondence between SSL algorithms and variational methods, ideas that we will borrow for our interpretation.

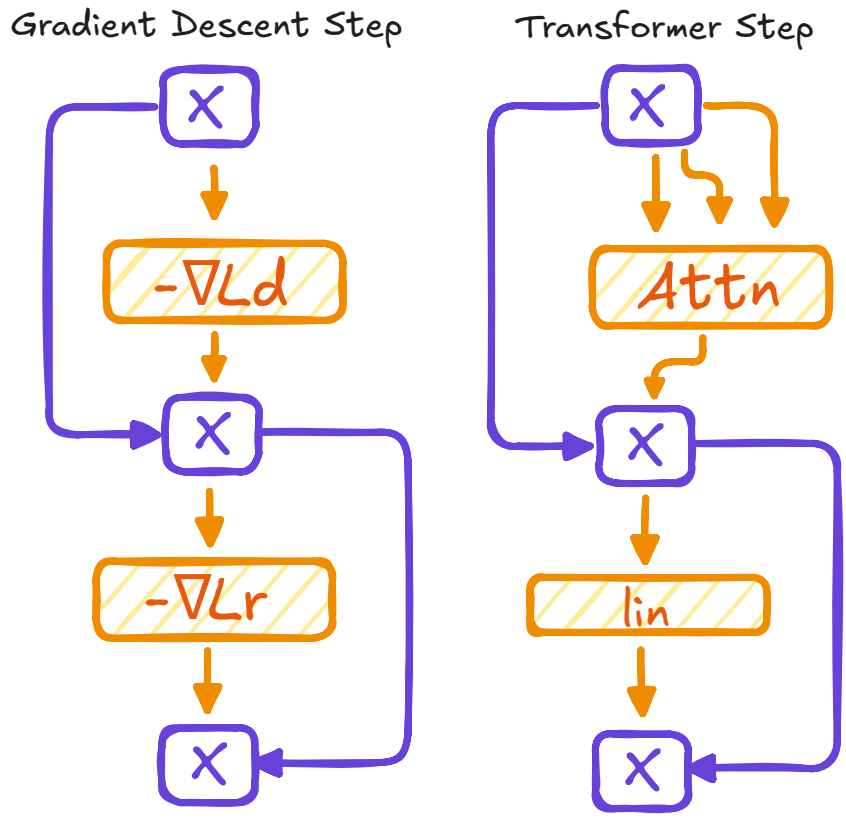


Figure 4.1: A graphical abstract of the idea that unrolled optimisation corresponds to a transformer step, showing that skip connections (in blue) inside transformer blocks are interpreted as the first term of gradient descent ($\mathbf{X} \leftarrow \mathbf{X} - \eta \nabla_{\mathbf{X}} \mathcal{L}$), with the rest of the architecture interpreted as the derivative of a log-posterior. There are two skip-connections, corresponding to alternating optimisation between a data-dependent term in the negative log-posterior $\mathcal{L}_d$ and a regularising term in the negative log-posterior $\mathcal{L}_r$.

4.1.1 The transformer architecture

Transformers are extensively used general-purpose architectures that have enabled large-scale high-performance models used for language as part of Large Language Models (LLMs), such as BERT (Devlin et al., 2019), for vision (using vision transformers, ViTs, Dosovitskiy et al. (2021)), and foundation models for speech (e.g., wav2vec, Baevski et al. (2020)). Transformers update embeddings successively by constructing a matrix that guides which tokens should meaningfully interact, rather than using fixed connectivity constraints as in CNNs and RNNs. They are flexible, as they can be used to model long-range dependencies in sequential data, interactions between patches of images, etc.

In this work, we use a simplified version of the transformer architecture, one without separate encoder and decoder blocks—we simply just use encoder blocks, as used in BERT

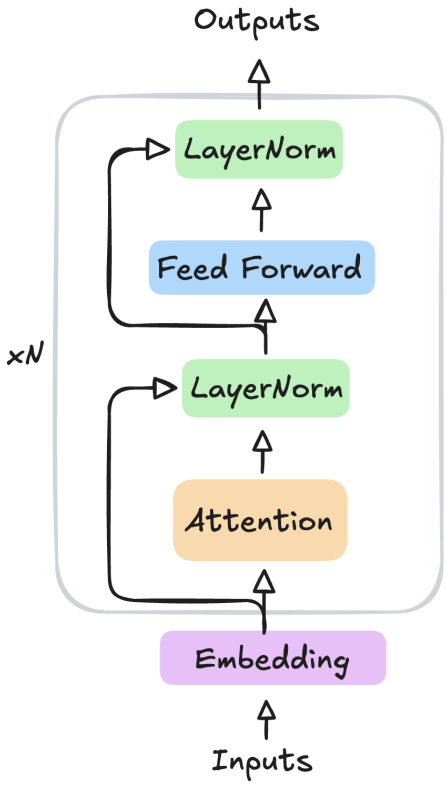


Figure 4.2: A simplified transformer architecture considered in this work, showing that the architecture consists of an initial embedding layer reducing the dimensionality of the input, with every subsequent block involving an attention step and a feed-forward step, both of which are then followed by a normalisation.

(Devlin et al., 2019), and one with single attention heads, for simplicity of exposition. The simplified architecture is illustrated in fig. 4.2; with our simplifications, the architecture first embeds a high-dimensional data point split into n tokens $\mathbf{X} \in \mathbb{R}^{n \times d}$ into $\mathbb{R}^{n \times d_q}$, and adds a positional encoding to differentiate token positions. Then, the embedding is passed into one of several blocks, made of two sub units. In the first unit, an attention matrix is computed that estimates how different tokens relate to each other, capturing dependencies between the tokens. The tokens may represent parts of an image, or a temporal sequence, and contributions of the identified "neighbours" are added to the token's embedding. Then, the embedding is normalised for stability reasons. Mathematically, the operations are as follows,

$$\mathbf{A} \longleftarrow \sigma\left(\sqrt{{d_q}^{-1}}\mathbf{X}\mathbf{W}_{\mathrm{q}}\mathbf{W}_{\mathrm{k}}^{T}\mathbf{X}^{T} + \mathbf{M}\right)$$

$$\mathbf{X} \longleftarrow \mathbf{X} + \mathbf{A}\mathbf{X}\mathbf{W}_{\mathrm{v}}$$

$$\mathbf{X} \longleftarrow \mathrm{LayerNorm}(\mathbf{X}).$$

Then, as part of the second unit, the embeddings are passed through a feed-forward network, and another normalisation layer,

$$\mathbf{X} \longleftarrow \mathbf{X} + \text{FeedForward}(\mathbf{X})$$
$$\mathbf{X} \longleftarrow \text{LayerNorm}(\mathbf{X}).$$

These embeddings are then processed in simple ways downstream, for example, a single layer acting as a regression or classification head.

In our work, we interpret the attention matrix as an adjacency matrix of a nearest-neighbour graph and show that unrolled optimization in a dimensionality reduction model leads to the transformer architecture. Prior work has studied the interpretation of attention matrices as matrices of data-point similarity or relevance; Vaswani et al. (2017) and many works since, for instance, Weng (2018); Chefer et al. (2021), have visualised attention matrices corresponding to text inputs, image patches, etc., for the purposes of interpretability. Recent work has interpreted the attention matrix as an adjacency matrix and shown that graph convolutions improve the performance of the architecture (Choi et al., 2024). In parallel, we show that the graph diffusion steps can also increase the performance of the architecture. In the realm of graph convolutional networks, Kipf and Welling (2017) (based on results in part of Defferrard et al. (2016)) motivate their architecture from a spectral graph convolutional perspective, and using a slightly different derivation of their updates, we find that an update involves a graph Laplacian term of the form $\theta_0\mathbf{x} + \theta_1\mathbf{L}\mathbf{x}$, similarly to the ideas presented in this chapter. More recently, Joshi (2025) laid out transformer attention matrices as fully connected graph adjacencies to relate transformers to graph attention networks of Veličković et al. (2018).

Next, we present the ideas of the white-box transformer, as we heavily borrow ideas from their work, to show that each block of the transformer is one step of gradient descent assuming the probabilistic model underpinning ProbDR.

4.1.2 Transformers as unrolled optimisation

We now summarise the idea of Yu et al. (2023), who base some of their work on ideas from ReduNet (Chan et al., 2021), on how transformers correspond to unrolled optimisation.[2] The

[2]Transformers can also be seen to perform gradient descent within in-context settings, as explored by von Oswald et al. (2023).

following ideas will form the main methodology of the chapter. Assume a random variable representing the post-embedding representation $\mathbf{X} \in \mathbb{R}^{n \times d_q}$, where d_q is the number of latent dimensions and n is the number of tokens corresponding to a data point (of image patches, text tokens, etc.) to which rows of the representations $\mathbf{X}$ correspond. Unrolled optimisation of a probabilistically inspired objective, assuming a probabilistic model on $\mathbf{X}$, specifically that each $\mathbf{X}_i$ is sampled from a Gaussian mixture model, leads to each gradient descent step resembling the operations in a block of a transformer. In other words, transformers perform unrolled inference assuming a probabilistic model that learns representations. The objective used to derive these results is an objective inspired by ideas in information theory, compression and linearity of representations; termed the sparse rate reduction objective $\mathcal{E}$,

$$\mathcal{E}(\mathbf{X}) = \frac{1}{2} \sum_k \underbrace{-\log\det\left(\mathbf{I} + \frac{p}{n\epsilon^2}\mathbf{X}^T\mathbf{U}_k^T\mathbf{U}\mathbf{X}\right)}_{\mathcal{E}_{\text{data}}} + \underbrace{\log\det\left(\mathbf{I} + \frac{d_q}{n\epsilon^2}\mathbf{X}^T\mathbf{X}\right)}_{\mathcal{E}_{\text{reg}}},$$

where p is the subspace dimension that bases $\mathbf{U}_i$ (which make up the attention parameters) act within, K is the number of mixture components, giving rise to K heads of multi-head attention. Optimisation of this objective with respect to $\mathbf{X}$ using gradient descent with m steps can be unrolled as a sequence of random variables,

$$\mathbf{X}_{i\cdot} \xrightarrow{T_{i\cdot}} \mathbf{X}_{ii\cdot} \xrightarrow{T_{ii\cdot}} \ldots \xrightarrow{T_m} \mathbf{X}_m.$$

Within each step of gradient descent $T_{i\cdot}$, optimisation is tackled in two steps, one that optimises the first term involving $\mathbf{U}$, and another step that optimises the regularising second term (alternating optimisation). The first step consists of steps that make up the first step of the transformer block involving attention, and the second step leads to the second block of the transformer involving the feedforward network. The softmax results from an approximation of

a matrix inverse that appears in the gradient of the first term w.r.t. $\mathbf{X}$. Mathematically[3],

$$T_i = \begin{cases} \mathbf{X}_{ii.} \longleftarrow \mathbf{X}_{i.} - \eta \nabla_{\mathbf{X}_{i.}} \mathcal{E}_{\text{data}} \\ \mathbf{X}_{ii.} \longleftarrow \mathbf{X}_{ii.} - \eta \nabla_{\mathbf{X}_{ii.}} \mathcal{E}_{\text{reg}} \end{cases}$$
$$\implies T_i \approx \begin{cases} \mathbf{X}_{ii.}^T \longleftarrow \text{LayerNorm}(\gamma_a \mathbf{X}_{i.}^T + \gamma_b \mathbf{U}^T \mathbf{U} \mathbf{X}_{i.}^T \text{softmax}(\mathbf{X}_{i.} \mathbf{U}^T \mathbf{U} \mathbf{X}_{i.}^T) \\ \mathbf{X}_{ii.}^T \longleftarrow \text{LayerNorm}(\text{ReLU}(\mathbf{R}\mathbf{X}_{ii.}^T + \mathbf{C})) \end{cases},$$

where γs are constants, $\mathbf{R}$ is an orthogonal matrix, and $\mathbf{C}$ is a constant matrix.[4] Due to the representations being latent, the model considered in Yu et al. (2023) can also be thought of as a mixture of principal component analysers[5], therefore suggesting that transformers perform inference within a linear (non-kernelised) latent variable model. In our work, we focus on single-head attention, and provide a different perspective on how the softmax arises. Instead of a Gaussian model on the latents, if our interpretation of Laplacian Eigenmaps in its variational formulation is used, the softmax term arises as an approximation of a data adjacency matrix. In the next subsection, we briefly recap the pertinent ProbDR results from chapter 3.

4.1.3 A recap of ProbDR's Laplacian Eigenmaps

Before moving onto our main result, we briefly recap ProbDR's variational Laplacian Eigenmaps formulation, which forms the basis of our interpretation. Laplacian Eigenmaps of Belkin and Niyogi (2001) is a dimensionality reduction algorithm that reduces the size of a dataset $\mathbf{Y} \in \mathbb{R}^{n \times d}$ to a smaller matrix of representations $\mathbf{X} \in \mathbb{R}^{n \times d_q}, d_q << d$. The probabilistic Laplacian Eigenmaps model is a probabilistic interpretation of the algorithm (i.e. a model, inference within which leads to the algorithm in question). It can be written as follows, where a Wishart distribution is placed on a precision matrix, of which the symmetrically normalised graph Laplacian $\mathbf{L}$ is an estimate,

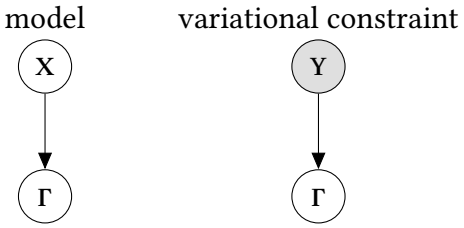
$$\nu * \mathbf{L}(\mathbf{Y}) \sim \mathcal{W}((\mathbf{X}\mathbf{X}^T + \beta \mathbf{I})^{-1}, \nu).$$

[3] assuming the simplification of the architecture—assuming a single head.
[4] The second step above is an approximation to the ISTA step, given in eqn. 92 of Yu et al. (2023).
[5] in a dual sense—acting on the latent "coordinates" and not the components.

MAP inference for latent embeddings $\mathbf{X} \in \mathbb{R}^{n \times d_q}$ in this model is equivalent to KL minimization over a random variable $\mathbf{\Gamma}$, where the model and variational constraints are written as,

$$\log p(\mathbf{\Gamma}) = \log \mathcal{W}(\mathbf{\Gamma}|(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}, \nu), \qquad \log q(\mathbf{\Gamma}) = \log \mathcal{W}(\mathbf{\Gamma}|\mathbf{L}(\mathbf{Y}), \nu),$$

where $\mathbf{L}(\mathbf{Y}) \in S_+^n$ is a graph Laplacian matrix encoding a k-nearest neighbour graph, calculated using the data $\mathbf{Y}$. The model graph can be drawn as,



The maximum of ELBO w.r.t $\mathbf{X}$, which simplifies as $-\mathrm{KL}(q(\mathbf{\Gamma})\|p(\mathbf{\Gamma}))$, is attained when the latent embeddings are estimated as follows,

$$\hat{\mathbf{X}} = \mathbf{U}_{d_q}\left(\mathbf{\Lambda}_{d_q}^{-1} - \beta\mathbf{I}_{d_q}\right)^{1/2}\mathbf{R},$$

where $\mathbf{U}_{d_q}$ are the d_q eigenvectors of the graph Laplacian corresponding to the smallest non-zero eigenvalues encoded within the diagonal matrix $\mathbf{\Lambda}$, and where $\mathbf{R} \in O(d_q)$ is an arbitrary rotation matrix. With an additional constraint, $\mathbf{X}^T\mathbf{X} = \mathbf{I}$, the optimal estimate becomes,

$$\hat{\mathbf{X}} = \mathbf{U}_{d_q}\mathbf{R}.$$

This is a consequence of the trace minimisation theorem, as the objective is simply $\mathrm{tr}(\mathbf{L}\mathbf{X}\mathbf{X}^T)$. Any arbitrary rotation still remains a solution as the objective and the constraint are invariant to rotations. In the case that we constrain the embeddings to have column norm one ($\mathbf{X}^T\mathbf{X} = \mathbf{I}$), assuming that the empirical mean of the embeddings is zero, the empirical variance of the embeddings (across the data points) is equal to $\sum_k \hat{\mathbf{X}}_{kj}^2/n = 1/n$. We will use this fact to initialise the transformer in an experiment (section 4.3.1).

Lastly, before we present our main results, we briefly review an interpretation of SSL as variational methods, as these ideas will be useful for our main exposition.

4.1.4 A variational interpretation of SSL

In this final subsection, as part of the background, we make a short digression to show how the model graph of ProbDR appears in the wider representation learning field, as self-supervised methods follow a very similar graph. Ideas from this section will be used to argue for why gradients with respect to certain terms (interpreted to be under stop-grad) do not appear in our derivation. Let $\mathbf{Y}_i^a, \mathbf{Y}_i^b, \ldots$ be augmentations/views/modalities of a data point. SimSiam, introduced in Chen and He (2020), is a self-supervised learning method that constructs representations of the data by minimising the negative inner product,

$$\mathcal{L}_i = -\sum_{m_a, m_b} f(h(\mathbf{Y}_i^{m_a}))^T f(\mathbf{Y}_i^{m_b}),$$

where the element in red is under stop-grad, and with $f(\mathbf{Y}_i^m), f(h(\mathbf{Y}_i^m)) \in \mathcal{S}^{d_q - 1}$. Nakamura et al. (2023) show that this loss function has a variational interpretation, where, if,

$$p(\mathbf{X}_i|\mathbf{Y}_i) \propto \prod_m \text{vMF}(\mathbf{X}_i|f(h(\mathbf{Y}_i^m)), \kappa),$$
$$q(\mathbf{X}_i|\mathbf{Y}_i) \propto \sum_m \delta(\mathbf{X}_i|f(\mathbf{Y}_i^m))$$

Then,

$$\Rightarrow \text{KL}(q||p) \stackrel{+}{=} -\mathbb{E}_{q(\mathbf{X}_i|\mathbf{Y}_i)}(\log p(\mathbf{X}_i|\mathbf{Y}_i)) = -\sum_{m_a, m_b} f(h(\mathbf{Y}_i^{m_a}))^T f(\mathbf{Y}_i^{m_b}) = \mathcal{L}_i.$$

Due to the stop-grad applied to the elements of the loss that form the variational constraint, we posit that the model graphs are similar to ProbDR, in that the variational constraint is treated as an observed random variable.[6] We see the variational constraint as approximating a reasonable embedding of the data *at every iteration* of the optimisation process. As an example, if f were initialised as a random projection of the data, then certain properties of the data are retained in the resulting embedding (due to the Johnson–Lindenstrauss lemma). If an

[6]The model graph implied by the specification of p and q as above, is as we saw in section 3.4.2.1, where a static variational constraint simply encodes the observed data statistic, and the function of the data that appears within the model is a variational posterior over latent variables. Therefore, viewed from a ProbDR perspective, we see the SSL framework as doing two things: defining the observed random variable through a variational constraint, and also performing variational inference for the latent random variable that appears in the model. This perspective may allow for specifying SSL methods that are full-form.

optimisation step corresponding to the model preserves/improves these properties (and does not make f degenerate or collapse), we can rely on the variational constraint to always provide an approximate but valid "view" of the data for the model to approach. In fact, Richemond et al. (2023) show that in models similar to BYOL (Grill et al., 2020) with a linear projector head, the projection step corresponds to a PCA update, leading to a "sensible" target, in that a majority of the variance in the data will be preserved. The PCA-like effect of the predictors is also noted in Tian et al. (2021). We apply a similar principle, arguing that the variational constraint is a static statistic, in section 4.2.[7]

With these components—the transformer architecture and the ProbDR interpretation of Laplacian Eigenmaps—we are now positioned to derive our main result in the coming section: that the two are approximately equivalent under a variational view. In doing so, we provide an alternative explanation as to how the softmax arises in the attention step in a transformer.

4.2 Transformers as unrolled inference in ProbDR

In this section, we present an alternative interpretation to that of Yu et al. (2023), that shows that transformers perform gradient descent on a variational objective derived using a variational form of the probabilistic Laplacian Eigenmaps model.

Firstly, rewrite the random variable corresponding to latents as $\mathbf{Z}$ (which will be the random variable that is marginalised in the variational step), and treat $\mathbf{X}$ as a parameter that encodes latent positions. Further, we add a prior to the model constraining the latents as is done with large neural networks,

$$\log p(\mathbf{\Gamma}, \mathbf{Z}) = \log \mathcal{W}\left(\mathbf{\Gamma} | (\mathbf{Z}\mathbf{Z}^T + \beta \mathbf{I})^{-1}, \nu\right) + \log \mathcal{U}^*(\mathbf{Z}).$$

$\mathcal{U}^*$ is a matrix von-Mises-Fisher distribution (a uniform distribution over matrices, with rows that lie on a d_q-dimensional hypersphere), with an additional constraint that for every row $\mathbf{x}$, $\sum_i^{d_q} x_i = 0$ (the rows have zero mean, and hence the coordinates lie on a hyperplane). Projected optimisation with this prior will lead to LayerNorm steps during optimisation. Then, we

[7] We note in passing that the objective of SimSiam, when focusing specifically on the predictors, can be written as $\mathcal{L} = -\text{tr}(\mathbf{Y}_b^T \mathbf{Y}_a \tilde{\mathbf{W}})$. With a constraint on the weights, this is the objective of canonical correlation analysis (Hardoon et al., 2004), for which, there exists a pPCA-like probabilistic interpretation, due to Bach and Jordan (2005). This may offer yet another method to see that the prediction layers of SSL-like methods can have a spectral solution, based on inference in a probabilistic model.

force the random variable $\mathbf{Z}$ to take values $\mathbf{X}$ a.s., and we modify the calculation of the graph Laplacian used in the variational constraint, so that it is a function of the latents $\mathbf{Z}$ and not the data $\mathbf{Y}$,

$$q(\mathbf{\Gamma}, \mathbf{Z}) = \mathcal{W}(\mathbf{\Gamma}|\tilde{\mathbf{L}}(\mathbf{Z}), \nu) * \delta(\mathbf{Z}|\mathbf{X}).$$

The graph Laplacian is computed as $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}(\mathbf{Z}) = \mathbf{I} - \sigma(\kappa \mathbf{Z}\mathbf{Z}^T - \mathbf{M})$ where σ is the softmax function, applied row-wise (so that the row sums of the input matrix all equal one). $\tilde{\mathbf{A}}$, we argue, is a soft (differentiable) proxy to the true nearest neighbour adjacency matrix, particularly when the latent embeddings $\mathbf{X}$ are initialised with PCA or random projections, as $\mathbf{X}\mathbf{X}^T$ is a minimal-error estimate of the empirical covariance of the data, and the covariance between similar points is expected to be similar in value. This leads to the row-wise softmax being similar and high for similar points, encoding a similarity structure. $\tilde{\mathbf{L}}$ is a random-walk (left) normalised graph Laplacian, with $\tilde{\mathbf{A}}$ having an interpretation of a transition matrix. Similarly to our approach in section 3.2.2.3, we ignore the asymmetry of the matrix as it shares eigencomponents with the symmetrically normalised graph Laplacian, which defines the Laplacian Eigenmaps result. $\mathbf{M}$ is a mask matrix (for example, if we were to disallow self-adjacency, $\mathbf{M}$ can be set to $\iota\mathbf{I}$, with $\iota \longrightarrow \infty$), and κ is a hyperparameter that can be tuned such that the proxy adjacency $\tilde{\mathbf{A}}$ is "close to" a reference nearest neighbour matrix. Even when $\mathbf{M}$ is asymmetric, the ELBO below is a function of only a symmetric matrix[8].

In a similar fashion to ProbDR, and the variational interpretation to SimSiam, we treat the variational constraint as an observed random variable, and hence do not account for gradient updates w.r.t. $\mathbf{X}$ leading from terms corresponding to the variational constraint. Hence, the KL-divergence with stop-grad applied to the variational constraint is,

$$\mathrm{KL}\big(q(\mathbf{\Gamma}, \mathbf{Z}) \| p(\mathbf{\Gamma}, \mathbf{Z})\big) \propto \underbrace{\mathrm{tr}\big(\tilde{\mathbf{L}}(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})\big)}_{\mathcal{L}_{\text{data}}} - \underbrace{\log\det(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})}_{\mathcal{L}_{\text{reg}}} + c,$$

where $\forall i : \mathbf{X}_i \in \mathcal{S}^{d_q - 1}$ and $\sum_j \mathbf{X}_{ij} = 0$. In the white-box transformer work, a transformer block's sequence of updates follows gradient descent of an objective in steps; given an objective $\mathcal{L}(\mathbf{X}) = \mathcal{L}_{\text{data}}(\mathbf{X}) - \mathcal{L}_{\text{reg}}(\mathbf{X})$, where a transformer block (at initialisation) calculations correspond to an

[8] Let $\mathrm{Sym}(\mathbf{L}) = 0.5(\mathbf{L} + \mathbf{L}^T)$. This is because, $\mathrm{tr}(\mathbf{L}\mathbf{P})$ with asymmetric $\mathbf{L}$ follows,

$$\mathrm{tr}(\mathrm{Sym}(\mathbf{L})\mathbf{P}) = 0.5\mathrm{tr}((\mathbf{L} + \mathbf{L}^T)\mathbf{P}) = 0.5\mathrm{tr}(\mathbf{L}\mathbf{P}) + 0.5\mathrm{tr}(\mathbf{P}\mathbf{L}) = \mathrm{tr}(\mathbf{L}\mathbf{P}).$$

alternating optimisation process involving the updates,

$$\mathbf{X}' \longleftarrow \mathbf{X} - \eta * \frac{d\mathcal{L}_{\text{data}}}{d\mathbf{X}}, \qquad \mathbf{X} \longleftarrow \mathbf{X}' + \eta * \frac{d\mathcal{L}_{\text{reg}}}{d\mathbf{X}'}.$$

Furthermore, in our work, we ignore the positivity constraint that leads to the ReLU activation, which forms a part of the fully connected segment of the transformer for ease of exposition; however, this can be re-added simply by incorporating a sparsity prior used with the white-box transformer, as our regularization term is identical to theirs, the sparsity terms notwithstanding.

We now show how an (encoder) transformer block's operations[9] arise as optimisation steps of our objective. First, observe that,

$$\frac{d\mathcal{L}}{d\mathbf{X}} = 2\tilde{\mathbf{L}}\mathbf{X} = 2(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{X}$$

and so, a gradient descent update for optimisation of $\mathcal{L}_{\text{data}}$ follows,

$$\mathbf{X} \longleftarrow \mathbf{X} + 2\eta(\sigma(\kappa\mathbf{X}\mathbf{X}^T - \mathbf{M}) - \mathbf{I})\mathbf{X}.$$

The element highlighted (which is the degree matrix, in this case, the identity matrix) in red shows the only difference to a standard attention operation (as the attention matrix is the only term that appears in the ordinary architecture). Next, we must take a projection step to ensure that $\forall i : \mathbf{X}_i \in \mathcal{S}^{d_q - 1}$ and $\sum_j \mathbf{X}_{ij} = 0$, and hence,

$$\mathbf{X} \longleftarrow \text{LayerNorm}(\mathbf{X}).$$

We now optimise with respect to $\mathcal{L}_{\text{reg}}$. This is exactly the same form of regularisation (apart from the sparse prior that gives rise to the ReLU, which is ignored for the sake of exposition) as the term that appears in the work of the white-box transformer. We refer the reader to that work for a careful argument for how this term approximately gives rise to a linear update (and a ReLU network if a positivity and sparsity priors are included), but here, we simply approximate,

$$\frac{d\mathcal{L}_{\text{reg}}}{d\mathbf{X}} = 2(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}\mathbf{X} = 2\mathbf{X}(\mathbf{X}^T\mathbf{X} + \beta\mathbf{I})^{-1} \approx \mathbf{X}\boldsymbol{W},$$

[9]With a single head as previously noted; we believe that this can be extended by considering a multiple-expert type distribution as part of the variational constraint.

where $\mathbf{W}$ is an estimate of a decorrelating matrix $(\mathbf{X}^T\mathbf{X} + \beta\mathbf{I})^{-1}$. Therefore our remaining optimisation steps simply involve a linear update and another projection,

$$\mathbf{X} \longleftarrow \mathbf{X} + \eta\mathbf{X}\mathbf{W}$$
$$\mathbf{X} \longleftarrow \text{LayerNorm}(\mathbf{X}),$$

which completes the transformer block operations, assuming simple initialisations. A key insight is that our probabilistic interpretation does Laplacian smoothing (**graph diffusion**—i.e. the subtraction of an identity matrix, or a degree matrix, from the attention matrix), whereas the standard attention step does not. The resulting approximations do not change the computational complexity of the transformer as this can be implemented simply as,

```
x = att @ value - value.
```

For our tiny-shakespeare experiments presented in the next section, this is the only change we made to the nanoGPT source code.

We believe that this helps optimisation due to **stability** reasons. It can be shown that optimisation of any loss function of the form $\mathcal{L} = \sum_{ij} \phi(d_{ij}^2)$ involves gradients of the form of a graph Laplacian. We hypothesise that the form of such updates is crucial for optimisation stability, and to prevent collapse to a degenerate solution or oversmoothing[10].

Lastly, we explore what weights do in this framework. We posit that an update such as $\mathbf{X} \longleftarrow \mathbf{X} + \mathbf{X}\mathbf{W}_{\text{lin}}$ can be interpreted as a rotation (which, under the probabilistic Laplacian Eigenmaps model, the solution is invariant to) and a scaling, which, under our interpretation, corresponds to a learnt step size $\eta = |\mathbf{W}|^{1/d_q}$. This is a restatement of the belief that transformers *learn to learn*, in other words, perform optimisation (assuming a dimensionality reduction or clustering model) with just n_{blocks} steps.[11]

In the next section, we show two experiments showing that transformers can indeed do act as clustering/dimensionality reduction models at initialisation, and that our modified architecture increases performance in a language and vision task.

[10] A related argument is given in Miller (2023), who point out that the standard attention forces every head to contribute non-trivially, without the possibility of "doing nothing".

[11] In this view, the transformer is not merely a static function, but an optimisation process where the weights encode the meta-parameters (like step-size) of a gradient descent algorithm tailored to the data distribution. Chen et al. (2021) provide an overview of the field of learning-to-optimise, a field where models are trained on optimisation problems to enable fast and data-oriented approximate optimisation.

4.3 Experiments

We provide two main experiments to show validity of the ideas presented thus far. In the first, we show that an embedding+transformer network initialised in a simple way performs dimensionality reduction and clustering, using flattened images from the MNIST dataset. In the second, we show that removing an identity matrix from the attention matrix as suggested by our derivation increases performance on the Shakespeare dataset and a downsampled (16-by-16) version of ImageNet.

4.3.1 Transformers cluster high-dimensional points

The details of our dimensionality reduction experiment are as follows. We set up a sequential neural network, with the structure:

$$\text{flattened image} \longrightarrow \text{linear projection that reduces dim.} \longrightarrow \text{transformer}.$$

The initial projection layer was initialised with weights,

$$\mathbf{W}_{\text{proj}} \sim \mathcal{MN}(0, \mathbf{I}_d/d, \mathbf{I}_{d_q}),$$

that is, randomly initialised with Gaussian entries, enforcing a Gaussian random projection. Next, the blocks making up the transformer were initialised as follows:

```
block = torch.nn.TransformerEncoderLayer(
    d_model=128,
    nhead=1,
    dim_feedforward=128,
    dropout=0.0,
    activation=torch.nn.Identity(),
    norm_first=False,
)
```

which leads to the `forward(X, src_mask)` method behaving as,

$$\mathbf{X} \longleftarrow \mathbf{X} + \sigma\left(\sqrt{{d_q}^{-1}}\mathbf{X}\mathbf{W}_{\mathrm{q}}\mathbf{W}_{\mathrm{k}}^{T}\mathbf{X}^{T} + \mathbf{M}\right)\mathbf{X}\mathbf{W}_{\mathrm{v}}$$

$$\mathbf{X} \longleftarrow \mathrm{LN}(\mathbf{X}) \odot \boldsymbol{\sigma}_{\mathrm{ln1}} + \boldsymbol{\mu}_{\mathrm{ln1}},$$

$$\mathbf{X} \longleftarrow \mathbf{X} + \mathbf{X}\mathbf{W}_{\mathrm{lin}} + \boldsymbol{\mu}$$

$$\mathbf{X} \longleftarrow \mathrm{LN}(\mathbf{X}) \odot \boldsymbol{\sigma}_{\mathrm{ln2}} + \boldsymbol{\mu}_{\mathrm{ln2}}.$$

The initialisations are as follows.

- **Number of blocks:** 8. We found that increasing the number of blocks makes the latents collapse into extremely tight clusters.
- **LayerNorms:** The LayerNorms have post-normalization weights associated with them. We set $\boldsymbol{\sigma}_{\mathrm{ln}*} = \mathbf{1}/\sqrt{n}$, which is because we expect the optimum to be akin to eigenvectors of a graph Laplacian, which would have variance $1/n$, as explained in the background. All translations are zeroed; $\boldsymbol{\mu} = 0$.
- **Transformer weights**: the transformer block weights are $\mathbf{W}_{\mathrm{q}} = \sqrt{\kappa n}\mathbf{I}$, $\mathbf{W}_{\mathrm{k}} = \sqrt{\kappa n/d_q}\mathbf{I}$ and $\mathbf{W}_{\mathrm{v}} = 2\eta$ corresponding to the query, key, value weight matrices. The query and key matrices were set up such that the inner product matrix, pre-softmax, has a diagonal equal to κ. We set $\kappa = 30$, based on the clustering empirically observed in the resulting graph Laplacian's eigenvectors.
- **Feedforward block**: As we use the out-of-the-box implementation of attention in torch, we emulate the removal of the diagonal degree matrix times $\mathbf{X}$ through the feedforward block. Therefore, the feed-forward block is functionally a single layer with weight $\mathbf{W}_{\mathrm{lin}} = -2\eta$.
- **Learning rate**: $\eta = 0.4$ and **latent dimension**: $d_q = 128$.

With these settings, passing the flattened images through the transformer (where every flattened MNIST vector is treated as one token, with the self-attention being computed across all data points) leads to the embeddings recovered clustering by digit; we show the embeddings in fig. 4.3. This suggests that the transformer acts as a dimensionality reduction and clustering algorithm in our setting. In the next experiment, we explore whether our modified architecture leads to better performance on standard tasks.

Figure 4.3: The first two latent dimensions of embeddings constructed from flattened MNIST images after the random initialisation making up the initial embedding later (i.e. the initial projection layer that converts pixels to a latent representation) (**left**), and after eight steps through a transformer block (**right**), showing that embeddings that pass through the transformer blocks cluster points in the latent space.

4.3.2 Graph diffusion improves performance

In the second experiment, we simply replace the attention matrix $\mathbf{A}$ within a transformer architecture, found in nanoGPT (Karpathy, 2022) with the negative graph Laplacian $\mathbf{A} - \mathbf{I}$ (the change suggested by our derivation), and measure validation performance over multiple runs on the Shakespeare dataset. We also repurpose the code to build a small vision transformer, and train it naively (i.e. without random augmentations, learning rate schedules, etc.) on the downsampled Imagenet dataset, where all images are 16 by 16 pixels. On this dataset, a benchmark given in Chrabaszcz et al. (2017) achieves 40% validation accuracy, whereas our naïve ViT achieves around 26%.

In both cases however, **validation performance** improves when we replace the attention matrix by the negative graph Laplacian. Our implementation was a very simple modification of nanoGPT (Karpathy, 2022), in accordance with the change proposed above, to the attention matrix. We measure validation performance using held-out accuracy on the image task, with about 4% being used as the held-out sample, and use the validation loss reported by the implementation as a validation metric for the language task.

In addition to these small-scale experiments, we also ran two pretraining runs of GPT-2 on the OpenWebText dataset (Gokaslan et al., 2019) using nanoGPT with and without our modification, but instead of using `x = x - value`, we use `x = x - a*value`, where $a \in (0, 1)$ is an optimised parameter. This adds just one single parameter to the full model. This is done

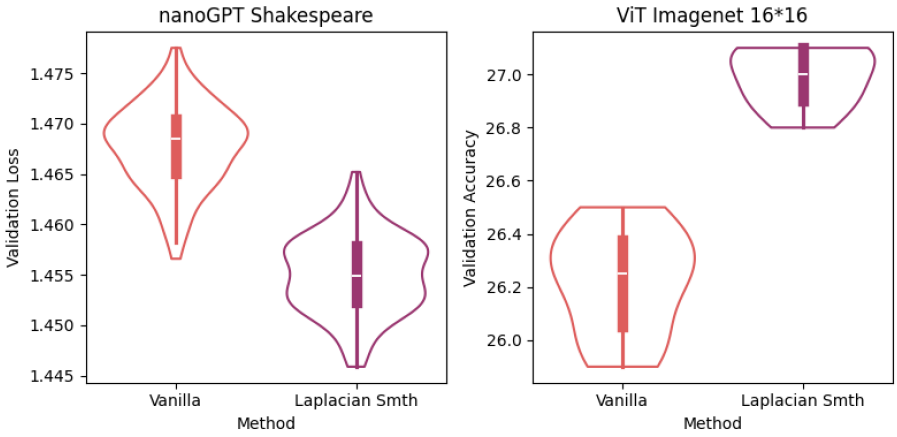


Figure 4.4: Left: validation losses on the Shakespeare dataset and **right:** validation accuracies on a downsampled Imagenet dataset, showing that Laplacian smoothing achieves a better performance in both cases.

because GPT-2 initialisations and training schedules are somewhat optimised for the base transformer architecture[12], whereas, we do not run sweeps to find optimal hyperparameters for our architecture. Furthermore, our runs do not fully converge as in the case of tiny-shakespeare, and we hypothesise that the model is not large enough for the imposed regularisation to work as well out of the box, without hyperparameter sweeps. We find that our modification yields an improvement in the validation loss of 3.1×10^{-3}; we leave extensive testing of the idea to future work.

This concludes our results showing that transformers perform inference in probabilistic Laplacian Eigenmaps, which leads to a graph diffusion step in place of the standard attention step. In the next section, we briefly recap the main takeaways of the chapter.

4.4 Discussion

In this chapter, we have provided an interpretation within which transformer blocks correspond to unrolled inference assuming a probabilistic Laplacian Eigenmaps model. We also show that the ideas of ProbDR, specifically relating to its variational graphs, extend beyond dimensionality reduction to self-supervised learning and transformers, and therefore representation learning more generally. We show that a simple architectural tweak—using a negative Laplacian $\mathbf{A} - \mathbf{I}$ in place of the attention matrix $\mathbf{A}$—can yield gains in language and vision settings.

[12]Based on the insights found by Hoffmann et al. (2022); Radford et al. (2019), relating to projection weight initialisations, model size, hyperparameter settings and learning rate schedules.

Tying the results obtained back to the central claim of the thesis, we have shown that our probabilistic model underpins transformers, and that this insight may lead to a better architecture for general-purpose machine learning. This exemplifies our central philosophy that **probabilistic models are applicable outside of their primary intended use-cases**.

We envision that it may be possible to start with probabilistic models that are appropriate to domain-specific cases, and unroll optimisation as done here, to introduce new neural architectures. Future work can explore whether non-linear (kernelised) probabilistic models of dimensionality reduction can increase performance in models with lower latent dimensionality.

CHAPTER 5

Conclusion of Thesis

In the thesis, we posed the question: is there a probabilistic model that explains methods in scientific representation learning? We found that the answer, from various case-studies and probabilistic interpretations of the methods, to be that latent variable models of similarity—the ProbDR models—underpin the methods studied in this thesis, including neural architectures. Moreover, **estimators** constrain large aspects in the latent variable models, specifically, the **what** that is modelled. A summary of the key ideas presented is as follows.

We started with the background. In section 2.2, we showed how probabilistic interpretations can be formed: by interpreting algorithms as optimisation algorithms of certain objectives, with those objectives being interpreted to be either likelihoods of probabilistic models, or as lower-bounds on related quantities, such as the evidence-lower bound. We showed that the latter (variational) interpretations appear when the optimised variable is on the left hand side of a sampling statement (i.e. when the derived model statement looks like $f(\text{variable}) \sim$ fixed distribution) or when the derived model statement seems circular (i.e. as $\mathbf{Y} \to \cdots \to \mathbf{Y}$). In the former case specifically, the variational perspectives can provide a generative picture. Lastly, we pointed out that the variational constraints do not form approximate posteriors in the sense of variational inference, but act as a form of observation and define a model class. In section 2.3, we reviewed "projective" methods of representation learning that correspond to model classes of the form $\mathbf{Y} \to \mathbf{X}$, and showed how they explicitly construct estimators for aspects of data that are valuable (which can include estimators of data density, or vectors that preserve data distances). We showed how Monte-Carlo dropout can improve zero-shot scalar-property prediction with models that are estimators of data density, presumably for

calibration reasons. In section 2.4, we presented some generative models (i.e. of the form $\mathbf{X} \rightarrow \mathbf{Y}$) for representation learning, and showed that many methods in science constrain aspects of such models by explicitly estimating/constraining aspects of the latent variable models through prior knowledge. In section 2.5, we then presented dimensionality reduction algorithms that form latents using eigendecompositions, and neighbour-embedding methods that perform probabilistic graph-matching to obtain latents. We mentioned that these do not fall easily into the taxonomy of models we described earlier in the chapter.

Chapter 3 formed the core of the thesis. In section 3.2 and section 3.3, we showed that most classical methods of dimensionality reduction are MAP inference methods assuming the model,

$$\mathbf{S}|\mathbf{X} \sim \mathcal{W}^{\{-1\}}(\mathbf{X}\mathbf{X}^T + \beta K(\mathbf{X}, \mathbf{X}) + \gamma \mathbf{I}, \nu),$$

where all hyperparameters are fixed, and where the main difference across different algorithms arises from the estimator of the covariance $\mathbf{S}$. We posited that the estimated statistic is minimal in many methods, in the sense that it estimates only a specific small aspect of the data. For example, if the covariance is estimated as $\mathbf{S} = \mathbf{L}^+$, the covariance must only be a function of the nearest neighbour graph, which may correspond to different high-level concepts (such as zero occurrence rates) depending on the distribution of the data. We showed that, in the case of (t-)SNE/UMAP-like algorithms, an intermediate interpretation that gives rise to the above statement is an explicit edge-detection model,

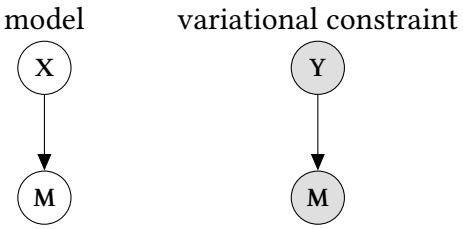
$$\mathbf{A}_{ij} \sim \text{Bernoulli}\left(\tilde{\epsilon}\frac{1}{1 + \tilde{s}d_{ij}^2(\mathbf{X})}\right).$$

We show that embeddings found by our interpretations resemble those of t-SNE-like algorithms. We then showed in section 3.3 that these are semantically coherent models (e.g. because the kNN graph only has a few edges, $\tilde{\epsilon}$ ensures that the average probability of adjacency is small). Moreover, we showed that our probabilistic interpretations hint towards an SGNS-kind algorithm that, although not very qualitatively similar to CNE, shows that there may be a link between models of the form,

$$\texttt{transformed data} \sim \texttt{linear model} \text{ and } \texttt{raw data} \sim \texttt{non-linear model}.$$

In section 3.4, we showed that the models with MAP interpretations above can be interpreted

to be KL-minimising algorithms assuming a variational framework illustrated below.



Finally, in chapter 4, we showed that transformers and SSL correspond to inference within a similar variational framework to ProbDR. Concretely, we showed that transformers correspond to unrolled inference assuming our variational model underpinning Laplacian Eigenmaps, where the softmax operation estimates an adjacency matrix (and through it, a graph Laplacian), the skip-connection is the first part of the gradient descent step $\mathbf{X} \leftarrow \mathbf{X} + \eta \nabla_{\mathbf{X}} \mathcal{E}$, and the latter part forms the gradient of the ELBO. Moreover, we argued that stop-grad elements within this process, and the variational frameworks corresponding to SSL, correspond to the fact that the variational constraints are treated as observed random variables estimating some aspect of the data. We show that our derivations suggest that a negative graph Laplacian should be used in place of the standard attention matrix, and that this architecture change increases performance on two tasks. Future work can explore whether these ideas can be used for neural architecture search, or at least to reason about other frameworks with.

Before concluding the thesis, we provide a brief summary of the other future directions discussed in the thesis.

5.1 Summary of future directions

First and foremost, in the thesis, we introduced a probabilistic framework that estimates certain quantities of interest using non-traditional estimators. Identifying what *exactly* they estimate in context-specific terms, e.g. what is retained in a kNN graph built from zero-inflated data, is left for future work. Transforms of the data may exist such that the sample covariance approximately retains only the information contained in the pseudo-inverse of the graph Laplacian. A study of this will shed light into building simpler models that estimate directly the aspect of the data that one is interested in.

Secondly, studying natural gradients in ProbDR, Bayesian decision-theoretic interpretations

of the framework and sampling behaviour within it may unlock new methods of inference of latent variables.

Thirdly, the thesis uses coarse approximations in many places, tighter approximations of our ideas may shed light into whether it is possible to show differences between interpretations leading to t-SNE-like or UMAP-like behaviour. Moreover, better analytical inference algorithms may exist for inference within models that underpin t-SNE-like methods.

Lastly, in the context of explaining neural architectures using ProbDR, non-linear (kernelised) probabilistic models of dimensionality reduction may increase performance in models with lower latent dimensionality. Furthermore, these ideas may enable discovery of new architectures via consideration of gradient descent in more specialised/context-specific latent variable models. We also note that, as transformer embeddings are typically high-dimensional, the emergence of linear "concept directions" predicted by the linear representation hypothesis may make linear latent-variable interpretations surprisingly effective. Future work can study whether kernelised ProbDR variants can provide benefits when the embedding dimension is constrained.

To conclude, the core contributions of the thesis, with respect to the wider field, were the introduction of the ProbDR framework that unifies many methods in classical dimensionality reduction, and explain aspects of neural architectures, from a probabilistic perspective, and is the first of its kind to do so. Secondarily, we showcased many case-studies in science and showed how one can approach them through a probabilistic lens, how one constrains models in practice, and how existing ideas in the field can be extended.

I hope that the thesis is a valuable reference providing case-studies for approaching various problems in scientific representation learning and for those wishing to understand various representation learning methods from a probabilistic perspective. The ideas presented in this thesis can be developed, to further understand what the various methods model in a high-level semantic sense, to improve the approximations made, and find new model frameworks or architectures using the ideas presented.

References

Ahmed, S., Rattray, M., and Boukouvalas, A. (2018). GrandPrix: scaling up the Bayesian GPLVM for single-cell data. *Bioinformatics*, 35(1):47–54. (Cited on page 188.)

Bach, F. R. and Jordan, M. I. (2005). A probabilistic interpretation of canonical correlation analysis. (Cited on pages 49 and 123.)

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. (Cited on pages 116, 161, and 163.)

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W. H., Ng, L. G., Ginhoux, F., and Newell, E. W. (2019). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44. (Cited on page 58.)

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press. (Cited on pages 56, 79, 80, and 120.)

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828. (Cited on pages 14 and 16.)

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236. (Cited on page 80.)

Bhatia, R. (2007). *Positive Definite Matrices*. Princeton University Press. (Cited on page 96.)

Birnbaum, A. (1962). On the foundations of statistical inference. *Journal of the American Statistical Association*, 57(298):269–306. (Cited on page 29.)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg. (Cited on page 168.)

Bishop, C. M., Svensén, M., and Williams, C. K. I. (1998). Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234. (Cited on page 47.)

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877. (Cited on pages 106 and 187.)

Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc. (Cited on page 59.)

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M. P., and Durrande, N. (2021). Matérn Gaussian processes on graphs. (Cited on pages 189 and 201.)

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. (2020). Matérn gaussian processes on riemannian manifolds. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc. (Cited on pages 77, 189, and 196.)

Bui, T. D. and Turner, R. E. (2015). Stochastic variational inference for gaussian process latent variable models using back constraints. In *Black Box Learning and Inference NIPS workshop*. (Cited on pages 106 and 187.)

Bürkner, P.-C. (2017). brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software*, 80(1):1–28. (Cited on page 33.)

Cai, D., Campbell, T., and Broderick, T. (2021). Finite mixture models do not reliably learn the number of components. In *International conference on machine learning*, pages 1158–1169. PMLR. (Cited on pages 45, 168, and 169.)

Cao, J., Spielmann, M., Qiu, X., Huang, X., Ibrahim, D. M., Hill, A. J., Zhang, F., Mundlos, S., Christiansen, L., Steemers, F. J., Trapnell, C., and Shendure, J. (2019). The single-cell

transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502. (Cited on page 189.)

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32. (Cited on page 34.)

Chan, K. H. R., Yu, Y., You, C., Qi, H., Wright, J., and Ma, Y. (2021). Redunet: A white-box deep network from the principle of maximizing rate reduction. (Cited on page 118.)

Chandrasekaran, S. N., Ackerman, J., Alix, E., Ando, D. M., Arevalo, J., Bennion, M., Boisseau, N., Borowa, A., Boyd, J. D., Brino, L., Byrne, P. J., Ceulemans, H., Ch'ng, C., Cimini, B. A., Clevert, D.-A., Deflaux, N., Doench, J. G., Dorval, T., Doyonnas, R., Dragone, V., Engkvist, O., Faloon, P. W., Fritchman, B., Fuchs, F., Garg, S., Gilbert, T. J., Glazer, D., Gnutt, D., Goodale, A., Grignard, J., Guenther, J., Han, Y., Hanifehlou, Z., Hariharan, S., Hernandez, D., Horman, S. R., Hormel, G., Huntley, M., Icke, I., Iida, M., Jacob, C. B., Jaensch, S., Khetan, J., Kost-Alimova, M., Krawiec, T., Kuhn, D., Lardeau, C.-H., Lembke, A., Lin, F., Little, K. D., Lofstrom, K. R., Lotfi, S., Logan, D. J., Luo, Y., Madoux, F., Marin Zapata, P. A., Marion, B. A., Martin, G., McCarthy, N. J., Mervin, L., Miller, L., Mohamed, H., Monteverde, T., Mouchet, E., Nicke, B., Ogier, A., Ong, A.-L., Osterland, M., Otrocka, M., Peeters, P. J., Pilling, J., Prechtl, S., Qian, C., Rataj, K., Root, D. E., Sakata, S. K., Scrace, S., Shimizu, H., Simon, D., Sommer, P., Spruiell, C., Sumia, I., Swalley, S. E., Terauchi, H., Thibaudeau, A., Unruh, A., Van de Waeter, J., Van Dyck, M., van Staden, C., Warchoł, M., Weisbart, E., Weiss, A., Wiest-Daessle, N., Williams, G., Yu, S., Zapiec, B., Żyła, M., Singh, S., and Carpenter, A. E. (2023). Jump cell painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *bioRxiv*. (Cited on pages 170 and 173.)

Chefer, H., Gur, S., and Wolf, L. (2021). Transformer interpretability beyond attention visualization. (Cited on page 118.)

Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z., and Yin, W. (2021). Learning to optimize: A primer and a benchmark. (Cited on page 126.)

Chen, X. and He, K. (2020). Exploring simple siamese representation learning. (Cited on page 122.)

Choi, J., Wi, H., Kim, J., Shin, Y., Lee, K., Trask, N., and Park, N. (2024). Graph convolutions enrich the self-attention in transformers! (Cited on page 118.)

Chrabaszcz, P., Loshchilov, I., and Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the cifar datasets. (Cited on pages 115 and 129.)

Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30. Special Issue: Diffusion Maps and Wavelets. (Cited on pages 81 and 83.)

Cooper, E., Huang, W.-C., Tsao, Y., Wang, H.-M., Toda, T., and Yamagishi, J. (2023). The voicemos challenge 2023: Zero-shot subjective speech quality prediction for multiple domains. (Cited on page 164.)

Cullot, G., Boutin, J., Toutain, J., Prat, F., Pennamen, P., Rooryck, C., Teichmann, M., Rousseau, E., Lamrissi-Garcia, I., Guyonnet-Duperat, V., Bibeyran, A., Lalanne, M., Prouzet-Mauléon, V., Turcq, B., Ged, C., Blouin, J.-M., Richard, E., Dabernat, S., Moreau-Gaudry, F., and Bedel, A. (2019). Crispr-cas9 genome editing induces megabase-scale chromosomal truncations. *Nature Communications*, 10(1):1136. (Cited on page 173.)

Damrich, S., Böhm, N., Hamprecht, F. A., and Kobak, D. (2022). From t-sne to umap with contrastive learning. In *The Eleventh International Conference on Learning Representations*. (Cited on pages 85, 87, 109, 191, and 192.)

Damrich, S. and Hamprecht, F. A. (2021). On umap's true loss function. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5798–5809. Curran Associates, Inc. (Cited on pages 87 and 109.)

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366. (Cited on page 161.)

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA. Curran Associates Inc. (Cited on page 118.)

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142. (Cited on page 86.)

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. (Cited on pages 41, 116, and 117.)

Dorta, G., Vicente, S., Agapito, L., Campbell, N. D. F., and Simpson, I. (2018). Structured uncertainty prediction networks. (Cited on page 52.)

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. (Cited on page 116.)

Draganov, A. and Dohn, S. (2023). Unexplainable explanations: Towards interpreting tsne and umap embeddings. *arXiv preprint*. (Cited on page 95.)

Drew, K., Wallingford, J. B., and Marcotte, E. M. (2021). hu.map 2.0: integration of over 15,000 proteomic experiments builds a global compendium of human multiprotein assemblies. *Molecular Systems Biology*, 17(e10016). (Cited on page 176.)

Feragen, A. and Hauberg, S. (2016). Open problem: Kernel methods on manifolds and metric spaces. what is the probability of a positive definite geodesic exponential kernel? In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1647–1650, Columbia University, New York, New York, USA. PMLR. (Cited on page 77.)

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. (2020). Se(3)-transformers: 3d roto-translation equivariant attention networks. (Cited on page 41.)

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. (Cited on pages 42 and 164.)

Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Valiollah Pour Amiri, V., Hong, J., Wu, K., Jayasuriya, M., Mehlman, E., Langevin, M., Liu, Y., Samaran, J., Misrachi, G., Nazaret, A., Clivio, O., Xu, C., Ashuach, T., Gabitto, M., Lotfollahi, M., Svensson, V., da Veiga Beltrame, E., Kleshchevnikov, V., Talavera-López, C., Pachter, L., Theis, F. J., Streets, A., Jordan, M. I., Regier, J., and Yosef, N. (2022). A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, 40(2):163–166. (Cited on page 189.)

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC. (Cited on pages 15, 26, and 28.)

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459. (Cited on pages 15 and 28.)

Gillespie, M. et al. (2022). The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, 50(D1):D687–D692. (Cited on page 176.)

Giurgiu, M. et al. (2019). CORUM: the comprehensive resource of mammalian protein complexes–2019. *Nucleic Acids Research*, 47(D1):D559–D563. (Cited on page 176.)

Gokaslan, A., Cohen, V., Pavlick, E., and Tellex, S. (2019). Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`. (Cited on pages 115 and 129.)

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338. (Cited on page 76.)

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. (Cited on page 43.)

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243. (Cited on pages 36 and 157.)

Griffiths, R.-R., Klarner, L., Moss, H. B., Ravuri, A., Truong, S., Stanton, S., Tom, G., Rankovic, B., Du, Y., Jamasb, A., Deshwal, A., Schwartz, J., Tripp, A., Kell, G., Frieder, S., Bourached, A., Chan, A., Moss, J., Guo, C., Durholt, J., Chaurasia, S., Strieth-Kalthoff, F., Lee, A. A., Cheng, B.,

Aspuru-Guzik, A., Schwaller, P., and Tang, J. (2023). Gauche: A library for gaussian processes in chemistry. (Cited on pages 23 and 163.)

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. (Cited on page 123.)

Gundersen, G. W., Zhang, M. M., and Engelhardt, B. E. (2020). Latent variable modeling with random features. (Cited on page 52.)

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings. (Cited on pages 41, 42, 43, and 59.)

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806. (Cited on page 50.)

Haque, A., Engel, J., Teichmann, S. A., and Lönnberg, T. (2017). A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9(1):75. (Cited on page 55.)

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664. (Cited on pages 49 and 123.)

Hastie, T., Tibshirani, R., Friedman, J., et al. (2009). The elements of statistical learning. (Cited on page 43.)

Hendrycks, D. and Gimpel, K. (2018). A baseline for detecting misclassified and out-of-distribution examples in neural networks. (Cited on page 167.)

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*. (Cited on pages 52 and 187.)

Hinton, G. and Roweis, S. (2002). Stochastic neighbor embedding. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, page 857–864, Cambridge, MA, USA. MIT Press. (Cited on pages 56, 65, and 200.)

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. (Cited on pages 38 and 106.)

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. (2022). Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc. (Cited on page 130.)

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441. (Cited on page 72.)

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. (2022). Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122. (Cited on page 164.)

Hu, T., Liu, Z., Zhou, F., Wang, W., and Huang, W. (2023). Your contrastive learning is secretly doing stochastic neighbor embedding. (Cited on page 65.)

Huang, W.-C., Cooper, E., Tsao, Y., Wang, H.-M., Toda, T., and Yamagishi, J. (2022). The voicemos challenge 2022. (Cited on page 164.)

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430. (Cited on page 48.)

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. (Cited on page 175.)

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press. (Cited on page 33.)

Johnson, W. B., Lindenstrauss, J., et al. (1984). Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1. (Cited on page 40.)

Jones, D. T., Buchan, D. W. A., Cozzetto, D., and Pontil, M. (2011). Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190. (Cited on page 77.)

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233. (Cited on page 106.)

Joshi, C. K. (2025). Transformers are graph neural networks. (Cited on page 118.)

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589. (Cited on page 52.)

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. (Cited on page 51.)

Karpathy, A. (2015). char-rnn. `https://github.com/karpathy/char-rnn`. (Cited on page 115.)

Karpathy, A. (2022). NanoGPT. `https://github.com/karpathy/nanoGPT`. (Cited on page 129.)

Khan, M. E. and Rue, H. (2023). The bayesian learning rule. *J. Mach. Learn. Res.*, 24(1). (Cited on page 38.)

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. (Cited on pages 38 and 86.)

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. (Cited on pages 36 and 47.)

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. (Cited on pages 41 and 118.)

Kolmogorov, A. N. (1950). *Foundations of the Theory of Probability*. Chelsea Publishing Company, New York, 1st english edition. Translated from the Russian original "Grundbegriffe der Wahrscheinlichkeitsrechnung". (Cited on page 27.)

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc. (Cited on page 41.)

Kumasaka, N., Rostom, R., Huang, N., Polanski, K., Meyer, K., Patel, S., Boyd, R., Gomez, C., Barnett, S., Panousis, N., et al. (2021). Mapping interindividual dynamics of innate immune response at single-cell resolution. *bioRxiv*. (Cited on pages 22, 186, 187, and 189.)

Lalchand, V., Ravuri, A., and Lawrence, N. D. (2022). Generalised gplvm with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, pages 7841–7864. PMLR. (Cited on pages 22 and 187.)

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press. (Cited on page 81.)

Lawrence, N. D. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816. (Cited on pages 17, 47, 48, 50, 51, 68, 70, 72, and 108.)

Lawrence, N. D. (2012). A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *J. Mach. Learn. Res.*, 13(1):1609–1638. (Cited on pages 79, 80, and 82.)

Lawrence, N. D. and Quiñonero Candela, J. (2006). Local distance preservation in the gp-lvm through back constraints. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 513–520, New York, NY, USA. Association for Computing Machinery. (Cited on page 106.)

Lazar, N. H., Celik, S., Chen, L., Fay, M. M., Irish, J. C., Jensen, J., Tillinghast, C. A., Urbanik, J., Bone, W. P., Gibson, C. C., and Haque, I. S. (2024). High-resolution genome-wide mapping of chromosome-arm-scale truncations induced by crispr–cas9 editing. *Nature Genetics*, 56(7):1482–1493. (Cited on pages 170, 172, 174, and 176.)

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc. (Cited on pages 19, 59, 100, 103, and 190.)

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130. (Cited on pages 77 and 164.)

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498. (Cited on page 81.)

Liu, D. and Yu, J. (2009). Otsu method and k-means. In *2009 Ninth International Conference on Hybrid Intelligent Systems*, volume 1, pages 344–349. (Cited on pages 43 and 168.)

Loh, P.-L. and Bühlmann, P. (2013). High-dimensional learning of linear causal networks via inverse covariance estimation. (Cited on page 81.)

Luecken, M. D. and Theis, F. J. (2019). Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746. (Cited on pages 14, 16, and 55.)

Lyu, R., Qiao, P., Kiselev, V., Andrews, T., Westoby, J., Büttner, M., Lee, J., Polanski, K., Müller, S. Y., Madissoon, E., Ballereau, S., Primo, M. D. N. L., Martinez Nunez, R., Hemberg, M., and McCarthy, D. J. (2019). Trajectory inference. Online workshop/course notes (MIG 2019 scRNA-seq workshop), SVI Bioinformatics and Cellular Genomics. Part of the *Analysis of single cell RNA-seq data* course material. (Cited on page 189.)

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., and McCarroll, S. A. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214. (Cited on page 86.)

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. (Cited on page 175.)

Mao, Q., Wang, L., Goodison, S., and Sun, Y. (2015). Dimensionality reduction via graph structure learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 765–774, New York, NY, USA. Association for Computing Machinery. (Cited on page 156.)

Masuyama, Y., Yatabe, K., Koizumi, Y., Oikawa, Y., and Harada, N. (2019). Deep griffin-lim iteration. (Cited on pages 37 and 157.)

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform manifold approximation and projection for dimension reduction. (Cited on page 58.)

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29287–29303. Curran Associates, Inc. (Cited on page 163.)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. (Cited on page 59.)

Miller, E. (2023). Attention is off by one. `https://www.evanmiller.org/attention-is-off-by-one.html`. Blog post. (Cited on page 126.)

Minka, T. (1997). Old and new matrix algebra useful for statistics. In *Online*. (Cited on page 71.)

Mnih, A. and Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc. (Cited on page 47.)

Mohamed, S. and Lakshminarayanan, B. (2017). Learning in implicit generative models. (Cited on page 43.)

Mostowsky, P., Dutordoir, V., Azangulov, I., Jaquier, N., Hutchinson, M. J., Ravuri, A., Rozo, L., Terenin, A., and Borovitskiy, V. (2024). The geometrickernels package: Heat and matérn kernels for geometric learning on manifolds, meshes, and graphs. (Cited on pages 23 and 189.)

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. (Cited on pages 26 and 201.)

Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press. (Cited on pages 30, 43, and 167.)

Murphy, K. P. (2023). *Probabilistic Machine Learning: Advanced Topics*. MIT Press. (Cited on pages 16, 46, 49, and 197.)

Nakamura, H., Okada, M., and Taniguchi, T. (2023). Representation uncertainty in self-supervised learning as variational inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16484–16493. (Cited on pages 65, 115, and 122.)

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? (Cited on page 44.)

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. (Cited on pages 58 and 196.)

Nikolentzos, G., Siglidis, G., and Vazirgiannis, M. (2022). Graph kernels: A survey. *J. Artif. Int. Res.*, 72:943–1027. (Cited on pages 40 and 162.)

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66. (Cited on pages 43 and 168.)

Pasad, A., Chou, J.-C., and Livescu, K. (2022). Layer-wise analysis of a self-supervised speech representation model. (Cited on page 41.)

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. (Cited on page 72.)

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510. (Cited on page 71.)

Pham, D. T. and Garat, P. (1997). Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Transactions on Signal Processing*, 45(7):1712–1725. (Cited on page 49.)

Pietal, M. J., Bujnicki, J. M., and Kozlowski, L. P. (2015). Gdfuzz3d: a method for protein 3d structure reconstruction from contact maps, based on a non-euclidean distance function. *Bioinformatics*, 31(21):3499–3505. (Cited on page 77.)

Poličar, P. G., Stražar, M., and Zupan, B. (2024). opentsne: A modular python library for t-sne dimensionality reduction and embedding. *Journal of Statistical Software*, 109(3):1–30. (Cited on page 86.)

Popper, K. R. (1959). *The logic of scientific discovery.* The logic of scientific discovery. Basic Books, Oxford, England. (Cited on page 26.)

Probst, D. and Reymond, J.-L. (2018). A probabilistic molecular fingerprint for big data settings. *Journal of Cheminformatics*, 10(1):66. (Cited on page 161.)

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. (2018). Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 459–467, New York, NY, USA. Association for Computing Machinery. (Cited on page 190.)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. (Cited on page 130.)

Raftery, A. E. (1995). Bayesian model selection in social research. *Sociological methodology*, pages 111–163. (Cited on page 169.)

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc. (Cited on pages 51 and 194.)

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press. (Cited on pages 29, 31, 50, and 101.)

Ravuri, A., Andersson, T. R., Kazlauskaite, I., Tebbutt, W., Turner, R. E., Hosking, J. S., Lawrence, N. D., and Kaiser, M. (2022a). Ice core dating using probabilistic programming. (Cited on pages 23, 183, and 185.)

Ravuri, A., Cooper, E., and Yamagishi, J. (2024a). Uncertainty as a predictor: Leveraging self-supervised learning for zero-shot mos prediction. In *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 580–584. (Cited on pages 22 and 163.)

Ravuri, A., Dann, E., Lalchand, V., Kumasaka, N., Sumanaweera, D., Lindeboom, R. G., Madad, S., Teichmann, S., and Lawrence, N. D. (2022b). Modelling technical and biological effects in scrna-seq data with scalable gplvms. In *Machine Learning in Computational Biology*, pages 46–60. PMLR. (Cited on pages 22, 98, 187, and 189.)

Ravuri, A. and Lawrence, N. D. (2024). Towards one model for classical dimensionality reduction: A probabilistic perspective on umap and t-sne. (Cited on pages 23 and 84.)

Ravuri, A. and Lawrence, N. D. (2025a). Protein language model zero-shot fitness predictions are improved by inference-only dropout. (Cited on pages 21 and 164.)

Ravuri, A. and Lawrence, N. D. (2025b). Transformers as unrolled inference in probabilistic laplacian eigenmaps: An interpretation and potential improvements. (Cited on pages 23 and 114.)

Ravuri, A., Muir, J., and Lawrence, N. D. (2024b). On feature learning for titi monkey activity detection. (Cited on pages 22 and 160.)

Ravuri, A., Ulicna, K., Osea, J., Donhauser, K., and Hartford, J. (2025). Weakly supervised latent variable inference of proximity bias in crispr gene knockouts from single-cell images. In *Learning Meaningful Representations of Life (LMRL) Workshop at ICLR 2025*. (Cited on pages 22, 170, 171, 173, 174, and 176.)

Ravuri, A., Vargas, F., Lalchand, V., and Lawrence, N. D. (2023). Dimensionality reduction as probabilistic inference. In *Fifth Symposium on Advances in Approximate Bayesian Inference*. (Cited on pages 20, 23, 65, 87, and 156.)

Rezende, D. J. and Mohamed, S. (2016). Variational inference with normalizing flows. (Cited on page 187.)

Richemond, P. H., Tam, A., Tang, Y., Strub, F., Piot, B., and Hill, F. (2023). The edge of orthogonality: A simple view of what makes byol tick. (Cited on page 123.)

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754. (Cited on page 162.)

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326. (Cited on page 79.)

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252. (Cited on page 115.)

Sarkka, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61. (Cited on page 185.)

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. (Cited on page 163.)

Schoelkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. (2012). On causal and anticausal learning. (Cited on page 39.)

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg. Springer Berlin Heidelberg. (Cited on pages 77 and 163.)

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. (Cited on page 80.)

Skinnider, M. A., Squair, J. W., and Foster, L. J. (2019). Evaluating measures of association for single-cell transcriptomics. *Nature Methods*, 16(5):381–386. (Cited on page 82.)

Stephenson, E., Reynolds, G., Botting, R. A., Calero-Nieto, F. J., Morgan, M. D., Tuong, Z. K., Bach, K., Sungnak, W., Worlock, K. B., Yoshida, M., et al. (2021). Single-cell multi-omics analysis of the immune response in covid-19. *Nature medicine*, 27(5):904–916. (Cited on page 186.)

Stirling, D. R., Swain-Bowden, M. J., Lucas, A. M., Carpenter, A. E., Cimini, B. A., and Goodman, A. (2021). Cellprofiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*, 22(1):433. (Cited on page 173.)

Svensson, V., Gayoso, A., Yosef, N., and Pachter, L. (2020). Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, 36(11):3418–3421. (Cited on pages 186 and 189.)

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323. (Cited on page 77.)

Tian, Y., Chen, X., and Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. (Cited on page 123.)

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622. (Cited on pages 47, 68, 70, and 71.)

Tong, Y. L. (1990). *Statistical Computing Related to the Multivariate Normal Distribution*, pages 181–201. Springer New York, New York, NY. (Cited on page 53.)

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419. (Cited on page 76.)

Tseng, W.-C., Kao, W.-T., and yi Lee, H. (2022). Ddos: A mos prediction framework utilizing domain adaptive pre-training and distribution of opinion scores. (Cited on page 164.)

Turner, R. and Sahani, M. (2007). A maximum-likelihood interpretation for slow feature analysis. *Neural Computation*, 19(4):1022–1038. (Cited on page 46.)

Uhlig, H. (1994). On Singular Wishart and Singular Multivariate Beta Distributions. *The Annals of Statistics*, 22(1):395 – 405. (Cited on page 69.)

Van Assel, H., Espinasse, T., Chiquet, J., and Picard, F. (2022). A probabilistic graph coupling view of dimension reduction. *Advances in Neural Information Processing Systems*, 35:10696–10708. (Cited on pages 23 and 72.)

van de Meent, J.-W., Paige, B., Yang, H., and Wood, F. (2021). An introduction to probabilistic programming. (Cited on pages 15 and 33.)

van den Oord, A., Li, Y., and Vinyals, O. (2019). Representation learning with contrastive predictive coding. (Cited on page 163.)

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605. (Cited on pages 57, 64, 65, 199, and 200.)

van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc Biol Sci*, 265(1394):359–366. (Cited on page 49.)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. (Cited on pages 65, 115, and 118.)

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. (Cited on page 118.)

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416. (Cited on pages 80 and 81.)

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. (Cited on page 118.)

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305. (Cited on page 193.)

Weinberger, K. Q., Sha, F., and Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 106. (Cited on page 78.)

Weisfeiler, B. and Lehman, A. A. (1968). A Reduction of a Graph to a Canonical Form and an Algebra Arising During This Reduction. *Nauchno-Technicheskaya Informatsia*, Ser. 2(N9):12–16. (Cited on page 162.)

Weng, L. (2018). Attention? attention! *lilianweng.github.io*. (Cited on page 118.)

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25. (Cited on page 67.)

Williams, C. K. I. and Agakov, F. V. (2002). Products of Gaussians and Probabilistic Minor Component Analysis. *Neural Computation*, 14(5):1169–1182. (Cited on pages 68, 72, and 73.)

Winstrup, M. (2011). *An Automated Method for Annual Layer Counting in Ice Cores*. PhD thesis, University of Copenhagen. (Cited on page 183.)

Winstrup, M. (2016). A hidden markov model approach to infer timescales for high-resolution climate archives. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 4053–4060. AAAI Press. (Cited on page 183.)

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? (Cited on page 162.)

Young, G. and Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22. (Cited on page 76.)

Yu, Y., Buchanan, S., Pai, D., Chu, T., Wu, Z., Tong, S., Haeffele, B. D., and Ma, Y. (2023). White-box transformers via sparse rate reduction. (Cited on pages 20, 65, 115, 118, 120, and 123.)

Zhao, S., Ravuri, A., Lalchand, V., and Lawrence, N. D. (2024). Scalable amortized gplvms for single cell transcriptomics data. (Cited on pages 22 and 189.)

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):14049. (Cited on page 86.)

APPENDIX A

Additional content

The appendix contains additional scientific case-studies as mentioned in the main thesis, along with additional arguments supporting the main content. A summary of the scientific case-studies of the appendix is shown in table A.1.

Section	Case study	Narrative contribution
Appendix A.4	monkey call detection	We show how highly compact human-defined acoustic features can be effective for use-cases peripheral to their intended use (human speech representation), enabling small but effective models.
Appendix A.5	graph kernels	We show how feature function abstractions, specifically those implementing graph isomorphism tests, can be used to implement graph kernels.
Appendix A.6, A.7	speech and protein scalar property prediction	We show how classifier confidence is a proxy for zero-shot out-of-domain detection, and that this is improved using Monte-Carlo dropout.
Appendix A.11	proximity bias	We show how a model can drive pseudo-interpretable data analysis (to recover characteristics of and identify certain cells) in a severe data-noise regime (cell images), when model components are constrained using a scientific hypothesis. Also, as above, we show how an interpretable latent (visually striking perturbations) can be recovered from classifier confidence.
Appendix A.12	ice-cores	We show how HMMs can be significantly constrained using a priori knowledge relating to monotonicity of the latent (time), enabling interpretable recovery.
Appendix A.13	scRNA-seq	We show that various data and model constraints in sparse GPLVMs, and initialisation choices drive interpretability.

Table A.1: Summary of scientific case studies and their contributions.

A.1 A MAP perspective on DRTree

In this section, we exemplify the interpretation of an objective as a log-likelihood that involves a discrete random variable, as mentioned in section 2.2.[1] The DRTree (Mao et al., 2015) algorithm finds a minimal spanning tree representation of the data, as well as projecting the data onto a low-dimensional space. The objective is written as follows,

$$\begin{aligned}\mathcal{L} &= \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|^2 + \frac{\lambda}{2}\sum_{ij} b_{ij}\|\mathbf{W}^T\mathbf{X}_{i:} - \mathbf{W}^T\mathbf{X}_{j:}\|^2 \\ &= \mathrm{tr}((\mathbf{Y} - \mathbf{X}\mathbf{W})^T(\mathbf{Y} - \mathbf{X}\mathbf{W})) + \mathrm{tr}(\lambda\mathbf{L}\mathbf{X}\mathbf{X}^T),\end{aligned}$$

where the second step follows as $\mathbf{W}\mathbf{W}^T$ is constrained to be $\mathbf{I}$. We see that the objective is approximately a negative log posterior, assuming,

$$\begin{aligned}\mathbf{Y}|\mathbf{X}, \mathbf{W} &\sim \mathcal{MN}(\mathbf{X}\mathbf{W}, \mathbf{I}, \mathbf{I}) \\ \mathbf{X}|\mathbf{L} &\sim \mathcal{MN}(\mathbf{0}, (\lambda\mathbf{L} + \beta\mathbf{I})^{-1}, \mathbf{I}) \\ \mathbf{L} &\sim \text{Uniform over tree-structured graphs}\end{aligned}$$

for small β and such that $\mathbf{W} \in \mathrm{SO}(d_q)$. The optimisation occurs with respect to the joint distribution, in an alternating manner. To optimise over $\mathbf{L}$ given the other parameters, we maximise,

$$\log \underbrace{p(\mathbf{Y}|\mathbf{X}, \mathbf{W})}_{c}\, p(\mathbf{X}|\mathbf{L})\, \underbrace{p(\mathbf{L})}_{\propto 1},$$

which, ignoring the log det term, is simply the cost of traversing all adjacencies $\sum_{ij} b_{ij}\|\mathbf{W}^T\mathbf{X}_i - \mathbf{W}^T\mathbf{X}_j\|^2$. Kruskal's algorithm finds a minimal spanning tree for the data, with known graph weights, hence we use this algorithm to find $\mathbf{L}$ at each step of the alternating optimisation, given the current estimates of $\mathbf{X}, \mathbf{W}$.

A.2 The Griffin-Lim algorithm

Here, we describe the GLA case-study of section 2.2 in detail. In speech analysis, a common visual tool to analyse frequencies and how they change over time, is a **spectrogram**. This is

[1] We introduced this interpretation in **Ravuri et al. (2023)**.

typically calculated as the element-wise absolute value of a complex valued matrix known as the short-time Fourier transform (STFT) matrix. Within a STFT operation, a signal is first segmented, and a Fourier transform is computed for each segment. As an illustration,

$$\begin{bmatrix} 1 \\ \vdots \\ 8 \end{bmatrix} \xrightarrow{\text{window}} \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \underset{D\times}{\xrightarrow{\text{DFT}}} \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix},$$

where $\mathbf{D}$ is the discrete Fourier transform matrix. In the example above, the number of points hopped between windows is two, and the number of points in each window is four. The inverse STFT operation obtains the real part of the inverse Fourier transform of the STFT matrix, and then performs an overlap add operation, averaging together different windows' estimates of a particular signal point. As an illustration,

$$\text{STFT} \underset{D^\dagger\times}{\xrightarrow{\mathcal{F}^{-1}}} \begin{bmatrix} 1 & c_2 & e_2 \\ 2 & d_2 & f_2 \\ c_1 & e_1 & 7 \\ d_1 & f_1 & 8 \end{bmatrix} \xrightarrow{\text{OLA}} \{1, 2, \tfrac{1}{2}c_1 + \tfrac{1}{2}c_2, \ldots\}$$

The **Griffin-Lim algorithm (GLA)** of Griffin and Lim (1984), given the absolute value of an STFT $\mathbf{S}$, tries to find a complex matrix $\mathbf{X} = \mathbf{S} \odot \exp(\boldsymbol{\theta} i)$ such that the norm,

$$\mathcal{L} = \|\mathcal{G}(\mathcal{G}^\dagger(\mathbf{X})) - \mathbf{X}\|_F^2,$$

is minimised (Masuyama et al., 2019). Here, $\mathcal{G}$ corresponds to the short-term Fourier transform (STFT) operation, and $\mathcal{G}^\dagger$ corresponds to its pseudo-inverse.

For real-valued signals, the projection (taking the inverse-Fourier transform and then a Fourier transform) applied to spectrum should result in the starting spectrum, due to consistency, i.e. we expect $\mathcal{G}(\mathcal{G}^\dagger(\mathbf{X})) = \mathbf{X}$. Generally, at a random initialisation of $\boldsymbol{\theta}$, the norm is non-zero due to redundant information within the STFT, as an audio sample at a specific time point contributes information to multiple windows. Therefore, not all phase matrices correspond to typical audio signals (the inverse Fourier transform must be real, the first phase is 0 for every time window, and the mirrored frequencies are the complex conjugates of the preceding

frequencies, etc.). In other words, valid $\boldsymbol{\theta}$ occupies a small subspace of $[0, 2\pi]^{n_w \times w}$, where n_w is the number of time points per window, and w is the number of windows. GLA therefore, tries to minimise the disagreement between the two sides of the previous equation, by minimising the residual. The algorithm can produce perceptible speech simply from spectrogram images, and was used before the advent of large neural-vocoders. Phase reconstruction problems also appear in biological imaging.

Define $\mathbf{v} = \text{vec}(\text{STFT}(\mathbf{a}))$ and $\mathbf{D}_k = \mathbf{I}_w^\intercal \otimes \mathbf{D}$. Further, let $\mathbf{W}$ be a repetition matrix that repeats elements of a vector such that every n_w elements of the transformed vector form a window. $\mathbf{W}$ has the form,

$$\mathbf{W} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & \cdots & \end{bmatrix} \text{ and } \mathbf{W}_i = \begin{bmatrix} 1 & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 0.5 & & 0.5 & & & & & & \\ & & & 0.5 & & 0.5 & & & & & \\ & & & & & & 0.5 & & 0.5 & & \\ & & & & & & & 0.5 & & 0.5 & \end{bmatrix}.$$

Then,

$$\mathbf{v} = \mathcal{G}(\mathbf{a}) = \text{vec}(\mathbf{D} \times \text{to\_window}(\mathbf{a}) \times \mathbf{I}_w) = \mathbf{D}_k \mathbf{W} \mathbf{a}. \qquad \text{vec-trick}$$

Similarly, if $\mathbf{W}_i$ represents a matrix that performs the overlap-addition on a vector, one can also define the pseudo-inverse as a matrix operation,

$$\mathbf{a} = \mathcal{G}^\dagger(\mathbf{v}) = \mathbf{W}_i \Re\left(\mathbf{D}_k^\dagger \mathbf{v}\right).$$

Let $\mathbf{v} = \text{vec}(\mathbf{S} \odot \exp(\boldsymbol{\theta} i))$, $\tilde{\mathbf{a}} = \mathbf{D}_k^\dagger \mathbf{v}$ (the signal before taking the real part and overlap-adding) and $\boldsymbol{\Pi} = \mathbf{W}\mathbf{W}_i$. $\boldsymbol{\Pi}$ is symmetric and idempotent. Observe that the GLA objective can be written as,

$$\begin{aligned} \mathcal{L} &= \|\mathbf{D}_k \mathbf{W} \mathbf{W}_i \Re\left(\mathbf{D}_k^\dagger \mathbf{v}\right) - \mathbf{v}\|_F^2 \\ &= \|\mathbf{D}_k \left(\boldsymbol{\Pi} \Re\left(\tilde{\mathbf{a}}\right) - \tilde{\mathbf{a}}\right)\|_F^2 \\ &= \left(\boldsymbol{\Pi} \Re\left(\tilde{\mathbf{a}}\right) - \tilde{\mathbf{a}}\right)^\dagger \left(\boldsymbol{\Pi} \Re\left(\tilde{\mathbf{a}}\right) - \tilde{\mathbf{a}}\right) \\ &= \Re(\tilde{\mathbf{a}})^T (\mathbf{I} - \boldsymbol{\Pi}) \Re(\tilde{\mathbf{a}}) + \Im(\tilde{\mathbf{a}})^T \Im(\tilde{\mathbf{a}}). \end{aligned} \tag{A.1}$$

This is the log-density[2] (up to constants) of the improper Gaussian with precision $\mathbf{P}$,

$$\begin{bmatrix} \Re(\tilde{\mathbf{a}}) \\ \Im(\tilde{\mathbf{a}}) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P} = \begin{bmatrix} (\mathbf{I} - \mathbf{\Pi}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\right) \tag{A.2}$$

Consider the covariance that is implied by adding a small jitter matrix to the precision,

$$\begin{aligned} \mathbf{C} &= (\mathbf{P} + \delta\mathbf{I})^{-1} \\ &= \begin{bmatrix} (1+\delta)\mathbf{I} - \mathbf{\Pi} & \mathbf{0} \\ \mathbf{0} & (1+\delta)\mathbf{I} \end{bmatrix}^{-1}. \end{aligned}$$

The covariance corresponding to the imaginary part is simply $\mathbf{I}/(1+\delta)$. The covariance corresponding to the real part is,

$$\begin{aligned} C_{\Re(\tilde{\mathbf{a}})\Re(\tilde{\mathbf{a}})} &= ((1+\delta)\mathbf{I} - \mathbf{\Pi})^{-1} \\ &= \frac{1}{1+\delta}\left[\mathbf{I} - \frac{1}{1+\delta}\mathbf{W}\left(\frac{1}{1+\delta}\mathbf{I} - \mathbf{I}\right)^{-1}\mathbf{W}_i\right] \qquad \mathbf{W}_i\mathbf{W} = \mathbf{I} \\ &= \frac{1}{1+\delta}\left[\mathbf{I} + \frac{1}{\delta}\mathbf{\Pi}\right]. \end{aligned}$$

Therefore, we show that the variance for the real parts of the audio dominates the imaginary parts as $\delta \to 0$, therefore showing that the model is statistically meaningful.

A.3 Effect of optimiser choice on inference

In this section, as mentioned in section 2.2, we show that optimiser choice leads to difference in sparsity in recovered solutions in a logistic regression model. Concretely, in fig. A.1, we show that running gradient descent used for logistic regression, to construct a linear classifier that separates two Gaussian-distributed blobs, results in sparse solutions more often than when the Adam optimiser is used.

[2]The interpretation of the loss as a complex-normal likelihood fails to provide a consistent set of parameters, as the complex normal imposes more conditions on the structure of the covariance between blocks corresponding to $\Re(\mathbf{v})$ and $\Im(\mathbf{v})$ than a multivariate normal would. In other words, not all PSD matrices can correspond to complex-normal precisions.

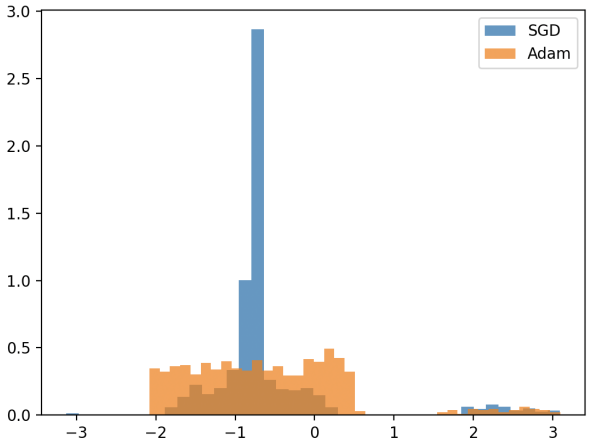


Figure A.1: Results of a logistic regression, i.e. the optimisation of the likelihood of a model $y_i|\mathbf{x}_i \sim \text{Bernoulli}(\sigma(\alpha x_1 + \beta x_2))$ fit to two linearly separable Gaussian blobs corresponding to two classes, in two dimensions. **Using SGD results in sparse solutions more often with respect to Adam.** Both optimisers were initialised randomly using a learning rate of 0.01 and run for 1000 epochs. The statistic visualised is the angle of the separator $\theta = \arctan(\alpha/\beta)$, and $\theta = \pi$ corresponds to a sparse solution.

A.4 Vocal activity detection in coppery titi monkeys

In this section, which we briefly mention in section 2.3.1, we show a simple bioacoustics conservation case-study involving auditory representations which offer a low-dimensional frequency-domain representation of speech.[3]

The problem Actively collected acoustic data can provide valuable insights into the ecological, behavioural, and health aspects of an animal species. We focus on Coppery titi monkeys (Plecturocebus cupreus), an accessible species in local zoos. Manual processing of large volumes of acoustic data is challenging and time-consuming. The primary challenge is the development of a framework for voice activity detection using large volumes of passively collected audio containing titi monkey vocalisations, as relatively small amounts of expert-annotated data is available. Traditional methods (such as identifying segments that meet energy thresholds after passing the audio through a band-pass filter guided by the range of frequencies at which the monkeys are known to call) result in high false positives, for example, from birds that have vocal activity in the same frequency regions.

Available data The dataset contains a large amount of passively collected audio data gathered in titi monkey enclosures, with a small proportion annotated by an expert. The annotations

[3]This section is based on **Ravuri et al. (2024b)**.

include the type of call and the start-stop times for those calls.

Problem representation We treat the problem as a simple classification due to the availability of enough training data (about 3300 calls of about 0.32s duration, across 3 zoos). We assume the model,

$$y_{[t,t+\delta)}|a_{[t,t+\delta)} \sim \text{Bernoulli}(f(a_{[t,t+\delta)})),$$

where f is a function acting on a segment of audio $a_{[t,t+\delta)}$, and $y_{[t,t+\delta)}$ is a label indicating whether or not the time segment contains a monkey call. We find that MFCC (Mel-frequency cepstral coefficient) representations of audio, primarily developed for human speech and introduced in Davis and Mermelstein (1980), are effective representations for our uses, despite being relatively low-dimensional and simple. MFCCs involve a lossy frequency-domain transformation of segments of audio (a discrete cosine transform of a perceptual-mapping of the segment's log-power spectrum), typically reducing a segment of a few thousand points representing the amplitudes of signals into as few as forty points representing the spectral information as it pertains to human-perceptible frequencies. Using these representations, we can then parameterise our function f as a simple LSTM that acts on these base representations, and then train the model using maximum-likelihood, optimising the few parameters of the LSTM. We use a three-layer LSTM with sixteen hidden features that are linearly compressed to one scalar per window for the purposes of classification. Figure A.2 illustrates a segment of the data, showing that expert annotations and model predictions, and that the MFCCs preserve the clusterability of distinct call types.

Future work Pre-training of wav2vec2-style (Baevski et al., 2020) models using the large amounts of available audio may yield better performance of our methods, as well as the use of source separation models for preprocessing the audio as part of a data-cleaning step.

A.5 Graph embeddings

One way to featurise graph inputs, as mentioned in section 2.3.1, is to consider bit vectors consisting of substructures of the graph, visualised for a molecule in fig. A.3. As an example, molecular fingerprints, such as the SECFP (Probst and Reymond, 2018), aim to (at least in an approximate sense) solve the molecular isomorphism problem while offering low-dimensional

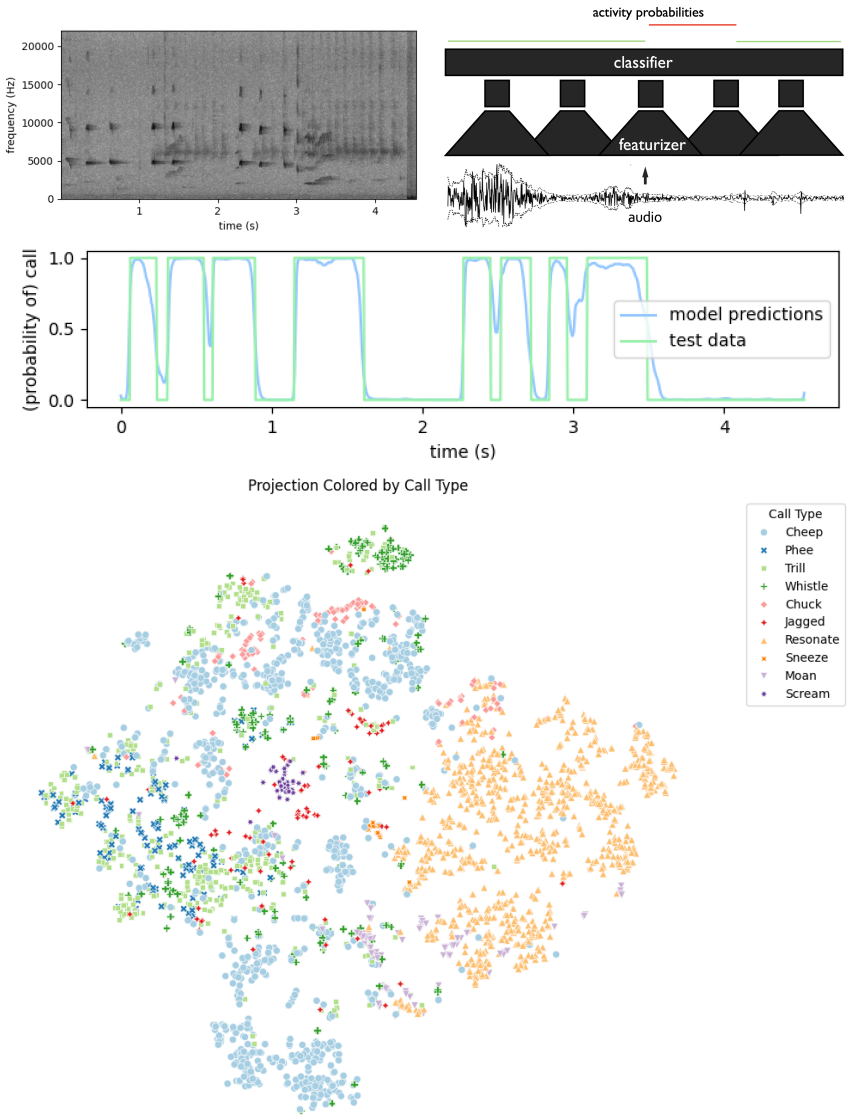


Figure A.2: Call detection case study. **Top left:** An illustration of a segment of audio as a spectrogram, used by domain experts to recognise and label calls. **Top right:** audio is compressed to its MFCC representation through a feature function, that is highly compressed, and can be used by a RNN that returns probability that an audio segment contains a call. **Center:** output of the RNN plotted against true labels. **Bottom:** call types visualised on a t-SNE dimensionality reduction of stretched and flattened call MFCCs, showing that MFCCs are capable of separating them.

representations of chemicals as bit vectors. Roughly, the molecular isomorphism problem, a case of the graph isomorphism problem, seeks to identify if two molecules "are the same" (Rogers and Hahn, 2010). In graph literature, a variety of similar methods exist to tackle the graph isomorphism problem more generally, such as the Weisfeiler-Lehman approach of Weisfeiler and Lehman (1968), which involves the construction of a feature vector which aids the comparison of two graphs Nikolentzos et al. (2022). Graph neural networks can be seen to generalise such featurisations (Xu et al., 2019).

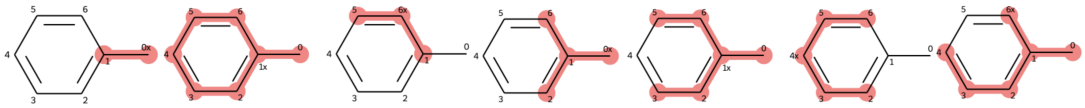


Figure A.3: An example of a feature function: a chemical can be represented as a bit vector, with every bit representing whether a substructure illustrated above is present.

Functions that featurise such inputs can be useful as they can be used to define kernel functions (that measure similarities between data points), used by model constructions such as Gaussian process regressors. Feature functions $\phi : \mathcal{G} \mapsto \mathbb{R}^m$ acting on graphs specifically can define kernels through the inner product, $k_{ij} = \langle \phi(\mathcal{G}_i), \phi(\mathcal{G}_j) \rangle$, leading to uncertainty measures over spaces of graphs. They can also provide a way with which to perform dimensionality reduction on graphs, using methods such as kernel PCA (Schölkopf et al., 1997).

Such interpretations are helpful in software settings, as to implement graph kernels, one need only implement a feature function, and use existing abstractions (e.g. a `LinearKernel` within a `GaussianProcess` implementation). In **Griffiths et al. (2023)**, we show how simple software can be written with this principle in mind within a chemical property prediction library.

A.6 Mean opinion score prediction

Many contrastive models of audio learn to classify next (latent) token from tokens randomly selected from the audio sequence. Some model frameworks follow the approach of NCE, with others following strategies similar to NCE but modifying it with information-theoretic arguments. Frameworks that describe such methods for SSL can be found in van den Oord et al. (2019); Schneider et al. (2019); Baevski et al. (2020).

We were inspired by methods in biology that use model uncertainty as a proxy scalar for property prediction, for example Meier et al. (2021), for our work in **Ravuri et al. (2024a)**. We found that out-of-the-box pretrained SSL models of human audio (specifically wav2vec and wav2vec2, introduced in Schneider et al. (2019); Baevski et al. (2020)) can also be used for zero-shot prediction of audio quality measured by human listeners, known as the mean-opinion score (MOS). This can be seen as an out-of-domain prediction problem, as many pretrained SSL models are trained on clean audio. The problem is tackled by measuring (a proxy of) uncertainty, such as entropy over the logits of the SSL models given a specific audio sequence

as input, and using this proxy as a predictor for audio quality, with the idea that higher model uncertainty should correspond to lower quality, as lower-quality audio samples would be out-of-distribution.

Alongside the entropy, a mean of the log-probabilities is a proxy for model uncertainty as, due to the requirement that the distribution be normalised, a case of high average log-probabilities corresponds to high model uncertainty. Consider a categorical distribution with probabilities $\mathbf{p}$; as with the entropy, the quantity $\sum_i^K \log p_i$ is uniquely maximised when $\mathbf{p}$ is uniform—i.e. when the average log-probabilities are relatively high.

Graphically, the model below is hypothesised to result in a relatively good fit,

$$y_i \longleftarrow \log \hat{p}(x_i)$$

We found that smaller models with fewer negatives (i.e. wav2vec) performed better, although the larger models could be handicapped (via feature dropout), using Monte-Carlo simulation to obtain average logits under dropout (similar to the approach of Monte-Carlo dropout, Gal and Ghahramani (2016), with the caveat that our models were not trained with dropouts to begin with) to raise zero-shot performance. Performing some pre-training steps on the evaluation data, seen as domain-adaptive pre-training, also used in Tseng et al. (2022), increased performance.

Figure A.4 shows results of using the ideas above, obtained on the datasets provided by Huang et al. (2022); Cooper et al. (2023). The plot of SRCC (measured against the spoken English MOS ratings) of our dropout-handicapped wav2vec2 Base 960H model against dropout probability shows that performance **increases** initially as the model is handicapped.

A.7 Biochemical property prediction

As mentioned in section 2.3.3, in **Ravuri and Lawrence (2025a)**, we apply the insights of dropout-averaging noted in the previous work on MOS prediction to the problem of protein fitness prediction, showing that a protein language model (ESM-2, of Lin et al. (2023)) zero-shot performance can be improved in a similar way for fitness prediction tasks (Hsu et al., 2022). fig. A.5 illustrates that increasing dropout and performing MCD increases zero-shot performance across all models. Future work could investigate whether the calibration of

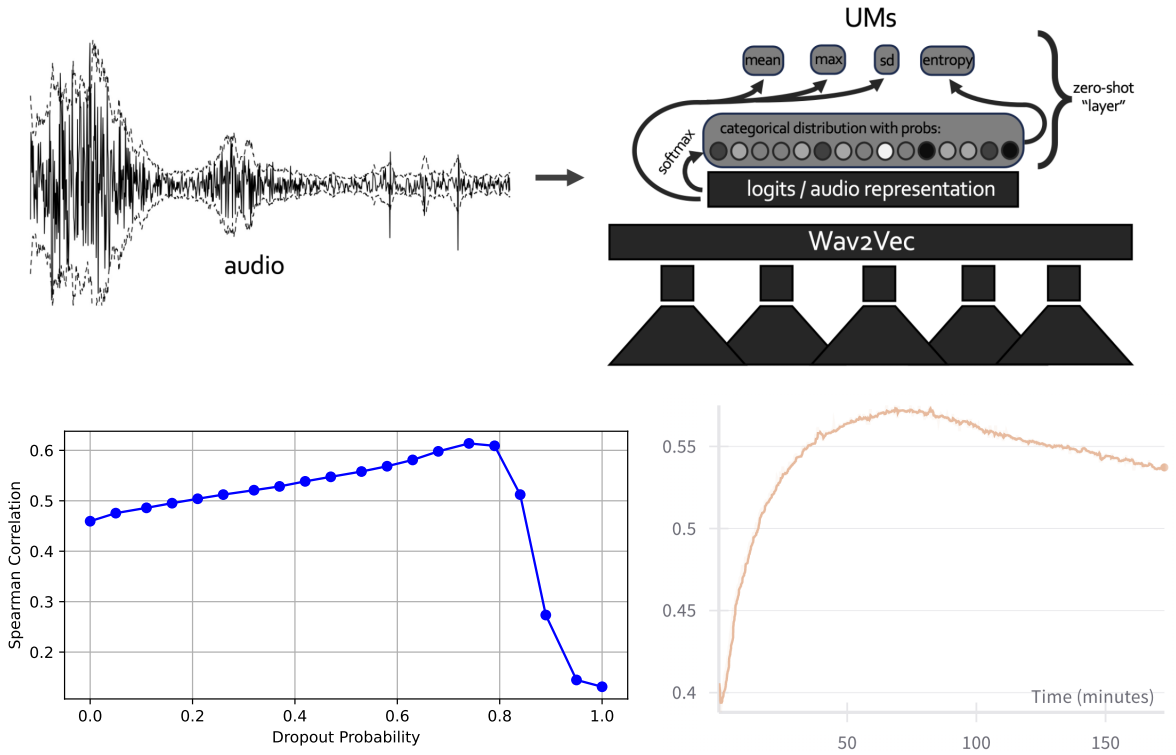


Figure A.4: Top: a graphical abstract of the method, showing that an audio sample is fed to pre-trained wav2vec models, with an uncertainty measure extracted using the logits of the model. These uncertainty measures are then used as scalar proxies for property prediction at the audio level. **Bottom left:** A graph showing that injecting a dropout layer between the featuriser and the transformer layers in the wav2vec model, and calculating the logits averaged over Monte-Carlo samples under the logits increases performance initially as the dropout probability is increased. **Bottom right:** A plot showing that model performance increases when the base wav2vec model is pre-trained on (all available) evaluation data, corresponding to domain-adaptive pre-training.

such models is indeed increased through Monte-Carlo dropout, and whether this increase in calibration driving performance boosts for scalar prediction problems.

A.8 Classifiers as density estimators: an LDA perspective

This section describes the idea of section 2.3.3 in more detail. We hypothesise that a classifier $f(\mathbf{x})$ trained on data that agrees with (or is in some way robust to) the LDA assumptions follows a rule that classifier confidence decreases on average as one traverses away from the mode of the normal distribution corresponding to that class.

Explicitly, consider a two class setup, with the random variable y taking values in $\{0, c\}$. Assuming LDA conditions, without loss of generality, there exists an affine transformation such that the empirical covariance of any data vector is $\mathbf{I}$ and such that the means of the classes sit

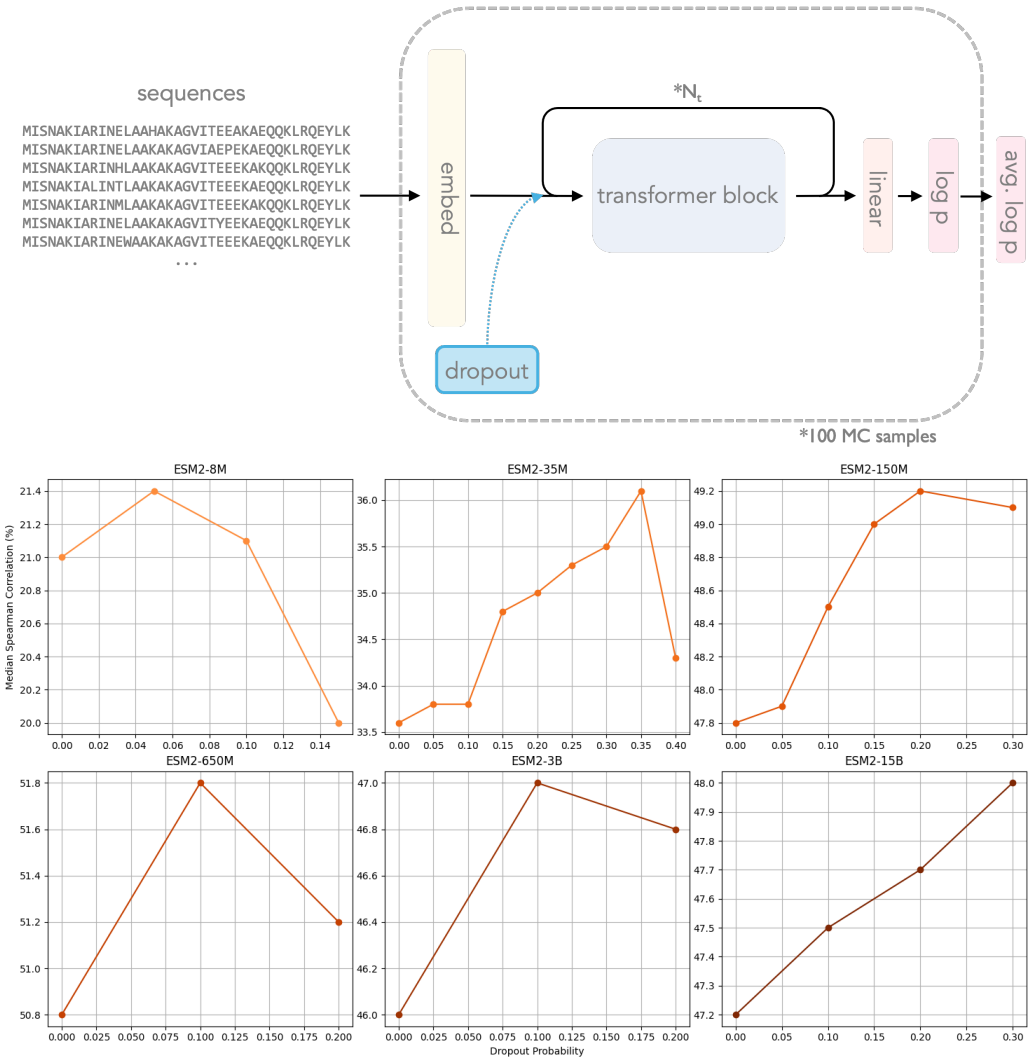


Figure A.5: Top: Graphical abstract showing that we simply introduce a dropout between the embedding layer and transformer block of a protein language model, run many forward passes through the model and average the output log probabilities. These dropout-averaged outputs are more effective for zero-shot fitness prediction, even though the model was not trained with dropout. **Bottom:** Graphs showing the zero-shot fitness performance of ESM2 with dropout added at inference-time only.

at $\mathbf{0}$ and $\boldsymbol{\mu}_c = \{\mu_c, 0, ..., 0\}$ respectively. Focusing on the class c, with mean $\boldsymbol{\mu}_c$, we hypothesise that,

$$\mathbb{E}_{\mathbf{x}:\|\mathbf{x}-\boldsymbol{\mu}_c\|^2=r^2}(\log f(\mathbf{x})) \text{ is decreasing in } r. \tag{A.3}$$

I.e. the average predicted classifier confidence drops as we move away from the cluster mode.

A sketch proof follows. Using the results of LDA in Murphy (2022), we know that,

$$\begin{aligned}
\hat{\mathbb{P}}(y = c|\mathbf{x}) &= f_c\mathbf{x} = \sigma(x_1\mu_c - \mu_c^2/2) \\
&= \sigma(z_1\mu_c + \mu_c^2/2) && \text{with } \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \\
\implies \mathbb{E}_{\mathbf{x}:\|\mathbf{x}-\mu_c\|^2=r^2}(\log f(\mathbf{x})) &= \mathbb{E}_{\mathbf{z}:\|\mathbf{z}\|^2=r^2}(\log \sigma(z_1\mu_c + \mu_c^2/2)) \\
&= \mathbb{E}_{\mathbf{u}}(\log \sigma(ru_1\mu_c + \mu_c^2/2)) && \text{with } \mathbf{u} \sim \mathcal{U} : \|\mathbf{u}\|^2 = 1.
\end{aligned}$$

We now simply show computationally in fig. A.6 that this function is decreasing in r, as the proof for this statement is tedious. The intuitive idea is that the log probability decreases towards the class boundary/linear separator faster than it increases as one moves away from the separator. Thinking of an isotropic sphere representing the data log density for a class, we can reason that on a contour of equal log-density there is a sharper fall off in the density towards the boundary than away from it.

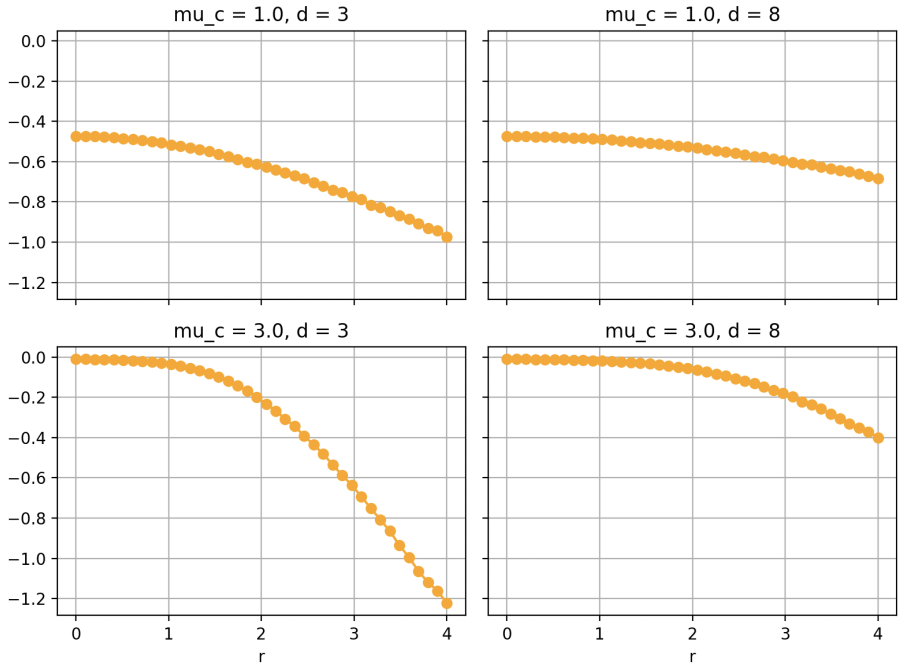


Figure A.6: Figure showing that log confidence decreases with increasing radius in LDA.

We posit that these reasons provide an explanation for phenomena where classifiers can be seen to be less confident for out-of-domain examples (i.e. where the empirical data density is low), for example, as observed in Hendrycks and Gimpel (2018).

A.9 An application of LDA in vision

An image segmentation algorithm, known as Otsu's method, introduced in Otsu (1979) and mentioned in section 2.3.3, is an algorithm that assumes that pixel intensities in an image come from two "classes" causing a bi-modality in the intensity histogram. The algorithm chooses a scalar threshold along the histogram that maximises the inter-class variance of the two groups. The method is equivalent to the k-means algorithm (Liu and Yu, 2009) and therefore is related to LDA[4].

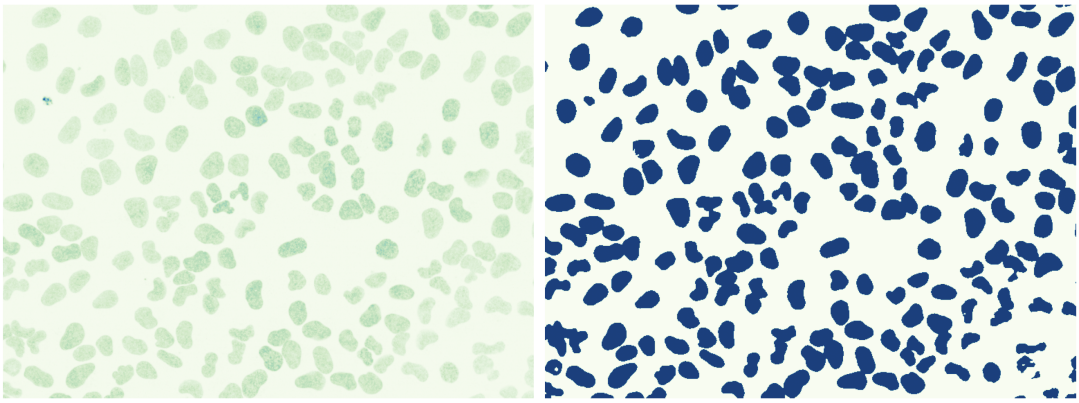


Figure A.7: Cell nuclei segmented using Otsu thresholding, which has an LDA interpretation.

We demonstrate the effectiveness of the method in segmentation of cellular nuclei from the JUMP cell painting dataset, specifically an image corresponding to the nuclear channel, of a well with cells affected by non-targeting guides used with the CRISPR-Cas9 complex. Simply applying a Gaussian filter and running the algorithm results in a clear identification of cellular nuclei, illustrated in fig. A.7.

A.10 Misspecified data distribution in GMMs

In this section, we illustrate the idea of Cai et al. (2021), mentioned in section 2.4.1. Assume that the data generating mechanism for a real world problem is non-Gaussian and is affected by larger tails than explained by a Gaussian (which is only one of the various common misspecification types that occur in practice, along with zero inflation). Assume, for this example that a data's

[4]Due to the equivalence of the k-means algorithm and EM in GMMs (Bishop, 2006).

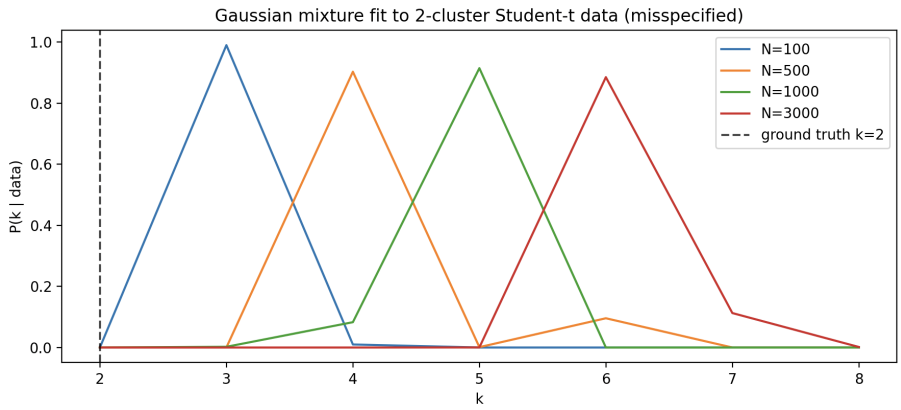


Figure A.8: An illustration showing that the posterior probabilities over the number of clusters in a GMM is maximised with more clusters than present in the data distribution (the ground truth was two clusters), as the number of data points increases, when the generating distribution is misspecified (as Gaussian, instead of multivariate-t, which was the data generating distribution).

generating mechanism is,

$$k \sim \text{Categorical}(\boldsymbol{\pi}),$$
$$\mathbf{x}|k \sim \text{t}(\nu = 3, \boldsymbol{\mu}_k, \Sigma).$$

Cai et al. (2021) show that, in such a scenario, if a GMM is assumed and if one uses a likelihood-based approach to determine the number of clusters present in the dataset, the true number is not recovered, and that the approach leads to an infinite number of components recovered as the number of data points increases.[5] The posterior probability of the number of classes in such an example is visualised in fig. A.8. These probabilities are calculated using a geometric prior on the number of classes, and using the Bayesian information criterion (BIC, which is simply a linear function of the maximal log likelihood value, evaluated at the optimal mean and covariance parameters) as an approximation to the model evidence,

$$p(k|\text{data}) \propto \exp(-\text{BIC}(k)/2) * \text{Geom}(k|r = 0.01).$$

This approximation is from Raftery (1995).

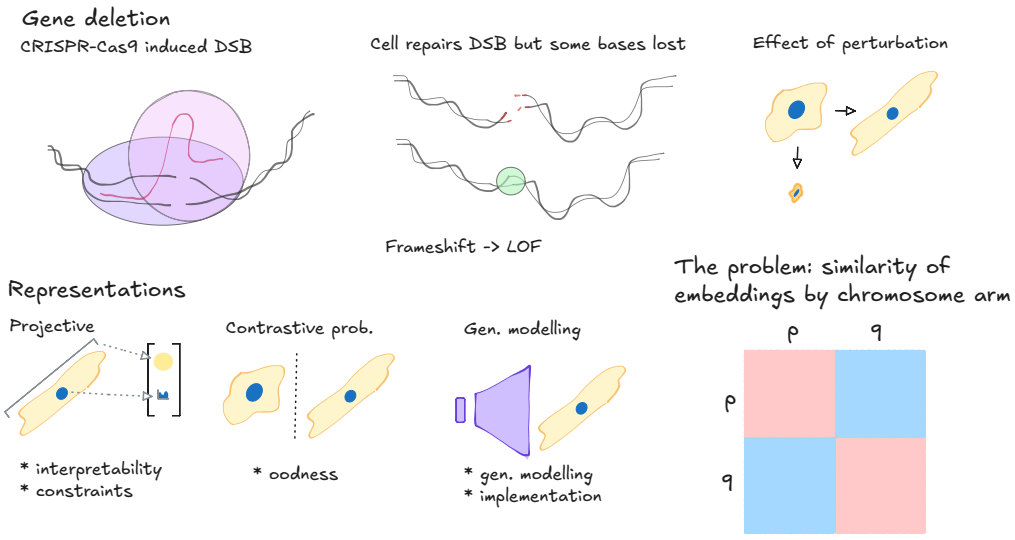


Figure A.9: Graphical abstract of the proximity bias case-study, showing (top left) a CRISPR-Cas9 complex inducing a double-strand break at a site that a guide RNA strand binds to, (top centre) the cell repairing this break, but not before some bases are lost, (top right) the cell's phenotype changing as a result as determined by images obtained using a microscope. The bottom row illustrates the computational pipeline, (bottom left) with representations first being generated, (bottom right) and a similarity matrix of gene-perturbation embeddings produced using those representations, ordered by gene position on a chromosome showing unexpected intra-chromosomal similarity.

A.11 Proximity bias reduction

In this section, we recount and expand on the results of our work in single-cell phenomics; Ravuri et al. (2025). The results below follow the setup and methodology, based on **publicly available resources** (JUMP-CPG0016, Chandrasekaran et al. (2023)), with independently written code based on the open source release of Lazar et al. (2024). No proprietary data, models or code were used for our presentation of ideas below.

In this section, we show how interpretable data analysis can be done, with a specific model (a Gaussian mixture model) in mind, that guides data exploration in the presence of extreme data noise. We also show how latent variables can be obtained through estimation of parameters guided through domain understanding, hence constraining the models such that useful latent variables can be recovered.

The section is organised as follows. First, we provide a biological background to explain the context of gene deletions, and the computational pipeline with which the data is analysed.

[5]Note that the Student-t is an infinite scale mixture of Gaussians.

We then show that this pipeline leads to an unexpected observation: correlations across representations of gene deletions within chromosome arms. We then present our work in **Ravuri et al. (2025)**, where we developed an algorithm to identify cells by minimising intra-arm correlation, and then argue that the method must be performing inference in a Gaussian mixture model. With this insight, we then discuss how such probabilistic models could be developed to recover interpretable cells in future work. Specifically, we show that a GMM with some parameters explicitly estimated using arm-averaged phenotypes and gene-averaged phenotypes reduces some amount of the unexpected correlations, and that classifier confidence can be used to measure a form of non-control-likeness that can be used for identifying visually striking perturbations.

A.11.1 Biological background

We first review the biological background of CRISPR-Cas9 induced gene deletions, and the phenomics pipelines that we are interested in.

Within the context of drug-development, understanding the behaviour of single-gene inactivations in cell lines of interest can be useful. For example, drug screens can be run to identify drugs that have a similar effect to certain known gene inactivations that have beneficial downstream effects. Additionally, single gene deletions can be helpful in discovery of biological pathways, as if two genes are on the same pathway, their single deletions can result in the same phenotype.

Experiments for such single gene deletions are done as follows. A high-throughput experimental setup is constructed, that has plates consisting of many wells in which cells of a specific kind (corresponding to certain cell-lines) are cultured. Each well holds cells which will be affected by a gene deletion, with some wells being reserved to house control cells.

Within each well, clones of a specific cell are introduced and are allowed to grow. Then, through viruses and lipofection, CRISPR-Cas9 with certain guide RNAs are introduced into cells. These complexes find a matching pattern (to the introduced guide RNA, matching portions of the gene being deleted) within DNA and induce a cut (double-strand break, DSB). The cell typically repairs such DSBs, but not before some bases at the cut-site are lost due to the activity of DNA nucleases. The loss of even a single base on the DNA can lead to the inactivation of a gene if the cut site is on an exon (a snippet used for making mRNA and subsequently protein),

as a sequence of three bases corresponds to an amino acid, and a deletion of one base can lead to an entirely different amino acid being interpreted from a specific triplet of bases. Control wells typically correspond to cells infected by viruses that are non-targeting (or, in some protocols, harmlessly target introns). For each gene, and controls, multiple guides are used to ensure the deletion of a gene, and to account for variability in the effect of a gene deletion depending on where in the sequence the deletion happens (for example, it may be that deletions at the active site of some proteins would be more catastrophic to function as opposed to changes on to the surface or tail).

The typical measurement pipeline that follows for the analysis of these experiments is one where images of these cells are taken a considerable amount of time (typically days) after the infection of the viruses, to allow for time for DSBs to occur, and for their effects to take place. Some proteins have half-lives of just minutes and therefore, waiting for days allows for transcription to account for most of the protein expression in a cell. These images are fed through a pipeline that extracts representation (e.g. through human-engineered feature functions or VAEs) of gene deletions, but also accounts for variability induced by experimental artifacts (technical effects). These representations are typically centred with respect to control cells, leading to an estimate of the effect of performing an intervention corresponding to a gene deletion. Similarity matrices between gene deletion representations are constructed, which are the main objects with which gene-gene and gene-drug relationships can be studied.

These similarity matrices show something unexpected, which we present next: "proximity-bias".

A.11.2 The proximity-bias problem

In the previous subsection, we described how gene deletion experiments produce gene-gene similarity matrices of representations of cell images when a certain gene deletion occurs.

When the similarity matrices corresponding to representations of gene deletions are ordered by chromosome arm, we see similarities of gene embeddings within chromosome arms. This is unexpected because (due to evolutionary reasons) gene location does not generally correlate with gene function. Lazar et al. (2024) hypothesise that the reason for the unexpected correlations is due to the fact that sometimes, during cell-repair of DSBs (through the non-homologous end joining mechanism, NHEJ), megabase-scale segments of the DNA can be lost, sometimes as

much as the entire chromosome arm (Cullot et al., 2019). Such truncations are expected to occur in few cells (less than 5%, depending on source). Such experimentally induced correlations are harmful to study true gene-gene interactions, so, in **Ravuri et al. (2025)**, we aimed to find single-cells that likely are affected by such off-target effects of using CRISPR-Cas9 for gene deletions. We term the hypothesised cells affected by these truncations "proximity-bias (inducing) cells" (PB cells).

In some cases, such as deletions on chromosome 17p, on which the tumour suppressor TP53 is located, deletions could induce a loss of P53 function, which is famously related to increased tumour rates, and thus, identifying PB cells is related to finding cancer-like cells in images. This pipeline is visualised in fig. A.9, and a real-life similarity matrix is visualised in fig. A.10.

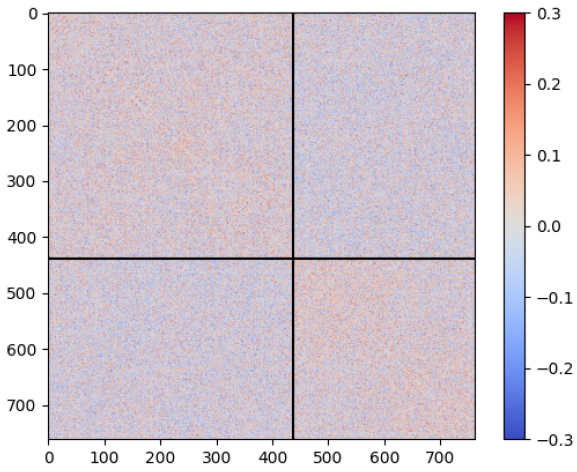


Figure A.10: A gene-gene similarity matrix corresponding to chromosome 1, based on the CPG0016 dataset. Each element in the matrix shows the correlation of embeddings corresponding to that gene's deletion, obtained from the imaging pipeline. We see, unexpectedly, intra-arm correlation within chromosome arms, hypothesised to arise due to cells where the off-target action of CRISPR-Cas9 leads to chromosomal truncations, making some cells within chromosome arms appear similar even when they correspond to unrelated gene deletions.

For our work, we use the JUMP cell-painting dataset CPG0016 of Chandrasekaran et al. (2023). They provide "CellProfiler" features (Stirling et al., 2021) available at the single-cell level, which measure statistics of cell images, for example, areas and intensities for each channel (nuclear, mitochondrial, etc.).

Assuming that, as discussed, problematic single-cells are responsible for this artefact, one can turn to unsupervised clustering methods to try to identify embeddings of a small population distinct to those of the other cells. However, in practice, cell-image embeddings carry an extreme level of variability, due to both biological and technical reasons (e.g. natural cell contortion

and imaging artefacts). UMAPs performed on the single-cell embeddings show no apparent clusters. We posit that this is because such visualisations encode little information typically (encoding just a nearest-neighbour graph, in a lossy way, the information retention within which is possibly low), and a correlation matrix may encode much more information. In a sense, this provides a raison d'être for methods that use the full covariance matrix instead of simply the nearest-neighbour graph.

A simple correction proposed by Lazar et al. (2024) involves subtracting an arm-specific correction such that the resulting similarities are decorrelated (with the corrections computed using gene embeddings corresponding to genes that are not expected to have a strong phenotype), but in this thesis, we detail how the underlying process can be studied.

In this subsection, we have presented the proximity bias problem as unexpected gene-gene similarities within arms, hypothetically caused due to chromosome-arm truncations that occur in some cells due to the off-target effect of CRISPR-Cas9. Next, we present a (non-probabilistic) algorithm to classify such cells.

A.11.3 A cell identification algorithm

In the previous subsection, we mention that single-cells suffering from off-target effects may cause unexpected gene-gene correlations. In this subsection, based on our work in **Ravuri et al. (2025)**, we introduce one possible method to identify the problematic PB cells. We directly try to identify single-cell representations, that when removed, reduces a loss function defined as the average intra-arm correlation. Our algorithm is shown in algorithm 1.

We use a weakly supervised approach to filter proximity bias ("PB") affected cells—i.e. cells corresponding to the within-arm correlations, *before* aggregating single-cell embeddings. A classifier is trained to label cells as "on-target" or PB-affected (a latent variable) using the measurable increase in *intra*-arm correlation as weak supervision. During training, the classifier minimises intra-arm similarities in aggregated gene embeddings by excluding cells.

Formally, we work with a matrix of embeddings $\hat{\mathbf{X}} \in \mathbb{R}^{n \times d}$, which contains batch-corrected d-dimensional embeddings of n single-cells, each of which is affected by a gene perturbation g. The batch correction step consists of centring and scaling, meaning that the feature-wise means are exactly zero across all cells (within plates). Our algorithm, detailed in Algorithm 1, uses a classifier $f_\theta : \mathbb{R}^d \to (-1, 1)$, parameterised by θ, constrained such that, in every well, a

Algorithm 1 Identifying single-cell embeddings to minimise average intra-arm correlation

Input:
Raw single-cell embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$
Batch-corrected embeddings $\hat{\mathbf{X}} \in \mathbb{R}^{n \times d}$.
Gene set $\mathcal{G}$ for chromosome N and arm a.
Classifier $f : \mathbb{R}^d \to \{-1, 1\}$, with params θ.
Initialise: Set initial weights θ_0 and $\mathbf{w} = 1$.
while not converged **do**
    Build the classification $\mathbf{w}$ by setting q out of every m cells with high PB score $f(\mathbf{X})$ to zero (approximately) by using softmax operations.
    Remove proportion of cells under percentile p with lowest scores: $\hat{\mathbf{X}} = \hat{\mathbf{X}} \odot \mathbf{w}$.
    Group $\hat{\mathbf{X}}$ by genes in $\mathcal{G}$: $\forall g \in \mathcal{G} : \bar{\mathbf{X}}_g = \frac{1}{|\{i:g_i=g\}|} \sum_{\{i:g_i=g\}} \hat{\mathbf{X}}_i$.
    Calculate correlation matrix: $\mathcal{P}(\mathbf{w}) = \text{cor}\left(\{\bar{\mathbf{X}}_g : g \in \mathcal{G}\}\right)$.
    Set objective as mean intra-arm correlation: $\mathcal{L} = \text{mean}(\mathcal{P}(\mathbf{w}))$.
    Minimise; update θ by gradient descent: $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta)$.
Output: Optimised classifier f and binary vector $\mathbf{w}$, indicating retained cell embeddings.

fraction of p cell embeddings are chosen to be removed (i.e., set to zero). After filtering, we average embeddings by gene, and compute a gene-by-gene correlation matrix $\mathcal{P}$ using these embeddings, ordered by position on chromosome.

Training f_θ is difficult because we cannot backpropagate through a hypothetical discrete filtering step in order to minimise the average correlation of the gene-by-gene correlation matrix $\mathcal{P}$ with respect to θ. In practice, we avoid this using a soft filtering during training. For every well we sample m cells from which we aim to remove $q = \lfloor m \times p \rfloor$ cells, and average over the rest to compute our filtered embeddings. To represent the cells, we construct a set of m single-cell embeddings $\mathbf{X}$, which a neural network maps to latent scores $\mathbf{Z} = f(\mathbf{X}) \in (-1, 1)^{m \times q}$. We then calculate a soft-selection vector $\mathbf{w}$ as, $\mathbf{w} = 1 - \min\left[\sum_j \sigma\left(\mathcal{T}\mathbf{Z}_{\cdot j}\right), 1\right]$, where σ corresponds to the softmax function, applied column-wise (i.e. across cells) and $\mathcal{T}$ is an inverse-temperature hyperparameter. This construction leads to a near-discrete vector when the inverse-temperature $\mathcal{T}$ is high, and at most q elements being approximately 0, as desired.

We chose this construction over sampling using the Gumbel-softmax trick (the concrete distribution, Jang et al. (2017); Maddison et al. (2017)) because we found that it led to lower variance gradients, which made the objective easier to optimise. Additionally, we found that averaging over relatively few cells (in practice we set m to be 100 cells) helps us get lower variance estimates of $\mathcal{P}$ because we could load larger batches of genes into memory, leading to more stable optimisation.

Figure A.11 shows that, using a single chromosome as a proxy for the rest of the genome

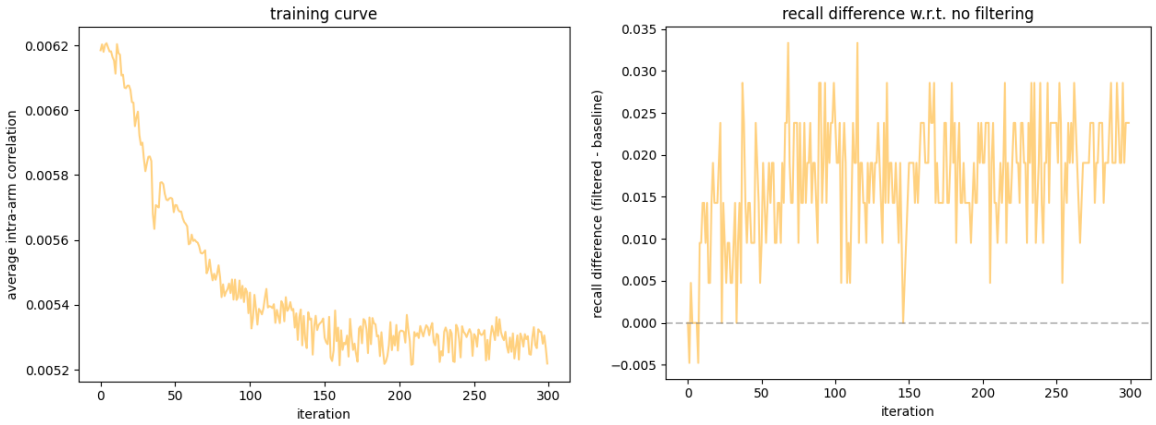


Figure A.11: Graphs showing that as proximity bias is reduced, recall of known biological relationships increases. These results are based on open-source resources (CPG0016, CORUM and HuMAP).

(as position does not generally relate to function), removing 1% of cells within the CPG0016 embeddings leads to both a reduction in the average intra-arm correlation **and** an increase in recall of known biological relationships. To measure the latter, we used the datasets suggested by Lazar et al. (2024), specifically the CORUM and HuMAP datasets of gene-gene relationships (Giurgiu et al., 2019; Drew et al., 2021)[6]. **Ravuri et al. (2025)** also found that the characteristics of cells removed is at least specific to chromosome arms (as a model trained on an arm cannot be used to reduce PB on another).

⋆ Unlike in the case of **Ravuri et al. (2025)**, where embeddings based on an autoencoder were used, for CPG0016 embeddings, our results are achieved only if the embeddings are first subjected to a Gaussian random projection. It may be the case that this process makes the embeddings more Gaussian-like, reducing implicit model misspecification within the algorithm. Recall of known relationships likewise is only increased when the embeddings are subjected to a Gaussian random projection.

Our objective function can be inspired by both a biological standpoint (as the inter-arm correlations are unexpected unless attributed to unintended effects of using CRISPR) but also a probabilistic perspective. In a Gaussian mixture model (where the mixtures correspond to cells affected by proximity bias and those that are not), with centred component means, the average correlation between component means naturally arises as an objective to be minimised.

To recap, we have shown an algorithm for filtering single-cells, removal of which seems

[6]We used the publicly available `humap2_complexes_20200809.txt`, where we used all complexes (with all confidence levels one through five) to form gene-gene pairs, and the publicly available `corum_allComplexes.txt`, where we used the "human" complexes to form our pairs. This lead to a similar number of known relationships as Lazar et al. (2024). We did not use the reactome dataset of Gillespie et al. (2022) as suggested by Lazar et al. (2024) as using `c2.cp.reactome.v2024.1.Hs.json` did not lead to any more unique gene-pairs.

to reduce proximity-bias. In the coming subsection, we argue that this inference process is approximately inference within a GMM.

A.11.4 A probabilistic interpretation of correlation minimisation

This subsection shows that assuming a Gaussian mixture model with a centred Gaussian prior over the component means, we recover an objective that penalises the average covariance between component means, which was used as a biologically-inspired heuristic in the last subsection.

The Model Consider a setting with two genes being perturbed, a and b, located on the same chromosome arm. We assume that genetic perturbations to single-cells have a mean impact on a gene's representation with respect to control cells, measured by $\boldsymbol{\mu}^a, \boldsymbol{\mu}^b \in \mathbb{R}^d$. With some probability, a cell is affected by a mechanism thought to be chromosomal truncation (i.e. if a cell is "proximally biased"), the effect of which is denoted by $\boldsymbol{\mu}^p$. We assume a multivariate Gaussian distribution on $\mathbf{M}$, and we assume that embeddings are centered,

$$\mathbf{M} = \begin{bmatrix} \boldsymbol{\mu}^{aT} \\ \boldsymbol{\mu}^{bT} \\ \boldsymbol{\mu}^{pT} \end{bmatrix} \sim \mathcal{MN}\left(\mathbf{0}, (1+\epsilon)\mathbf{I} - \frac{1}{n_g}\mathbf{O}, \mathbf{I}_d\right).$$

The choice of covariance is just a centring matrix $\mathbf{H}$ with jitter ϵ added along the diagonal. The covariance naturally arises if we zero-centre the embeddings by construction; assume a non-zeroed normal matrix $\mathbf{M}'$, centring $\mathbf{M}'$ as $\mathbf{M} = \mathbf{H}\mathbf{M}'$ leads to the row-covariance $\mathbf{H}\mathbf{I}_{n_g}\mathbf{H} = \mathbf{H}$ (as $\mathbf{H}$ is idempotent).

If the i-th cell of perturbation k is proximally biased, we represent it with $\mathbf{w}_i^k = 1$, and zero otherwise. The prior distribution on the vector $\mathbf{w} = [\mathbf{w}^a \quad \mathbf{w}^b]^T$ is such that, for every well (which contains n' cells), exactly $\lfloor p_w n' \rfloor$ cells are sampled uniformly. This ensures that the marginal probability $\mathbb{P}(\mathbf{w}_i^k = 1)$ is p_w. The observed control-centred representations of single-cells from the phenomics experiments are represented by $\mathbf{X}^a, \mathbf{X}^b \in \mathbb{R}^{n \times d}$. Assume that,

$$\mathbf{X}^a \mid \mathbf{w}^a, \boldsymbol{\mu}^a, \boldsymbol{\mu}^p \sim \mathcal{MN}\left((\mathbf{1} - \mathbf{w}^a) \otimes \boldsymbol{\mu}^{aT} + \mathbf{w}^a \otimes \boldsymbol{\mu}^{pT}, \sigma_m^2 \mathbf{I}_n, \mathbf{I}_d\right),$$
$$\mathbf{X}^b \mid \mathbf{w}^b, \boldsymbol{\mu}^b, \boldsymbol{\mu}^p \sim \mathcal{MN}\left((\mathbf{1} - \mathbf{w}^b) \otimes \boldsymbol{\mu}^{bT} + \mathbf{w}^b \otimes \boldsymbol{\mu}^{pT}, \sigma_m^2 \mathbf{I}_n, \mathbf{I}_d\right).$$

Inference: The Objective The posterior is as follows,

$$p\big(\boldsymbol{\mu}^a, \boldsymbol{\mu}^b, \boldsymbol{\mu}^p, \mathbf{w}\big|\mathbf{X}^a, \mathbf{X}^b\big) \propto p\big(\mathbf{X}\big|\mathbf{M}, \mathbf{w}\big)p\big(\mathbf{M}\big)p(\mathbf{w}).$$

We use a variational approximation,

$$q(\mathbf{w}|\mathbf{X}) = \delta(g_\theta(\mathbf{X})) := \delta(\tilde{\mathbf{w}}),$$

where g_θ is a neural network parameterised such that its support matches that of the prior (i.e. a fixed number of cells per well are identified as proximally biased). The implied evidence lower bound (ELBO) is,

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{q(\mathbf{w},\mathbf{M}|\mathbf{X})}\left(\log p\big(\mathbf{X}\big|\mathbf{M}, \mathbf{w}\big)p(\mathbf{M})\right) - \mathrm{KL}(q(\mathbf{w})\|p(\mathbf{w})) \\ &= \log p(\mathbf{X}|\mathbf{M}, \tilde{\mathbf{w}}) + \log p(\mathbf{M}) + c.\end{aligned}$$

We use block coordinate ascent for inference (i.e. CAVI/EM).

Inference: Step 1 First, we maximise the objective with respect to $\mathbf{M}$, resulting approximately in the intuitive maximum likelihood estimator. W.l.o.g., consider the partial derivative of $\mathcal{L}(\theta)$ with respect to a specific component $\mu_k^a \in \mathbb{R}$, and let $\tilde{\mathbf{X}^a} = \mathbf{X}^a - (\mathbf{1} - \mathbf{w}^a) \otimes \boldsymbol{\mu}^{aT} - \mathbf{w}^a \otimes \boldsymbol{\mu}^{pT}$. First, note that,

$$p\big(\mathbf{X}\big|\mathbf{M}, \mathbf{w}\big) = p\big(\mathbf{X}^a\big|\boldsymbol{\mu}^a, \boldsymbol{\mu}^p, \mathbf{w}^a\big) \cdot p\big(\mathbf{X}^b\big|\boldsymbol{\mu}^b, \boldsymbol{\mu}^p, \mathbf{w}^b\big),$$

and that,

$$\begin{aligned}\log p(\mathbf{M}) &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}^T\left((1+\epsilon)\mathbf{I} - \frac{1}{n_g}\mathbf{O}\right)^{-1}\right) + c \\ &= -\frac{1}{2}\mathrm{tr}\left(\mathbf{M}\mathbf{M}^T\left(\frac{1}{1+\epsilon}\mathbf{I} + \frac{1}{\epsilon n_g(1+\epsilon)}\mathbf{O}\right)\right) + c \qquad \text{Sherman-Morrison} \\ &\approx -\frac{1}{2}\sum_{k_a}\|\boldsymbol{\mu}^{k_a}\|^2 - \frac{1}{2n_g\epsilon}\sum_{k_a,k_b}\boldsymbol{\mu}^{k_a{}^T}\boldsymbol{\mu}^{k_b}.\end{aligned}$$

The derivative of the ELBO with respect to a component of the mean is,

$$\begin{aligned}
\frac{\partial}{\partial \mu_k^a}\mathcal{L}(\theta) &= \frac{\partial}{\partial \mu_k^a}\left[\log p(\mathbf{M}) + \log p\left(\mathbf{X}^a \middle| \boldsymbol{\mu}^a, \boldsymbol{\mu}^p, \mathbf{w}^a\right)\right] \\
&= \frac{\partial}{\partial \mu_k^a}\left[\log p(\mathbf{M}) - \frac{1}{2\sigma_m^2}\mathrm{tr}\left[\tilde{\mathbf{X}}^a\tilde{\mathbf{X}}^{aT}\right] - \frac{nd}{2}\log(\sigma_m^2)\right], \\
&\propto \frac{\partial}{\partial \mu_k^a}\left(\left[\frac{1}{\sigma_m^2}\sum_{i=1}^{n}\left\|\mathbf{X}_i^a - \left[(1-w_i^a)\boldsymbol{\mu}^a + w_i^a\boldsymbol{\mu}^p\right]\right\|^2\right] + \mu_k^{a2} + \frac{2\sum_m \mu_k^a\mu_k^m}{n_g\epsilon}\right).
\end{aligned}$$

Setting this derivative to 0 for the maximum-likelihood solution leads to,

$$\begin{aligned}
&2\mu_k^a + \frac{2(2\mu_k^a + \mu_k^b + \mu_k^p)}{n_g\epsilon} - \frac{1}{\sigma_m^2}\sum_{i=1}^{n} 2\left[X_{ik}^a - (1-w_i^a)\mu_k^a - w_i^a\mu_k^p\right]\left[1-w_i^a\right] = 0, \\
\Rightarrow &\mu_k^a + \frac{\mu_k^a}{n_g\epsilon} - \frac{1}{\sigma_m^2}\sum_{i=1}^{n}\left[X_{ik}^a(1-w_i^a) - (1-w_i^a)\mu_k^a\right] = 0, \\
\Rightarrow &\mu_k^a + \frac{\mu_k^a}{n_g\epsilon} + \frac{n(1-p_w)\mu_k^a}{\sigma_m^2} - \frac{1}{\sigma_m^2}\sum_{i=1}^{n}\left[X_{ik}^a(1-w_i^a)\right] = 0, \\
\Rightarrow &\hat{\mu}_k^a = \frac{\sum_{i=1}^{n}\left[X_{ik}^a(1-w_i^a)\right]}{\left(\sigma_m^2 + \frac{\sigma_m^2}{n_g\epsilon} + n(1-p_w)\right)} \overset{n\to\infty}{\approx} \frac{\sum_{i=1}^{n}\left[X_{ik}^a(1-w_i^a)\right]}{n(1-p_w)} \equiv \bar{\mathbf{X}}_{:k}^{a,\not\phi}.
\end{aligned}$$

We see that the solution for the mean embedding of a genetic perturbation is simply the average embedding corresponding to non-proximally biased cells known to have that perturbation induced.

Inference: Step 2 In this step, we optimise the ELBO with respect to θ, i.e. the parameters associated with the latent variables $\mathbf{w}$. In particular, we will argue that this step leads to minimisation of the average covariance between the average gene embeddings.

We conjecture that maximisation of $\log p(\hat{\mathbf{M}})$ is an important part to the optimisation of the objective $\mathcal{L}$. A sketch is as follows. The first term of the ELBO without constants (with

respect to θ) is,

$$\begin{aligned}
\mathcal{T}_1^a &\equiv -\frac{1}{2\sigma_m^2}\sum_{i=1}^{n}\left\|\mathbf{X}_i^a - (1-w_i^a)\boldsymbol{\mu}^a - w_i^a\boldsymbol{\mu}^p\right\|^2 \\
&= -\frac{1}{2\sigma_m^2}\sum_{i=1}^{n}\sum_{k=1}^{d}\left(X_{ik}^a - (1-w_i^a)\mu_k^a - w_i^a\mu_k^p\right)^2 \\
&= -\frac{1}{2\sigma_m^2}\sum_{i=1}^{n}\sum_{k=1}^{d}(1-w_i^a)\mu_k^{a2} + w_i^a\mu_k^{p2} - 2X_{ik}^a(1-w_i^a)\mu_k^a - 2X_{ik}^a w_i^a\mu_k^p + c \\
&= \frac{1}{2\sigma_m^2}\sum_{k=1}^{d} n(1-p_w)(\bar{\mathbf{X}}_{:k}^{a,\not p})^2 + np_w\bar{\mathbf{X}}_{:k}^{a,p}\frac{\bar{\mathbf{X}}_{:k}^{a,p} + \bar{\mathbf{X}}_{:k}^{b,p}}{2} + c,
\end{aligned}$$

where the last step follows from substituting the means μ with their estimates, and the fact that the mean of the perturbations will use both $\bar{\mathbf{X}}_{:k}^{a,p}$ and $\bar{\mathbf{X}}_{:k}^{b,p}$. Optimisation of this term with respect to θ (and therefore $\mathbf{w}$) should just lead to a balancing dynamic, as the relabelling of a cell would inversely affect $\bar{\mathbf{X}}_{:k}^{a,\not p}$ and $\bar{\mathbf{X}}_{:k}^{a,p}$.

Therefore, the optimisation of the ELBO with respect to θ should force $\log p(\hat{\mathbf{M}})$ upwards, which from the previous inference step is known to be inversely proportional to the average covariance between rows of $\hat{\mathbf{M}}$,

$$\log p(\hat{\mathbf{M}}) = -\frac{1}{2n_g\epsilon}\sum_{k_a,k_b}\boldsymbol{\mu}^{k_a T}\boldsymbol{\mu}^{k_b}(1+\epsilon n_g\delta_{k_a k_b}) \qquad \square$$

A sense-check, illustrated in figure A.12 verifies that average covariances are reduced by such an optimisation. We hypothesise that for our data, this proxy objective may have been a better objective than the likelihood, as our data distributions are verifiably non-normal, if there is no pathological/unanticipated behaviour being induced by the optimisation of the loss.

To conclude, in this subsection, we argued that a GMM underpins our objective that picks cells to minimise intra-arm correlations. In the next subsection, we show that there are features within chromosome arms that have a non-control-like distribution, and we posit that these may be the features of PB cells.

A.11.5 Characteristics of cells potentially causing PB

In the previous section, we showed that a GMM may be underpinning our method that

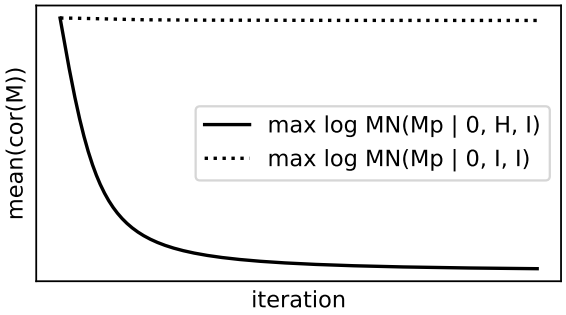


Figure A.12: A sense check that confirms that the maximization of log $\mathcal{MN}(\mathbf{M}'|\mathbf{0},\mathbf{H},\mathbf{I})$ w.r.t. θ leads to a minimization of average off-diagonal covariance between rows of $\mathbf{M}'$, as opposed to log $\mathcal{MN}(\mathbf{M}'|\mathbf{0},\mathbf{I},\mathbf{I})$, where the "data" $\mathbf{M}'$ is generated as $\mathbf{M}' = \mathbf{M} + p_w \tanh(\theta)$, $p_w = 0.1$, and $\mathbf{M}$ is a Gaussian random matrix such that $\text{cor}(\mathbf{M}_i, \mathbf{M}_j) = 0.2 + 0.8\delta_{ij}$.

classifies cells causing proximity-bias, by minimising intra-arm correlation. In this section, we propose a simple way to characterise them (although this has not been validated): by analysing features within arms that differ most to controls and other chromosome arms.

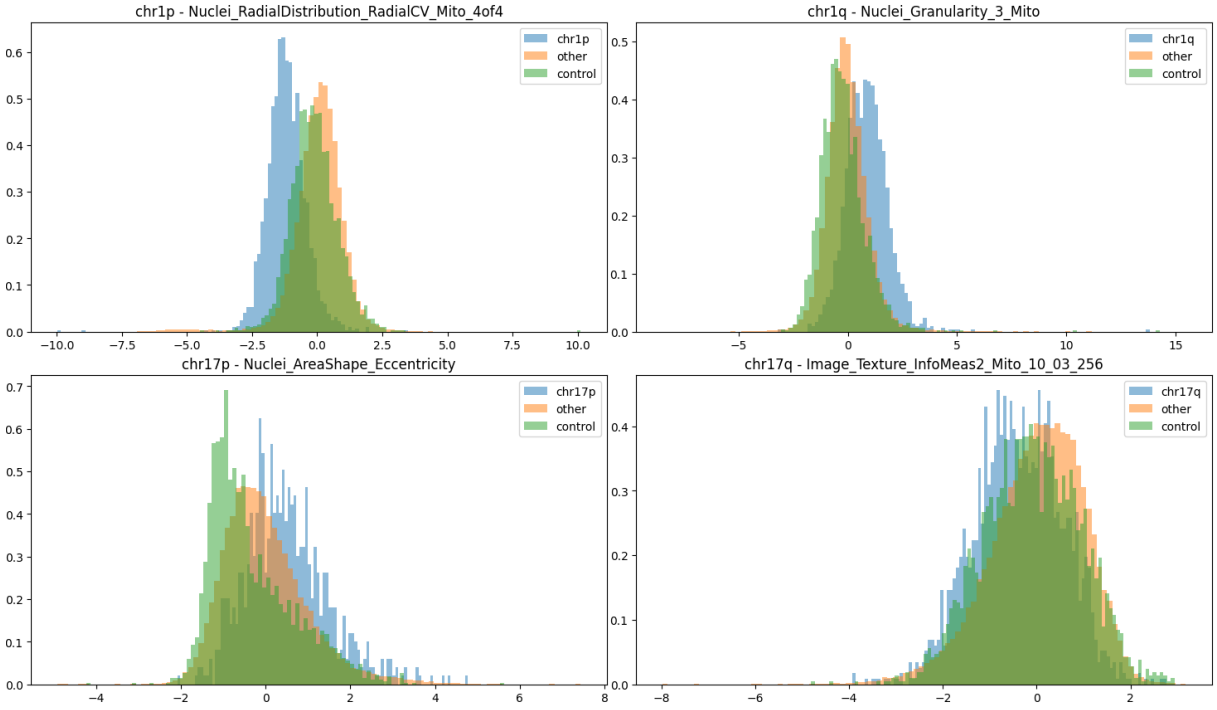


Figure A.13: Histograms of highly differing features by chromosome arm.

Figure A.13 shows features that differ most by a chromosome arm with respect to the rest of the genome, standardised by the subtraction of the feature median across all genes in the genome, and scaled by the median absolute deviation (after highly zero inflated features were dropped). This method may lead to the identification of characteristics of single-cells that suffer

from PB-effects as the features used here are somewhat interpretable, with the caveat that many features (co)vary together in reality, which can make joint interpretation of features difficult. Another caveat is that the features plotted are averaged across genes, and do not correspond to single-cell features. With single-cells, we therefore have a much higher noise-level.

Nevertheless, the figure highlights that arm average embeddings may identify characteristics of PB cells. One could therefore construct a GMM with arm-average embeddings to characterise PB cells, while using gene-specific embeddings to characterise non-PB cells, as we discuss in the next subsection.

A.11.6 Using an explicit GMM for inference

We find that taking one step of traditional expectation maximisation with a hard selection of the latents (respecting that only one cell out of a hundred is selected as causing PB) results in a reduction of proximity bias, when the parameters of the GMM are estimated directly using scientific heuristics. Specifically, we use a multiple (accounting for the low probability of PB-affected cells) of the average chromosome arm embedding as the expected PB embedding, and the average of embeddings corresponding to a gene perturbation as the expected non-PB embedding (as we expect few cells to be affected by PB, one would expect that this estimate would not be highly affected). This does not lead to quite as dramatic of an improvement as the algorithm derived filtering, but does lead to a meaningful difference to the amount of PB reduced, while using an interpretable method. Therefore, we propose that a refinement of the method may lead to identification of PB cells, albeit with a high false-positive rate induced by data noise.

A.11.7 Finding visually striking perturbations

To conclude the section, we present an observation; contrastive models can be used for finding out-of-domain examples. Using a linear classifier that has been trained to distinguish between control images and those corresponding to a single perturbation, we can define the control-likeness of a perturbation using the accuracy of the classifier as a proxy. The performance of the model is not as important here as its calibration, which is the reason behind the usage of a linear model. This leads to identification of gene deletions that are visually

striking, as shown in fig. A.14. We hypothesise that many genes found in this manner are highly conserved genes, as their deletion often seems drastic to many cells in a well.

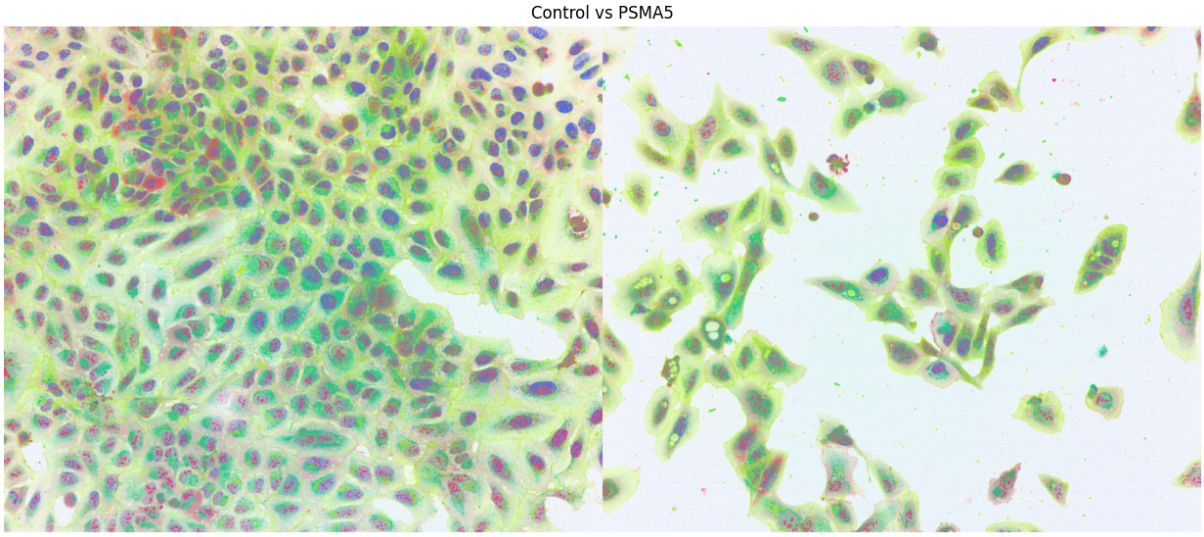


Figure A.14: Cell images, with control cells imaged on the left and perturbation on the right. This specific perturbation, clearly showing that the perturbation induces a phenotype, was found using classifier confidence, with the classifier being trained to distinguish between control cells and perturbation cells for one specific gene perturbation, using a linear model to ensure calibration.

A.12 Ice-cores

In this section, we detail an HMM-based model that allows for the inference of latent time as mentioned in section 2.4.2.[7] We show the versatility of such models, particularly to perform full-form inference with fine-grained assumptions such as monotonicity of the latent variable.

Background Ice cores preserve chemical records that reveal past climate, but extracting a reliable chronology (i.e. mapping depth to time) from these records is challenging. The traditional manual layer-counting approach based on chemical (proxy) concentration readings is both time-consuming and imprecise, failing to capture uncertainty in a principled manner.

Model Probabilistic latent variable models, namely HMMs, lend themselves to the problem naturally (Winstrup, 2011, 2016); the latent time process is discretised so that each depth measurement δ_i is associated with a time state t_{δ_i} taking values in $\{k/n_s : k \in \mathbb{N}\}$ (with n_s states per annual cycle). The transition matrix is set up such that as depth increases, the time

[7] This section is based on **Ravuri et al. (2022a)**.

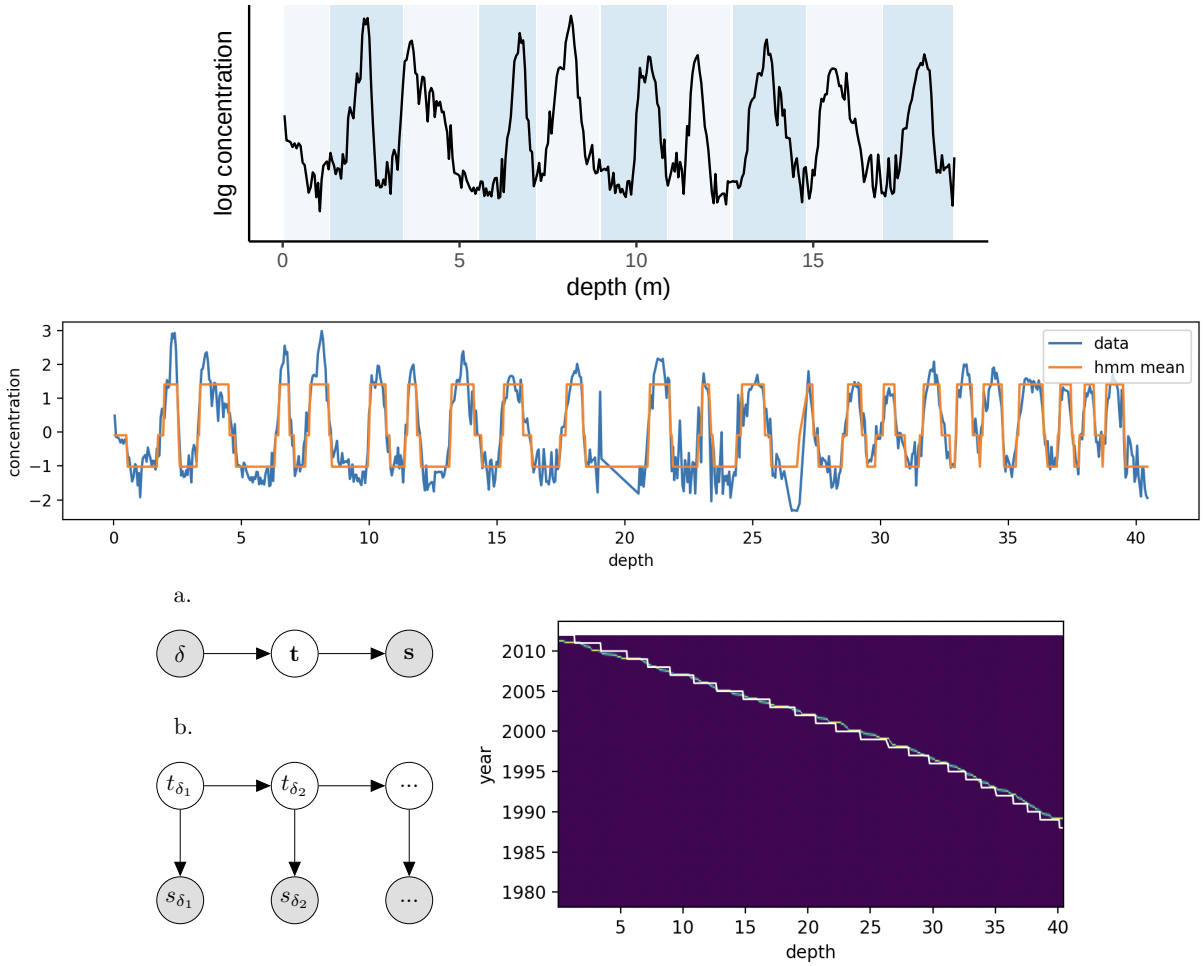


Figure A.15: Top: The ice-cores data, chemical concentrations s are a sinusoidal function of latent time, indexed by depth δ. **Middle:** Predicted mean of the observation model, using the most likely sequence of latent time, using a sinusoidal generative model as part of an HMM. **Bottom left:** an illustration of the model graph used for inference, showing known concentrations s being a function of latent time t indexed by depth d. Bottom right: Latent time recovered by the HMM plotted against expert samples.

either remains in the current state or advances, and the observation model is periodic with respect to time,

$$\mathbb{P}(t_{\delta_i} \mid t_{\delta_{i-1}}) = \begin{bmatrix} p_{1/n_s} & 1 - p_{1/n_s} & 0 & \dots \\ 0 & p_{2/n_s} & 1 - p_{2/n_s} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$
$$s_{\delta_i} | t_{\delta_i} \sim \mathcal{N}\Big(a \cos(2\pi t_{\delta_i}) + b,\ \sigma^2\Big),$$

which captures the seasonal cycle.

Discussion Figure A.15 illustrates the data and results obtained using the model above, showing agreement between expert annotated latent time and recovered time, as well as the fit of the generative model to the chemical concentration observations. The model can be seen to perform a variant of **dynamic time warping**, but due to its probabilistic formulation, automatic inference can be performed for such use-cases using PPLs, as done in **Ravuri et al. (2022a)**. Furthermore, PPLs enable simple extensions of the model with no inference implementation overhead. For example, this allows for tie-points. These can be volcanic eruptions that deposit a layer of soot at known depths. Such events can be incorporated using a model extension that sets the expected distribution of the latent time to be uniform within the year corresponding to the eruption, at the known depth that corresponds to the eruption,

$$p(s'_{\delta_{\text{tie}}} | t_{\delta_{\text{tie}}}) = \begin{cases} 1/n_s & \text{if } \lfloor t_{\delta_{\text{tie}}} \rfloor = t_{\text{tie}} \\ 0 & \text{o.w.} \end{cases}.$$

Moreover, PPLs enable inference of hierarchical variation in parameters to account for non-stationary parameters with depth (in such cases, as the MLEs lie outside the typical set, VI or MCMC is necessary as to marginalise the nuisance parameters). In the case above, as $n_s \to \infty$, the model becomes equivalent to an SDE[8] but inference in such a model class is difficult due to the fact that performing full-form inference for the latent path is not trivial, and parameterising

[8]Specifically an SDE that can be interpreted as a monotonicity-inducing transform of a Gaussian process, as GPs have representations as stochastic differential equations, described in Sarkka et al. (2013):

$$y_x \sim \mathcal{GP}\left(0, [1 + |x - x'|] \exp\left[-|x - x'|\right]\right) \Leftrightarrow \begin{bmatrix} dy_x \\ d\dot{y}_x \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3 & -\sqrt{12} \end{bmatrix} \begin{bmatrix} y_x \\ \dot{y}_x \end{bmatrix} dx + \begin{bmatrix} 0 \\ dW_x \end{bmatrix}.$$

the path using a function (the drift and diffusions of an SDE) leads to much worse performance.

Conclusion The case study is an example of how scientifically constrained latent variable models enable the inference of interpretable latents.

A.13 GPLVMs in single-cell biological data analysis

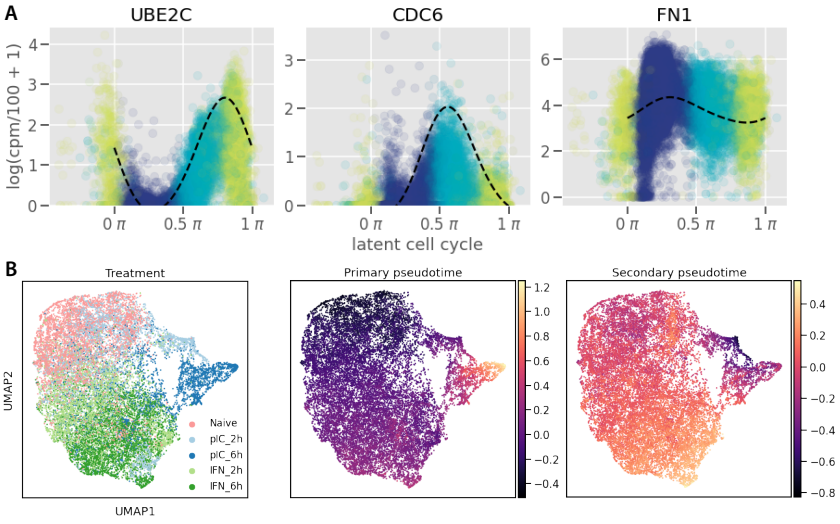


Figure A.16: Left: Reproduction of the results of Kumasaka et al. (2021) using our sparse extensions of GPLVMs. **Right:** 25k-dimensional gene expression projected into a 2d Poincaré disk, showing cell differentiation trajectories (coloured red to blue).

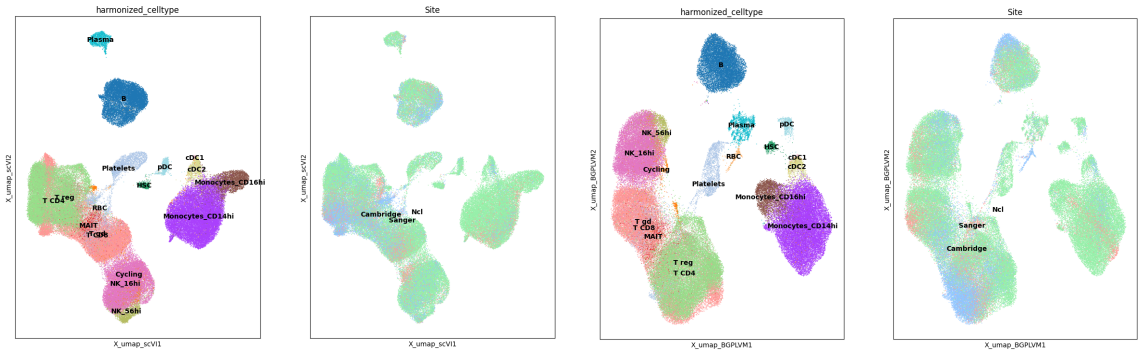


Figure A.17: Left: a UMAP of embeddings of an scRNA-seq dataset (of Stephenson et al. (2021)) generated using LinearSCVI of Svensson et al. (2020). Right: a reproduction—a UMAP of the embeddings generated by interpreting LinearSCVI as a GPLVM, showing a similar disentanglement of the embeddings by cell-type and a technical variable (site).

As mentioned in section 2.4.4, in this section, we explore how GPLVMs can be used in the context of single-cell data analysis. We briefly discuss how sparse GPLVMs are constructed,

and how they are constrained in practice for the recovery of interpretable latents. We end the section by a short description of how they can be extended to non-Euclidean settings.

Constructing sparse GPLVMs GPLVMs can be extended such that the GPs describing the data are made sparse (Hensman et al., 2013), which make inference on large datasets possible. **Lalchand et al. (2022)** and Bui and Turner (2015) show that inference in models such as,

$$\mathbf{Y}_{ij}|\mathbf{F_{ij}} \sim \mathcal{N}(\mathbf{F}_{ij}, \sigma^2), \quad \mathbf{F}|\mathbf{U} \sim \mathcal{MN}(K_{nm}K_{mm}^{-1}\mathbf{U}, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}, \mathbf{I})$$
$$\mathbf{U} \sim \mathcal{MN}(\mathbf{0}, K_{mm}(\mathbf{Z}), \mathbf{I}),$$

can be done variationally, with variational posteriors $q(\mathbf{U})$ and $q(\mathbf{X})$ that could potentially be parameterised by neural networks.[9] The ELBO (Blei et al., 2017) is simply,

$$\mathcal{L} = \mathbb{E}_{q(.)}\left(\sum_{n,d} \log p(\mathbf{Y}_{n,d}|\mathbf{F}_{n,d})\right) - \mathrm{KL}(q(\mathbf{U})||p(\mathbf{U})) - \mathrm{KL}(q(\mathbf{X})||p(\mathbf{X})).$$

The initial motivation for our work in **Lalchand et al. (2022)** was to try to fit normalising flows of Rezende and Mohamed (2016) to the posterior to capture complex forms of uncertainty, but this is difficult due to optimisation challenges. To solve such problems on a small-data scale, we found that using a large number of Monte-Carlo samples to calculate the variational bound, **under the same random seed** (a heuristic), leads to a successful optimisation. As an example, Gaussian samples can be sampled as $\mathbf{x} = \boldsymbol{\mu} + \Sigma^{1/2}\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and our method samples $\mathbf{z}$ just once and keeps them fixed throughout optimisation. That being said, we noticed that such methods often resulted in a characterisation of likelihood invariance, than multi-modality due to semantically interesting uncertainty due to data-led underspecification.

GPLVMs in single-cell data analysis In **Ravuri et al. (2022b)**, we show that frequentist GPLVMs with random effects of Kumasaka et al. (2021) can be interpreted as **Gaussian process with linear kernels describing the random-effects**, and hence extended to their sparse counterparts (using `LinearKernel` abstractions of a sparse Gaussian process implementation) to make them scalable. The embeddings obtained are visualised in fig. A.16. The variational

[9] Such variational posteriors can be specified to factorise in specific ways, for example, to respect causal ordering in temporal settings. Therefore, if one knows that a posterior decomposes in a certain way, variational posteriors can be specified to respect the factorisation, thereby narrowing the search space for the true posterior.

distributions on latents in this work are kept full-form. We found that,

1. Initialisations based on PCA, covariates and marker-gene based information, with pre-identified semantic dimensions lead to identifiable latents post inference. Specifically, a latent variable corresponding to where the cell is in, within the cell-division cycle is initialised using expression of known genes that are known to be activated during certain times within the cell cycle.

2. A periodic kernel describes the sinusoidal relationship between these latents and the expression. This imposes a strong functional constraint in the model leading to the retention of interpretable latent times, but good initialisation is crucial. Periodic covariances in such contexts have also been described, for example, by Ahmed et al. (2018).

3. We also note that "over-training" can occur, and that continued training leads to a previously interpretable latent variable losing its interpretability.

4. "Pre-training" the GPLVMs with respect to lengthscales and pseudo-inputs, with the initialisations kept frozen is needed to retain the effect of the initialisation.

5. The pseudo-inputs can be set up in such a way that their contributions to the resulting function are independent, leading to interpretable (disentangled) inducing functions.

   As an example, consider a Gaussian process using a kernel with a linear component and an RBF-like non-linear component. Partition the inducing points Z as follows,

$$Z \equiv (Z_{\not\text{in}} | Z_{\text{lin}}) = \begin{bmatrix} Z_1 & 0 \\ \pm\infty & Z_2 \end{bmatrix}.$$

   The zeros and infinities in this inducing inputs matrix are selected by considering inputs to a kernel that can set it to zero—for example, $k_{\text{rbf}}(., \pm\infty) = 0$ and $k_{\text{lin}}(., 0) = 0$. Due to this partitioning, the matrices of the sparse Gaussian process prior, $f|u \sim \mathcal{N}(K_{nm}K_{mm}^{-1} \times u, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn})$ simplify as follows,

$$K_{mm} = \begin{bmatrix} k_{\not\text{in}}(Z_1, Z_1) + k_{\text{lin}}(0,0) & k_{\not\text{in}}(Z_1, \pm\infty) + k_{\text{lin}}(0, Z_2) \\ k_{\not\text{in}}(\pm\infty, Z_1) + k_{\text{lin}}(Z_2, 0) & k_{\not\text{in}}(\pm\infty, \pm\infty) + k_{\text{lin}}(Z_2, Z_2) \end{bmatrix} = \begin{bmatrix} k_{\not\text{in}}(Z_1, Z_1) & 0 \\ 0 & k_{\text{lin}}(Z_2, Z_2) \end{bmatrix},$$

$$K_{nm} = \begin{bmatrix} k_{\not\text{in}}(X_{\not\text{in}}, Z_1) + k_{\text{lin}}(\Phi, 0) & k_{\not\text{in}}(X_{\not\text{in}}, \pm\infty) + k_{\text{lin}}(\Phi, Z_2) \end{bmatrix} = \begin{bmatrix} k_{\not\text{in}}(X_{\not\text{in}}, Z_1) & k_{\text{lin}}(\Phi, Z_2) \end{bmatrix}.$$

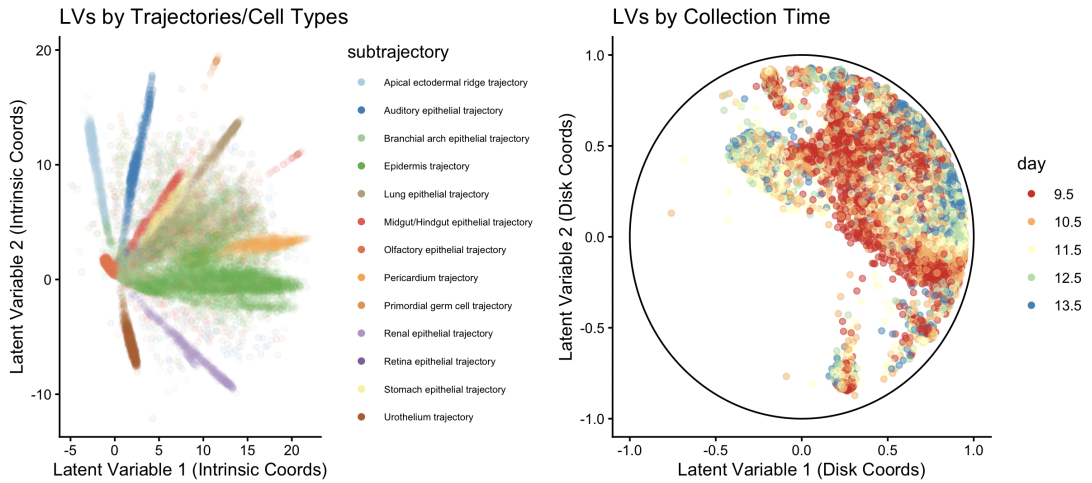


Figure A.18: UMAPs fit on mouse epithelial data (of Cao et al. (2019), processed using code from Lyu et al. (2019)) with a hyperbolic kernel, showing how undifferentiated cells are more likely to be found near the origin of the UMAP, thereby exposing the underlying tree-structure of the data.

In Zhao et al. (2024), we found that, using the observation model, $\mathbf{Y}_{i:} \sim \text{Poisson}(\sigma(\mathbf{F}_{\mathbf{i}:}) * n_l)$, (where n_l denotes the library size, fixed in our experiments), preprocessing the data so that it is row normalised, and using SCVI's (Svensson et al., 2020; Gayoso et al., 2022) encoder (that factorises across data points, parameterising $q(\mathbf{X}_{i:}|\mathbf{Y}_{i:})$) leads to embeddings similar to that of Linear SCVI (which is equivalent to a sparse GPLVM generative model with a linear kernel). These embeddings are illustrated in fig. A.17. Moreover, we find that pretraining on cell-cycle initializations used in Ravuri et al. (2022b) also recovers results of Kumasaka et al. (2021).

Non-Euclidean extensions In Mostowsky et al. (2024), we provide software for GP(LVM)s on non-Euclidean latents using kernels on manifolds and graphs (Borovitskiy et al., 2020, 2021). GPLVMs specified on hyperbolic surfaces, as an example, can recover interpretable tree-structures within the data. These embeddings are illustrated in fig. A.18. We hypothesise that such a choice of space also forms a strong constraint that enables recovery of interpretable latents, as in the case of hyperspheres.

Conclusion The examples of this section exemplify how **constraints** placed within GPLVMs, that otherwise have large degrees of freedom, are useful in practice.

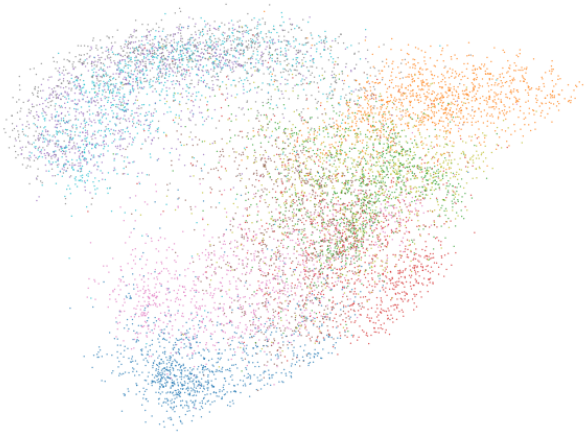


Figure A.19: An MNIST embedding obtained as an approximate solution to the CNE objective, using an eigendecomposition of an element-wise function applied to the adjacency matrix. This element-wise function results from a pairwise stationarity analysis of the CNE objective, and is inspired by the work of Levy and Goldberg (2014) in the context of word2vec/SGNS. The resulting embedding is more diffuse than Laplacian Eigenmaps, but with clusters that are not quite as well separated as in CNE, t-SNE or UMAP.

A.14 Embeddings in CNE following SGNS-style arguments

For this algorithm, following from our exposition in section 3.3.3, we use a heuristically modified adjacency matrix. The modification involves raising our adjacency matrix to the power r (selected so that all adjacencies are non-zero) that has a similar degree (achieved by multiplying the normalised adjacency with the desired degree k),

$$\tilde{\mathbf{A}} = k(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})^r.$$

Figure A.19 illustrates an embedding of the MNIST dataset obtained by the arguments above. The resulting embedding is a "smoother" version of Laplacian Eigenmaps, however, it fails to achieve the amount of separation of clusters achieved by CNE.[10]

[10]The style of our approximation is similar in form to SGNS-inspired matrix-factorisation results interpreting node2vec and similar algorithms (Qiu et al., 2018).

APPENDIX B

Additional proofs

B.1 Noise levels in dp-PCA and dp-MCA

Proof. of theorem 3.4. Assume the setup of lemma 3.3 and let λs be major eigenvalues of the sample covariance matrix. Due to lemma 3.2, theorem 3.3, and the fact that the major eigenvalues of the sample covariance are minor eigenvalues of the precision,

$$\hat{\sigma}^2 = \frac{\sum_{i=d_q+1}^{n} \lambda_i}{n - d_q} \text{ and } \hat{\beta} = \frac{n - d_q}{\sum_{i=d_q+1}^{n} \frac{1}{\lambda_i}}.$$

Therefore,

$$\frac{\hat{\sigma}^2}{\hat{\beta}} = \frac{\sum_{i=d_q+1}^{n} 1/\lambda_i \sum_{i=d_q+1}^{n} \lambda_i}{(n - d_q)^2} \overset{\text{AM-GM}}{\geq} \sqrt[n-d_q]{\prod_i \lambda_i/\lambda_i} = 1.$$

□

B.2 The derivation of our CNE objective

This section shows how the CNE bound of section 3.3.1.2 is derived.

Note that we've made some substitutions in the original formation that appears in Damrich et al. (2022), for example, we set their parameter $\tilde{q}_{ij} = 1/d_{ij}^2$, which corresponds to the UMAP

setting. Eqn. 8 of Damrich et al. (2022) reads,

$$\mathcal{L}^{\text{NEG}}(\theta) = -\mathbb{E}_{x\sim p}\log\left(\frac{q_\theta(x)}{q_\theta(x)+1}\right) - m\mathbb{E}_{x\sim\xi}\log\left(1-\frac{q_\theta(x)}{q_\theta(x)+1}\right).$$

We use the notation $n_{\text{neg}} = m$, and we consider the objective (resulting in eq. (3.8)) in terms of the negative loss $\mathcal{E} = -\mathcal{L}$. Next, Lemma 3 of Damrich et al. (2022) specifies the choice of q_θ corresponding to UMAP to be $q_\theta(x) = 1/d_{ij}^2(\mathbf{X})$. The objective simplifies to,

$$\begin{aligned}\mathcal{E}(\mathbf{X}) &= \frac{1}{\sum_{i>j}\mathbf{A}_{ij}}\sum_{i>j}\mathbf{A}_{ij}\log\left(\frac{1}{1+d_{ij}^2(\mathbf{X})}\right) + \frac{n_{\text{neg}}}{\sum_{i>j}(1-\mathbf{A}_{ij})}\sum_{i>j}(1-\mathbf{A}_{ij})\log\left(1-\frac{1}{1+d_{ij}^2(\mathbf{X})}\right)\\ &\propto \sum_{i>j}\mathbf{A}_{ij}\log\left(\frac{1}{1+d_{ij}^2(\mathbf{X})}\right) + \frac{n_{\text{neg}}\sum_{i>j}\mathbf{A}_{ij}}{\sum_{i>j}(1-\mathbf{A}_{ij})}\sum_{i>j}(1-\mathbf{A}_{ij})\log\left(1-\frac{1}{1+d_{ij}^2(\mathbf{X})}\right)\end{aligned}$$

The multiplicative constant is approximated as,

$$\tilde{\epsilon} := \frac{n_{\text{neg}}\sum_{i>j}\mathbf{A}_{ij}}{\sum_{i>j}1-\mathbf{A}_{ij}} \approx \frac{n * n_{\text{neg}} * n_{\text{neigh}}/1.5}{(n^2-n)/2} \approx \frac{4n_{\text{neg}}n_{\text{neigh}}}{3n}.$$

Therefore, the objective becomes,

$$\begin{aligned}\mathcal{E}(\mathbf{X}) &\approx \sum_{i>j}\mathbf{A}_{ij}\log\left(\frac{1}{1+d_{ij}^2(\mathbf{X})}\right) + \frac{4n_{\text{neg}}n_{\text{neigh}}}{3n}\sum_{i>j}(1-\mathbf{A}_{ij})\log\left(1-\frac{1}{1+d_{ij}^2(\mathbf{X})}\right)\\ &\propto \sum_{ij}\mathbf{A}_{ij}\log\left(\frac{1}{1+d_{ij}^2(\mathbf{X})}\right) + \frac{4n_{\text{neg}}n_{\text{neigh}}}{3n}\sum_{ij}(1-\mathbf{A}_{ij})\log\left(1-\frac{1}{1+d_{ij}^2(\mathbf{X})}\right),\end{aligned}$$

which is eq. (3.8).

B.3 On semantic consistency

In this section, as mentioned in section 3.3.3, we show that the assumptions that underpin our Wishart interpretation are semantically consistent, i.e. correspond to natural assumptions given the data structures being modelled. We also provide a possible reason for why interpreting a Wishart distribution from a multivariate Bernoulli is successful. We show these results to reiterate claim made in the introduction that the different model choices in ProbDR "translate" as expected. The first subsection explores the Wishart interpretation, while the latter considers

how our Wishart model can be translated to one directly comparable to GPLVM (i.e. we consider what happens when the distribution on the data covariance is changed from inverse-Wishart to Wishart).

B.3.1 Consistency of the Wishart interpretation

Consider the non-linear ProbDR model,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}\left((0.5\tilde{\epsilon}^{-1}\mathbf{I} + 0.5\mathbf{H}\mathbf{P}^{u}\mathbf{H} + \mathbf{X}\mathbf{X}^{T})^{-1}, n\right).$$

Assume that the posterior variance corresponds to the marginals $\mathrm{Var}(\mathbf{X}_i|\mathbf{L}) = \sigma_x^2\mathbf{I}$. This is more or less a reasonable assumption, as ignoring the kernel term results in the Laplacian Eigenmaps solution, with the MAP embedding found at the scaled eigenvectors of $\mathbf{L}/n$, which are orthonormal. Hence, the assumption made simply assumes that the top eigenvalues are of comparable scale[1].

A way to see that the relationship between a kernel term p and a term of the data denoted here as $\mathcal{D}$ is preserved between the Bernoulli and Wishart interpretations is that, within natural exponential families of the form (written in terms of the natural parameter),

$$\log p(\mathcal{D}|\mathbf{p}) = \boldsymbol{\eta}^T\mathcal{D} - A(\boldsymbol{\eta}),$$

The properties of the log-partition function imply that (Wainwright et al., 2008),

$$\mathbb{E}(\mathcal{D}) = \nabla_{\boldsymbol{\eta}}A(\boldsymbol{\eta}) \text{ and } \mathrm{Cov}(\mathcal{D}) = \nabla_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}^T A(\boldsymbol{\eta}) \succeq 0.$$

The second condition implies directly that $\mathrm{diag}(\nabla_{\boldsymbol{\eta}}\mathbb{E}(\mathcal{D})) \geq \mathbf{0}$, meaning that the mean of the data $\mathcal{D}$ is monotonic increasing in the natural parameter $\boldsymbol{\eta}$. With the Bernoulli and Poisson interpretations, the likelihood term has a positive sign, and the adjacency is described as having a mean that is increasing in the probability parameter. With the Wishart interpretations, the sign of the likelihood is negative, and hence the mean of the random variable describing the graph Laplacian is a decreasing function of the probabilities, which is semantically consistent,

[1] and are approximately the algebraic connectivity—although this can be approximated for Erdős–Rényi graphs, an approximation for kNN graphs is likely difficult to find, as these graphs have substantially different properties. For example, degrees of a kNN graph are guaranteed to be between the number of nearest neighbours or twice that number, the latter simply due to the symmetrisation of the graph adjacency.

as the off-diagonal elements of the graph Laplacian describe **negative** adjacency.

This argument can be made explicitly, and for this, we will derive an approximation of the precision matrix of a kernel matrix. The Cauchy kernel can be expressed in terms of random Fourier features described in Rahimi and Recht (2007),

$$\mathbf{K} \approx \mathbf{\Phi}\mathbf{\Phi}^{\mathbf{T}}, \mathbf{\Phi} \in \mathbb{R}^{n \times m} \text{ with } \mathbf{\Phi}_{ij} = \sqrt{\frac{2}{m}} \cos(\mathbf{X}_i^T \Omega_j + \mathbf{b}_j).$$

Let the covariance of our interpretation in eq. (3.11) be,

$$\Sigma = \mathbf{X}\mathbf{X}^T + 0.5\mathbf{K} + \beta\mathbf{I} = (\mathbf{X} \;\; \sqrt{0.5}\mathbf{\Phi})(\mathbf{X} \;\; \sqrt{0.5}\mathbf{\Phi})^T + \beta\mathbf{I} := \tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^T + \beta\mathbf{I},$$

where we drop the double centring of $\mathbf{K}$, which does not negatively impact the quality of the visualisations after optimisation, and $\beta = 1/2\tilde{\epsilon}$. Using Woodbury, we see that,

$$\begin{aligned}
\Sigma^{-1} &= 2\tilde{\epsilon}\mathbf{I} - 4\tilde{\epsilon}^2\tilde{\mathbf{\Phi}}\mathbf{M}\tilde{\mathbf{\Phi}}^T \text{ where,} \\
\mathbf{M}^{-1} &= \mathbf{I} + 2\tilde{\epsilon}\tilde{\mathbf{\Phi}}^T\tilde{\mathbf{\Phi}} \\
&= \begin{bmatrix} \mathbf{I}_q + 2\tilde{\epsilon}\mathbf{X}^T\mathbf{X} & 2\sqrt{0.5}\tilde{\epsilon}\mathbf{X}^T\mathbf{\Phi} \\ 2\sqrt{0.5}\tilde{\epsilon}\mathbf{\Phi}^T\mathbf{X} & \mathbf{I}_m + \tilde{\epsilon}\mathbf{\Phi}^T\mathbf{\Phi} \end{bmatrix}
\end{aligned}$$

The role of $\mathbf{M}^{-1}$ coarsely is one that (a) behaves as a factor $\mathbf{M}^{-1} = \mathbf{\Theta}(n\tilde{\epsilon}) = \mathbf{\Theta}(1)$ and (b) is otherwise a function of feature correlations. Using these results,

$$\mathbb{E}(\mathbf{L}_{ij}|\mathbf{X}) \sim \text{d.o.f.} * \Sigma_{ij}^{-1} \sim -n\tilde{\epsilon}^2 \sim -1/n,$$

as expected of the graph Laplacian.

Although we do not detail the computation here, $\mathbf{M}^{-1}$ can be approximated well using standard algebra, assuming that $\mathbf{X}|\mathbf{L} \sim \mathcal{MN}(0, \Sigma, \mathbf{I}_{d_q})$. This leads to an **approximation of a Gaussian process precision matrix**, (and to our knowledge, this is the first description of such an approximation).

B.3.2 From non-linear dp-MCA to non-linear dp-PCA

In this final subsection, we show that changing the distribution from non-linear dp-MCA

to dp-PCA is non-trivial, due to the fact that the observed statistic undergoes a non-linear transformation.

The Wishart interpretation of eq. (3.11) is a non-linear extension of dp-MCA. We have shown so far that simple reinterpretations of the objective can lead to various valid probabilistic interpretations (i.e., the Bernoulli, Poisson, and Wishart interpretations that describe the adjacency matrix or a linear transformation of it). The model in eq. (3.11) is written,

$$\mathbf{L}|\mathbf{X} \sim \mathcal{W}(\Sigma^{-1}, n),$$

where $\Sigma = \mathbf{X}\mathbf{X}^T + 0.5\mathbf{H}\mathbf{P}^u\mathbf{H} + 0.5\epsilon^{-1}\mathbf{I}$. The model is equivalent to (roughly),

$$\mathbf{L}^+|\mathbf{X} \sim \mathcal{W}^{-1}(\Sigma, n).$$

We justify the (pseudo-)inversion of the graph Laplacian using the simple fact that they share eigenvectors corresponding to non-zero eigenvalues. Instead of placing the initial distribution of eq. (3.11) on $\mathbf{L}$, if $\mathbf{L} + \gamma\mathbf{I}$ were considered instead, we can see that $\lim_{\gamma\to 0}(\mathbf{L} + \gamma\mathbf{I})^{-1} = \lim_{\gamma\to 0}\mathbf{U}(\mathbf{\Lambda} + \gamma)^{-1}\mathbf{U}^T = \lim_{\gamma\to 0}\mathbf{1}\mathbf{1}^T/\gamma + \mathbf{L}^+$. Double centring this matrix leads to $\mathbf{L}^+$.

We ask the question: what is the closest dp-PCA/GPLVM-like model, i.e., what is the closest Wishart (as opposed to an inverse-Wishart) distribution that models the covariance,

$$\mathbf{L}^+|\mathbf{X} \sim \mathcal{W}(\overset{?}{.}, n)?$$

While answering this question, we find that mean/mode matching does not lead to the preservation of the visual quality of embeddings. A simple reinterpretation of the objective is not possible, as the objective behind eq. (3.11) involves $-\text{tr}(\mathbf{L}\Sigma)$ whereas the likelihood of the model above would involve $-\text{tr}\left((\mathbf{L}\Sigma)^+\right)$.

We posit that the reason for the above failings is model misspecification, and the equivalence of model statements **is not resistant to non-linear changes of the sufficient statistic**. Model semantics are the solution to building a logical model.

The off-diagonal elements of the graph Laplacian are, on average, expected to scale with $1/n$, as every point has a constant number of neighbours. Thus, on average, the larger the number of data points, the smaller the probability that an off-diagonal element is non-zero. **We show that the pseudo-inverse of the graph Laplacian has a similar property.**

Consider an Erdős-Réyni graph, with probability of adjacency $p = k/n$. We approximate the distribution of eigenvalues of the graph Laplacian with a gamma distribution. $\mathbb{E}(\lambda) = \mathbb{E}(\text{tr}(\mathbf{L}))/n = k$. Moreover, eigenvalues of a graph correlate with its degrees; this is illustrated in fig. B.1. Therefore, we approximate $\text{Var}(\lambda) = \text{Var}(\text{Binomial}(n, p = k/n)) = k(1 - k/n)$.

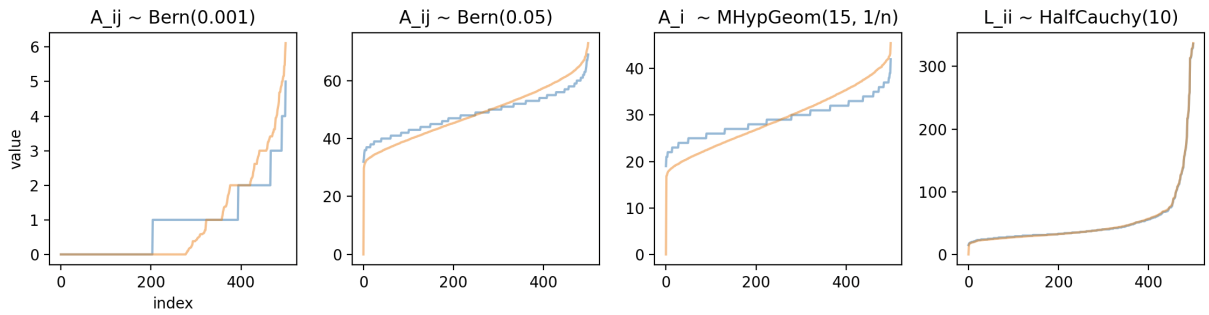


Figure B.1: Empirical evidence of graph degrees correlating with the corresponding combinatorial Laplacian's eigenvalues. The (quantile) plots show, for various random graphs, sorted eigenvalues (orange) and sorted degrees (blue) on the y-axis, with the x-axis corresponding to rank.

Matching moments, we approximate,

$$\begin{aligned} \lambda &\sim \text{Gamma}\left(\alpha = \frac{kn}{n-k}, \beta = \frac{n}{n-k}\right) \\ \implies \text{tr}(\mathbf{L}^+) &= n\mathbb{E}(\lambda^{-1}) = \frac{n}{k-1+k/n} \\ \implies \bar{\mathbf{L}^+_{ij}} &= -\frac{\text{tr}(\mathbf{L}^+)}{n^2} = \frac{1}{n(k-1)+k}, \qquad \text{centered } \mathbf{L}^+ \end{aligned}$$

where $\bar{\mathbf{L}^+_{ij}}$ is the average non-diagonal element of the pseudo-inverse of the graph Laplacian. **These values, as with the graph Laplacian, also scale with $1/n$, unlike those in Wishart matrices**. Therefore, the natural model for $\mathbf{L}^+$ is,

$$\mathbf{L}^+ \sim \mathcal{W}\left(\Sigma/n^2, n\right).$$

Future work will explore whether a transform of the data exists, such that its approximate covariance is $\mathbf{L}^+$, as a GPLVM is perhaps a better model definition due to, at least, marginal consistency reasons. Moreover, Gaussian processes on non-Euclidean manifolds have been studied in literature, for example by Borovitskiy et al. (2020), which are useful because hyperbolic kernels represent/recover tree structures in the data (Nickel and Kiela, 2017) and hyperspherical kernels can be used to represent cyclicality.

B.4 Dropping the variational constraint in ProbDR

In this section, we show that discarding the variational constraint as mentioned in section 3.4.1 and marginalising the moment in either case of our Wishart models leads to the standard Gaussian process assumed in many DR models[2].

Theorem B.1 (Marginal consistency with dp-PCA and GPLVM)**.** Assuming a generative model assumed as part of dp-PCA or GPLVM,

$$\mathbf{y}|\mathbf{S} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{\rho}\mathbf{S}\right), \quad \mathbf{S}|\mathbf{X} \sim \mathcal{W}(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}, \rho),$$

or as part of dp-MCA (which places a Wishart distribution on a precision matrix, equivalent to an inverse-Wishart on a covariance),

$$\mathbf{y}|\mathbf{S} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{S} * (\rho - n + 1)\right), \quad \mathbf{S}|\mathbf{X} \sim \mathcal{W}^{-1}\left(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I}, \rho\right)$$

the marginal distribution of any column of the data $\mathbf{y}$, as $\rho \to \infty$, is given by,

$$\mathbf{y}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}).$$

Proof of theorem B.1. The idea is simple: the generative models were chosen so that the covariance concentrates around the mean. In the first case,

$$\mathrm{Var}\left(\frac{\mathbf{S}_{ij}}{\rho}\right) = \frac{\left([\mathbf{X}\mathbf{X}^T]_{ij}^2 + [\sigma^2 + \mathbf{X}\mathbf{X}^T]_{ii}[\sigma^2 + \mathbf{X}\mathbf{X}^T]_{jj}\right)_{ij}\rho}{\rho^2} \to 0 \text{ and, } \mathbb{E}\left(\frac{\mathbf{S}}{\rho}\right) = \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}.$$

Therefore, $\mathbf{S}/\rho$ converges to a constant matrix, hence the marginal in the limit is $\mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I})$. In the second case, due to conjugacy (Murphy, 2023),

$$\mathbf{y}|\mathbf{X} \sim t_{\rho-n+1}(\mathbf{0}, \mathbf{X}\mathbf{X}^T + \beta\mathbf{I}),$$

which tends to the normal statement above as $\rho \to \infty$. Replacing the linear kernel with a

[2]We do not show how column independence arises, although this is trivial; plugging in vec(Y) into the results below and using the matrix normal distribution's definitions leads to the result.

general PD matrix recovers the GPLVM result.

A note on Wishart-normal conjugacy: Some common references state normal conjugacy results using inverted notation for Wishart distributions, so we prove the result above, using notation used in this thesis, from first principles. Let $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \kappa\mathbf{S})$ and $\mathbf{S} \sim \mathcal{W}^{-1}(\mathbf{M}, d)$. Then,

$$
\begin{aligned}
p(\mathbf{y}) &= \int p(\mathbf{y}|\mathbf{S})p(\mathbf{S})d\mathbf{S} \\
&\propto |\mathbf{M}|^{d/2} \int |\mathbf{S}|^{-(d+n+2)/2} \exp(-\kappa^{-1}\mathbf{y}^T\mathbf{S}^{-1}\mathbf{y}/2 - \mathrm{tr}(\mathbf{M}\mathbf{S}^{-1}))d\mathbf{S} \\
&\propto |\mathbf{M}|^{d/2} \int |\mathbf{S}|^{-(d+n+2)/2} \exp(-\mathrm{tr}(\kappa^{-1}\mathbf{y}\mathbf{y}^T\mathbf{S}^{-1})/2 - \mathrm{tr}(\mathbf{M}\mathbf{S}^{-1}))d\mathbf{S} \\
&\propto |\mathbf{M}|^{d/2} \int |\mathbf{S}|^{-(d+n+2)/2} \exp\left[-\mathrm{tr}\Big((\kappa^{-1}\mathbf{y}\mathbf{y}^T + \mathbf{M})\mathbf{S}^{-1}\Big)\right] d\mathbf{S} \\
&\overset{\int p=1}{\propto} |\mathbf{M}|^{d/2}|\kappa^{-1}\mathbf{y}\mathbf{y}^T + \mathbf{M}|^{-(d+1)/2} \\
&\propto |\mathbf{M}|^{d/2}|\mathbf{y}\mathbf{y}^T + \kappa\mathbf{M}|^{-(d+1)/2} \\
&\propto |\mathbf{M}|^{-1/2}\left[1 + \frac{1}{\kappa}\mathbf{y}^T\mathbf{M}^{-1}\mathbf{y}\right]^{-(d+1)/2} \qquad \text{(matrix determinant lemma)} \\
&\propto t_{d-n+1}\left(\mathbf{y}|\mathbf{0}, \frac{\kappa\mathbf{M}}{d-n+1}\right),
\end{aligned}
$$

which was the result used as part of the second case.

B.5 Variational views of t-SNE and UMAP

The proofs for theorem 3.8 are presented below.

Proof. of theorem 3.8, SNE case. The SNE probabilities were introduced in section 2.5. If we assume the distributions above,

$$
\begin{aligned}
q(\mathbf{A}'|\mathbf{Y}) &= \prod_i^n \mathrm{Categorical}(\mathbf{A}'_{i:}; \mathbf{Y}) = \prod_i^n \prod_{j\neq i}^n [v_{ij}^S]^{\mathbf{A}'_{ij}} \text{ and} \\
p(\mathbf{A}'|\mathbf{X}) &= \prod_i^n \mathrm{Categorical}(\mathbf{A}'_{i:}|\mathbf{X}) = \prod_i^n \prod_{j\neq i}^n [w_{ij}^S]^{\mathbf{A}'_{ij}}.
\end{aligned}
$$

This leads to the KL of eq. (3.18),

$$
\mathrm{KL}(q(\mathbf{A}'|\mathbf{Y})||p(\mathbf{A}'|\mathbf{X})) = \sum_i \mathrm{KL}(q(\mathbf{A}'_{i:})||p(\mathbf{A}'_{i:}|\mathbf{X})) = \sum_i \sum_{j\neq i} v_{ij}^S \log \frac{v_{ij}^S}{w_{ij}^S} = C_{SNE}.
$$

□

Proof. of theorem 3.8, t-SNE case. Similarly, we show here that assuming the t-SNE distributions as above leads to the objective of van der Maaten and Hinton (2008). Note that both sets of probabilities in t-SNE sum up to one, suggesting the categorical interpretation.

$$q(\mathbf{A}'|\mathrm{Y}) = \text{Categorical}(\text{vec}(\mathbf{A}')|\mathrm{Y}) = \prod_{i\neq j}^{n} [v_{ij}^t]^{\mathbf{A}'_{ij}} \text{ and}$$
$$p(\mathbf{A}'|\mathbf{X}) = \text{Categorical}(\text{vec}(\mathbf{A}')|\mathbf{X}) = \prod_{i\neq j}^{n} [w_{ij}^t]^{\mathbf{A}'_{ij}}.$$

Therefore the KL of eq. (3.18),

$$\text{KL}(q(\mathbf{A}'|\mathrm{Y})||p(\mathbf{A}'|\mathbf{X})) = \sum_{i\neq j} v_{ij}^t \log \frac{v_{ij}^t}{w_{ij}^t} = C_{t-SNE}.$$

□

Proof. of theorem 3.8, UMAP case. In the case of UMAP,

$$q(\mathbf{A}'|\mathrm{Y}) = \prod_{i\neq j}^{n} \text{Bernoulli}(\mathbf{A}'_{ij}; \mathrm{Y}) = \prod_{i\neq j}^{n} [v_{ij}^U]^{\mathbf{A}'_{ij}} [1 - v_{ij}^U]^{1-\mathbf{A}'_{ij}} \text{ and}$$
$$p(\mathbf{A}'|\mathbf{X}) = \prod_{i\neq j}^{n} \text{Bernoulli}(\mathbf{A}'_{ij}|\mathbf{X}) = \prod_{i\neq j}^{n} [w_{ij}^U]^{\mathbf{A}'_{ij}} [1 - w_{ij}^U]^{1-\mathbf{A}'_{ij}}.$$

Hence,

$$\begin{aligned}\text{KL}(q(\mathbf{A}'|\mathrm{Y})||p(\mathbf{A}'|\mathbf{X})) &= \sum_{i\neq j} \text{KL}(q(\mathbf{A}'_{ij})||p(\mathbf{A}'_{ij}|\mathbf{X})) \\ &= \sum_{i\neq j} v_{ij}^U \log \frac{v_{ij}^U}{w_{ij}^U} + (1 - v_{ij}^U) \log \frac{1 - v_{ij}^U}{1 - w_{ij}^U} = C_{UMAP}.\end{aligned}$$

□

This concludes our claim that (t-)SNE and UMAP correspond trivially to our variational framework, depicted in fig. 3.14.

B.6 A note on our notation w.r.t. (t-)SNE

Our notation (and only the notation) used in section 3.4.2 is flipped with respect to the (t-)SNE papers (Hinton and Roweis, 2002; van der Maaten and Hinton, 2008), i.e. we define the objective as $\mathrm{KL}(q\|p)$ rather than $\mathrm{KL}(p\|q)$, although the computation of the objective remains the same. To see why, note that the objective of Hinton and Roweis (2002) looks like,

$$\mathrm{KL}(\text{probabilities involving data}\|\text{probabilities involving latents}).$$

Noting that in typical variational models (such as VAEs and variational GPLVMs), the variational distributions are a function of data, and the model distributions are a function of latents (or parameters associated with the generative model), we propose that setting the data based probabilities to q is more natural (as they often represent approximate **posteriors**) and write the objective as $\mathrm{KL}(q\|p)$ as we do here. Noting this was one of the main inspirations for this work, along with the observation that many circularly-specified modelling methodologies can be written as variational inference algorithms. Note further that we denote high dimensional observed data by $\mathbf{Y}$ and low dimensional embeddings by $\mathbf{X}$, taking inspiration from how regression models are typically specified, whereas many older works such as (t-)SNE use reversed notation.

B.7 Choice of generative model in variational ProbDR

In this section, we provide generative models that could be used within our variational framework, all of which simply involve generative models that use the graph Laplacian or adjacency matrix.

Generative models, relating the learnt statistic $\hat{\mathbf{M}}$ to the data $\mathbf{Y}$, in the variational ProbDR framework allow for inferences to be made at the data level (e.g., reconstructions) using latent variables obtained through the various DR algorithms. In this section we consider generative models, specifically graph Gaussian processes, over the optional data edge of the graph in fig. 3.14, describing $p(\mathbf{Y}|\mathbf{M})$.

In models corresponding to (t-)SNE and UMAP, the adjacency matrix defined $\mathbf{A}'$ can be thought of as an adjacency matrix on a directed graph, as $\mathbf{A}'$ is typically asymmetric when

sampled. An option is to consider generative models over the directed graph $\mathbf{A}'$. As an example, Murphy (2012) describes the joint distribution of a Gaussian directed acyclic graphical model (a "Bayesian network") given the adjacency matrix. It appears as follows,

$$\mathbf{Y} \sim \mathcal{MN}(0, \mathbf{M}\mathbf{M}^T, \mathbf{I}),$$

where $\mathbf{M}$ is a lower triangular matrix (the Cholesky decomposition of the covariance) such that $\mathbf{M} = (\mathbf{I} - \mathbf{A})^{-1}$ and $\mathbf{A}$ is a row-normalised lower triangular adjacency matrix. This generative model is equivalent to:

$$\mathbf{Y}_{ij}|\text{pa}(i) \sim \mathcal{N}\left(\frac{1}{|\text{pa}(i)|}\sum_{k\in\text{pa}(i)}\mathbf{Y}_{kj}, 1\right),$$

where $\text{pa}(i)$ is the set of points that are parents to point i, and $|.|$ denotes the size of a set. Locally linear embedding can be interpreted using a similar model, as one first performs inference (in the MEU/GMRF view) for reconstruction weights that define a similar precision matrix over the data.

Alternatively, a Matérn-ν Gaussian process on a graph (Borovitskiy et al., 2021) can be employed after first symmetrising the adjacency matrix, within the generative model, $\mathbf{A}_{ij} = \mathbf{A}'_{ij} \vee \mathbf{A}'_{ji}$. This is used to define a suitable graph Laplacian $\mathbf{L}$ (e.g. the ordinary graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, or the symmetrically normalised matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{\dagger/2}\mathbf{A}\mathbf{D}^{\dagger/2}$) and a generative model can be specified as,

$$\forall i : \mathbf{Y}_{:i}|\mathbf{L} \sim \mathcal{N}\left(\mathbf{0}, \begin{cases} [\mathbf{L} + \beta\mathbf{I}]^{-1} & \text{Matérn-1 case} \\ \exp[-t\mathbf{L}] & \text{Matérn-}\infty\text{ case} \end{cases}\right).$$

The symmetrically normalised graph Laplacian is more useful in practice, as graph statistics (e.g, degrees) implied by the variational and model distributions on $\mathbf{A}'$ can be quite different. For example, the variational constraints on $\mathbf{A}'$ are extremely sparse, with the probability distributions being sharply bimodal (around zero and one). The model probabilities, however, defined by the latents $\mathbf{X}$, create a distribution on $\mathbf{A}'$, with the probabilities not being quite so bimodal. Hence, the graph Laplacians from the model and the variational constraint can be quite different, even after the optimisation of the latents. This highlights a significant weakness of such disentangled DR-then-regress frameworks: specifying a bad model is easy. In addition,

the usage of such methods can obfuscate exactly what is being modelled and what inferences can be made.

Note that the generative models above lack marginal consistency and have non-uniform marginal variances, although the latter problem can be addressed through converting the resulting covariance to a correlation matrix. The resulting covariances may also suffer from being diagonally dominant (although this too can be addressed by element-wise exponentiation) and do not correspond, except in edge cases to clean semantic modelling statements.

Therefore, as in section section 3.3, we conclude that although the ideas presented can lead to a unified and cleaner class of models, explicit modelling of covariances and edges through normal and Bernoulli models is recommended for traditional statistical modelling.

End.