Beyond independent masking in tabular self-supervision

Stuart Andrew Burrell

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy

Corpus Christi College September 2022

Declaration

I, Stuart Andrew Burrell of Corpus Christi College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. This thesis contains approximately **12567** words excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices.

Stuart Andrew Burrell
September 2022

Software

Experiments performed throughout this thesis were implemented from scratch using Python v3.8 (Python Core Team, 2019) and standard libraries. These included: PyTorch (Paszke et al., 2019), NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Scikit-learn (Pedregosa et al., 2011), PyTorch Metric Learning (Musgrave et al., 2020), Info-NCE (Elbers, 2021), Label-free XAI (Crabbé and van der Schaar, 2022) and Pandas (McKinney, 2010). Visualisations were generated with Matplotlib (Hunter, 2007) and Seaborn (Waskom, 2021). Documented code, complete with a tutorial, has been made publicly available at github.com/stuartburrell/tabular_ssl_suite.

Acknowledgements

I would like to give my sincerest thanks to Prof. Mihaela van der Schaar and Dr Fergus Imrie for their guidance, support, and many stimulating discussions. All of my family and friends who supported me through this challenging and rewarding year have my lasting gratitude, especially Heidi, Lawrence, and Nina.

Abstract

This thesis contributes to the development and understanding of masking schemes in tabular self-supervision. We demonstrate that correlated masking is a generally applicable technique that enhances state-of-the-art methods. This claim is validated on a family of synthetic datasets and several benchmarks. We also introduce *combination masks*, a technique to synthesise multiple masking schemes into a single approach to benefit from the advantages of each. We show combining independent and correlated masking in this way outperforms each individually on a family of seven proteomics datasets for predicting the efficacy of cancer drug treatments.

A masking strategy is one of numerous design choices in tabular self-supervision, yet the literature lacks a systematic approach for optimising these hyperparameters without at least some labelled data for cross-validation. To address this, we exploit recent advances in label-free explainable artificial intelligence to reveal why some representations are better than others. This deepens our understanding of why correlated masking is effective, and allows us to conjecture *general* properties that characterise good representations. We quantify these observations to form metrics that allow us to estimate approximately optimal hyperparameters on two standard benchmark datasets without access to *any* labelled data or downstream tasks. This is an initial contribution to a broader program of research, which aims to equip the practitioner with a suite of such tools and metrics to guide model development. As an application of this, we design and analyse a novel hierarchical architecture for ensembling encoders that uses collaboration between ensemble members to solve difficult tasks. This design outperforms standard ensembles and our tools explain why; collaboration acts as a form of regularisation and increases consistency amongst the feature importance scores.

Table of contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Contributions . . . 2

1.3 Outline . . . 2

2 Background **5**

2.1 Problem formulation . . . 5

2.2 Conceptual overview . . . 6

2.3 A unified view of tabular self-supervision . . . 9

2.3.1 Framework . . . 10

2.3.2 VIME . . . 12

2.3.3 Denoising autoencoders . . . 13

2.3.4 Context encoders . . . 13

2.3.5 SCARF . . . 13

2.4 Summary . . . 14

3 Self-supervision with correlated masks **15**

3.1 Building correlated masks . . . 16

3.1.1 Sampling process . . . 16

3.1.2 Theoretical justification 17
3.1.3 From copula samples to binary masks 18
3.2 Experiments 19
3.2.1 Setup 19
3.2.2 Synthetic data 20
3.2.3 Tabular benchmarks 23
3.3 Combination masks 26
3.3.1 Estimating the efficacy of cancer treatments 27
3.4 Summary 28

4 Optimisation through explainability **29**
4.1 Explainable artificial intelligence 30
4.2 Exploratory visual analysis 32
4.3 Metrics for label-free optimisation 36
4.4 Designing encoder ensembles 42
4.5 Summary 46

5 Conclusions **47**
5.1 Summary 47
5.2 Future directions 48
5.2.1 From correlation to causation 48
5.2.2 Dynamic masking schemes 50

References **51**

Appendix A Supporting material **59**
A.1 SubTab 59
A.2 Experimental details 61

A.3 Additional visualisations 62

Chapter 1

Introduction

1.1 Motivation

The growing presence of technology in our daily lives gives rise to vast quantities of data with potential for transformative social, academic and industrial impact. Deep neural networks, surging in popularity since the success of AlexNet (Krizhevsky et al., 2012), have been at the heart of a paradigm shift in supervised learning (LeCun et al., 2015). Yet, underlying state-of-the-art performances, in domains from computer vision (He et al., 2016; Yu et al., 2022) to language processing (Vaswani et al., 2017), is a critical dependency on time-consuming and costly manual annotation (Roh et al., 2021).

Self-supervised learning eases this burden by exploiting intrinsic structure within unlabelled data to learn an encoding function that yields informative representations of raw data. This encoder then may be used as a pre-processing stage to enhance the performance of future *downstream* supervised tasks. In this way it has been possible to obtain comparable or even superior performance to traditional supervision with orders of magnitude less labelled data (Chen et al., 2020; Dosovitskiy et al., 2021).

To date, the focus of the research community has been on self-supervision for the image and language domains Fang et al. (2022); Jing and Tian (2021); Min et al. (2021b). This has reimagined what is possible in these areas, particularly in the case of large language models such as BERT (Devlin et al., 2019) and GTP-3 (Brown et al., 2020). However, this success is, in part, based on leveraging our own human understanding of the strong spacial and semantic relationships in images and language, respectively (Borisov et al., 2021). Extending these methods to heterogeneous general tabular data is therefore challenging, obstructing the exploitation of data sources across industries including cybersecurity, energy, manufacturing, and healthcare. Motivated by this, the broad objective of this work is to develop techniques for enhancing the performance of self-supervised learning that are generally applicable to any tabular data.

1.2 Contributions

Our main contributions are summarised below. Definitions of technical terminology may be found in the corresponding introductory material of later chapters.

1. Demonstrate that correlated masking is a *generally applicable* method that may significantly enhance tabular self-supervision. This claim is validated across a wide-range of current reconstruction-based and contrastive methods on synthetic data, standard benchmarks, and a proteomics dataset.

2. Provide a publicly available[1] modular and extensible benchmarking suite for the testing and development of new methods in tabular self-supervision.

3. Introduce methods for optimising hyperparameters in self-supervised frameworks without access to *any* validation data or downstream tasks. This builds on recent advances in label-free explainable AI (Crabbé and van der Schaar, 2022) to characterise intrinsic and quantifiable properties of the best-performing representations.

4. Develop hierarchical ensembles of encoders that use collaboration between ensemble components to solve difficult pretext tasks. We show this outperforms standard ensembles and use our techniques based on label-free explainability to understand why.

1.3 Outline

Chapter 2 surveys general background material to set the scene, prepare the reader, and situate later chapters within exisitng literature. Specialist topics relevant only to a single part are contained in the corresponding chapter. We begin by formalising our problem statement and introducing important concepts in self-supervision, along with a review of important milestones in its development as a field. We then establish a unified mathematical notation that brings together five important methods used in tabular self-supervision: Denoising Autoencoders (Vincent et al., 2008), VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022),

[1]Available at: github.com/stuartburrell/tabular_ssl_suite

SubTab Ucar et al. (2021), and a tabular variant of Context Encoders (Pathak et al., 2016). This highlights their common elements and pinpoints their differences.

Chapter 3 demonstrates the impact of using correlated masking in tabular self-supervision. This form of masking uses the correlation structure of the input features to determine which features to corrupt, and has previously only been applied to specific tasks such as feature selection (Lee et al., 2022). In contrast, we show it is a *generally applicable* technique that enhances a variety of recent methods in the tabular domain, including VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), DAEs Vincent et al. (2008), SubTab (Ucar et al., 2021), and a tabular variant of Context Encoders (Pathak et al., 2016).

Correlated masks may be efficiently sampled using standard techniques and Gaussian copulas. A thorough mathematical description of this process is given in Section 3.1. Section 3.2 presents a series of validation experiments on standard benchmarks MNIST (LeCun et al., 2010), UCI Blog Feedback (Buza, 2014) and UCI Income (Kohavi and Becker, 1996), and illustrates how correlated masking mitigates a failure mode of existing methods using a carefully constructed family of synthetic datasets.

To conclude Chapter 3, in Section 3.3 we introduce *combination* masks, a hybrid approach that interpolates between different masking strategies. When applied to independent and correlated masking, this balances the learning of local and global information and significantly outperforms alternatives on a family of seven drug-discovery proteomics datasets derived from the Cancer Genome Atlas (TCGA) (Tomczak et al., 2015) and the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al., 2012).

Chapter 4 presents new methods for evaluating representations based on intrinsic information contained within the representations and encoder. This chapter is an initial contribution to an ambitious new program of research that aims to optimise hyperparameters in self-supervision without access to *any* labelled data or downstream tasks. To do this, we leverage recent work on label-free explainable artificial intelligence (XAI) (Crabbé and van der Schaar, 2022). A brief review of XAI and a mathematical introduction to the techniques required is given in Section 4.1. These techniques are applied in Section 4.2 to perform an exploratory visual analysis of *feature importance scores*, which measure the relative contribution of input variables to a final representation. This suggests several desirable properties of distributions over feature importance scores, which may characterise *good* representations. This is taken a step further in Section 4.3, where we formalize these intuitions into three metrics that quantitatively assess representations. These metrics are validated across 32 trained encoders on two datasets with different hyperparameters, and shown to approximately identify optimal configurations.

As a final application, we show how the insights of Chapter 4 may guide the development of ensembles of encoders. This is an alternative approach to fine-tuning the hyperparameters of a single encoder, where multiple encoders are trained and their representations aggregated. At the cost of compute, this reduces sensitivity to poor hyperparameters choices, since a downstream model may disregard unhelpful parts of the representation. Rather than using a standard ensemble however, we introduce a novel hierarchical architecture that allows for collaboration between ensemble members to solve difficult tasks. This approach outperforms standard ensembles and we use our visual and quantitative tools to explain why; collaboration regularises the encoder at the base of the hierarchy.

To conclude, Chapter 5 draws together our findings and proposes several avenues for further research. This includes moving beyond correlation to leverage advances in causal graph learning (Gao et al., 2022; Zheng et al., 2020), and the creation of dynamic masking strategies that evolve throughout training based on feedback from the metrics developed in Chapter 4.

Chapter 2

Background

This chapter introduces preliminary conceptual and technical knowledge required to situate our contributions within the literature. We set the scene by establishing a precise formulation of our problem. Then, we give a conceptual overview, discussing three broad families of techniques, and highlighting key milestones in the literature. To conclude, we narrow our focus to tabular data, and introduce a unified notation to bring together several important techniques within a coherent mathematical framework; VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), SubTab (Ucar et al., 2021), DAEs (Vincent et al., 2008) and tabular Context Encoders (Pathak et al., 2016). In doing so we depart from the exposition of the original authors, but aim to bring clarity to the commonality and distinctions of these methods.

2.1 Problem formulation

Let $\mathcal{D}_u = \{\mathbf{x}^{(i)} \in \mathbb{R}^d\}_{i=1}^{N_u}$ denote a set of *unlabelled* data and $\mathcal{D}_l = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N_l}$ a set of *labelled* data from the same feature distribution, with classes or regression targets given by $\mathbf{y}^{(i)}$. We are interested in the setting where $N_u >> N_l$. Our question is how to use $\mathcal{D}_u$ to enhance performance of a *downstream* supervised learning task on $\mathcal{D}_l$. In a multi-task setting, see (Zhang and Yang, 2021), there may be multiple downstream datasets $\mathcal{D}_l$ and associated tasks of interest. For brevity, we focus on a single task throughout.

Self-supervised learning tackles this problem by training an encoder $e : \mathbb{R}^d \to \mathbb{R}^k$, typically a deep neural network, to use as a pre-processing step to generate informative feature *representations* for the downstream supervised learning task. The codomain of e is known as a representation space of dimension $k \in \mathbb{N}$. This setup is illustrated in Figure 2.1.

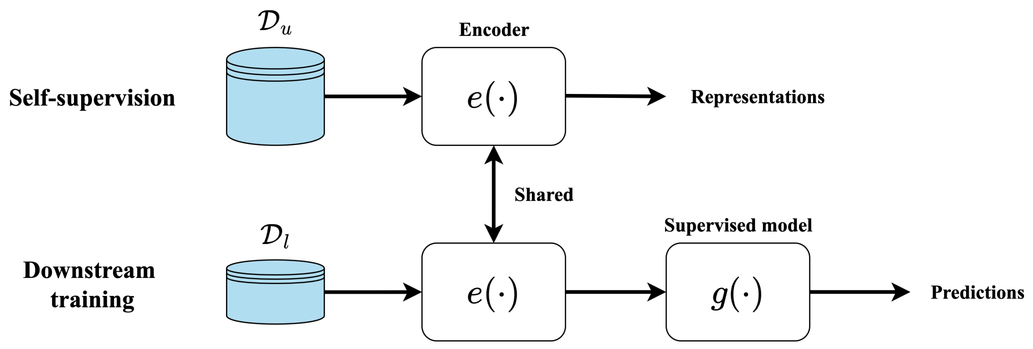

Fig. 2.1 The self-supervised learning pipeline. An unlabelled dataset $\mathcal{D}_u$ is used to train an encoder $e(\cdot)$, which is used as a feature pre-processing function for labelled data $\mathcal{D}_l$. This may help to train the downstream supervised model $g(\cdot)$ when $N_u >> N_l$. This figure was inspired by (Yoon et al., 2020, Figure 1 (Supplementary)).

There are a broad array of techniques for learning the encoder e, yet many share common aspects. Before diving into the technical details, we give a high-level overview of these techniques.

2.2 Conceptual overview

Self-supervised learning utilises supervised mechanisms despite the lack of targets. To overcome this, artificial *pretext* tasks are created with a known supervisory signal. Examples found in computer vision are often based on augmentations, and include solving jigsaw puzzles (Noroozi and Favaro, 2016) of permuted images, colourising gray-scale images (Zhang et al., 2016), or reversing transformations such as rotations (Gidaris et al., 2018). Typically, these tasks are solved by appending a *projection head* network to e, to be discarded after self-supervision. The intuition behind this argues that the intermediate representations obtained by e are likely to encode lower-level generalisable information, such as edges, shapes and spatial relationships, which may benefit a wider range of downstream tasks than fine details.

In all of these tasks the essential content we typically care about, such as object classes, is left invariant. This presents a challenge in the tabular domain; it may be hard or impossible to identify analogous augmentations, with concepts such as rotation lacking a general interpretation. However, two strategies have emerged for designing pretext tasks that are applicable to general tabular data, which we refer to as *reconstructive* and *contrastive* tasks. The remainder of this section is split into two parts, and gives a high-level introduction to both types of task. This prepares the reader for Section 2.3, where we delve into the mathematics of tabular methods and draw on both families of techniques.

Reconstructive tasks

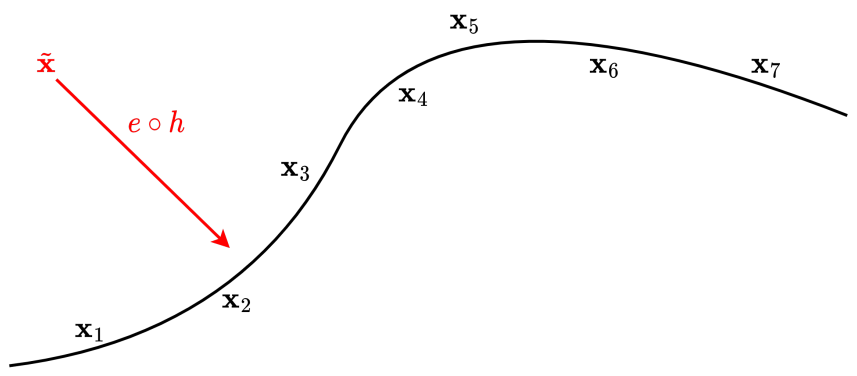

Fig. 2.2 The manifold interpretation of reconstructive pretext tasks (Vincent et al., 2008). The composition $e \circ h$ projects corrupted data back onto a lower dimensional data manifold. The intermediate representations obtained by $e(\cdot)$ may be thought to parameterise a co-ordinate system for the data manifold (Vincent et al., 2008).

Reconstructive tasks attempt to recover data from a corrupted version obtained through an algorithmic process. The term corruption is used, rather than augmentation as in the image domain, to capture the generality of this approach and emphasize it may not be label-preserving. Corruption could involve replacing certain variables in an input with a placeholder value such as zero (Vincent et al., 2008), the mean over that variable (Lee et al., 2022), or with random draws from the empirical marginal distribution of that variable (Yoon et al., 2020). Other forms of distortion such as additive Gaussian noise may also be used, as in (Ucar et al., 2021), though their suitability is dependent on the type of the data.

The idea underpinning reconstructive tasks is that of understanding which parts of an input are inconsistent with other parts, and is perhaps one of the most central ideas in self-supervised learning (LeCun, 2022). A simple but instructive mathematical line of reasoning justifying this is based on the idea of a data manifold (Vincent et al., 2008), and illustrated in Figure 2.2. For features living on a low-dimensional manifold within a high-dimensional ambient space, the process of corruption will typically result in points lying away from the manifold. Therefore, learning how to project them back onto their original values is a rephrasing of the task of learning the data manifold itself. This is helpful because supervised techniques often benefit from receiving lower-dimensional input; it allows for fewer parameters, simpler model complexity and mitigates the risk of over-fitting.

Reconstructive tasks are prevalent in language settings, with examples including predicting the completion of partial sentences (Devlin et al., 2019; Lan et al., 2020) or reversing random permutations of sentences (Lewis et al., 2020). BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020) are two particularly important methods in this area, and have had a major impact in language processing and translation (Min et al., 2021a).

However, reconstructive tasks are by definition *generative*, and implicitly model fine-scale structure during reconstruction that may not be beneficial to a downstream task. This may require more expressive models with a greater risk of overfitting, or cause models to focus on microscopic detail such as texture, rather than salient macroscopic information like the shape of an object. This motivates a competing family of contrastive techniques.

Contrastive tasks

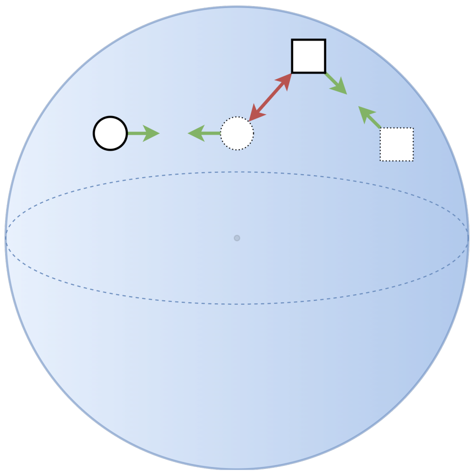

Fig. 2.3 An illustration of contrastive representation learning on the unit sphere. Similar points according to a contrastive loss function are attracted (green arrows) and dissimilar points are repulsed (red arrows). Examples of similar samples are typically generated through some form of data corruption or augmentation, illustrated here with dashed or solid perimeters.

Contrastive methods operate directly in the representation space by encouraging the embeddings of similar points to cluster, and those of dissimilar points to separate. This idea is illustrated in Figure 2.3. This requires us to define a notion of when two samples are similar. For images, combinations of data augmentations such rotations or color distortion (Chen et al., 2020) are commonly used to generate similar *positive* examples. Dissimilar *negative* examples may be obtained by comparing two different samples, or their augmentations. While computer vision has been a focal point of these techniques, SCARF (Bahri et al., 2022) demonstrates that corruption techniques used in general reconstructive tasks are effective augmentations for tabular data.

The intuition behind contrastive learning is simple, but requires a *contrastive loss function* to quantify similarity in a differentiable manner for back-propagation and learning. This relates to the wider field of *deep metric learning*, see Kaya and Bilge (2019). Early approaches compared pairs (Chopra et al., 2005) or triplets (Weinberger et al., 2005) of samples, using simple loss functions based on Euclidean norms to attract or repel positive or negative pairs. In recent years, the standard choice has become more sophisticated options such as InfoNCE (van den Oord et al., 2018), which built upon Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010), or close variants used in, for example, (Chen et al., 2020; Sohn, 2016; Tian et al., 2020; Wu et al., 2018). These losses are based on the *mutual information* between representations, a term that describes how dependent two variables are on each other. InfoNCE loss, formally defined in Section 2.3, maximises the mutual information between representations of positive pairs, while minimizing it for negative ones.

Despite contrastive learning being highly effective in many settings (Jing and Tian, 2021), it is not without drawbacks. LeCun (2022) argues the main challenge to these methods is their vulnerability to the curse of dimensionality. This is due to the need to effectively sample a diverse array of negative examples away from the data manifold, a task which grows exponentially harder with the dimension of the feature space (LeCun, 2022). Therefore, in this work we experiment with reconstructive and contrastive approaches, since both are salient to modern practice. In the next section we narrow our focus and provide concrete descriptions of the methods we use in later chapters.

2.3 A unified view of tabular self-supervision

This section establishes a unified notation bringing together a wide variety of techniques for tabular self-supervision. By doing so, we crystallise their similarities and differences, which are often clouded by stylistic choices in the original expositions. To begin, we define basic notation and detail the general framework, and then show how it permits a succinct description of VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), Denoising Autoencoders (DAEs) (Vincent et al., 2008) and Context Encoders (Pathak et al., 2016). We also cover SubTab (Ucar et al., 2021), though this is deferred to Appendix A for brevity. Throughout, we assume a familiarity with undergraduate level probability theory, calculus and basic mathematical notation, for example as found in (Deisenroth et al., 2020).

2.3.1 Framework

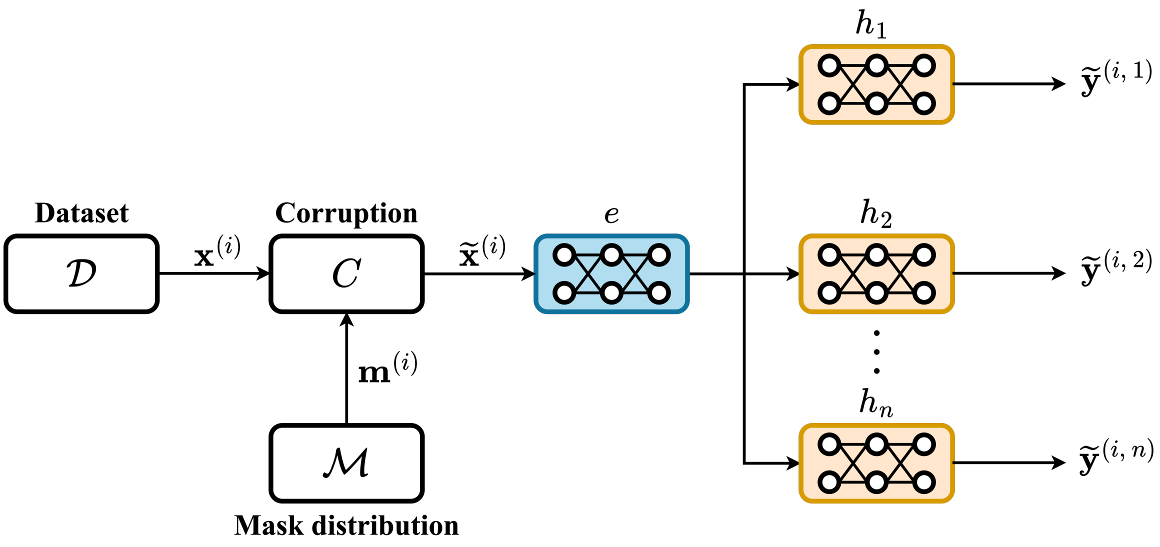

Fig. 2.4 A general framework for tabular self-supervision. An unlabelled data point $\mathbf{x}^{(i)}$ is corrupted by $C(\cdot,\cdot)$ based on a mask $\mathbf{m}^{(i)}$, and then propagated through an encoder e and projection heads $h_1,\dots,h_n$. This yields n outputs $\tilde{\mathbf{y}}^{(1,k)},\dots,\tilde{\mathbf{y}}^{(1,n)}$.

Recall from Section 2.1 that, given an unlabelled dataset $\mathcal{D}_u$, our aim is to learn an encoder $e:\mathbb{R}^d\to\mathbb{R}^k$ for use in a downstream supervised learning task. During self-supervision, one or more projection heads $h_1,h_2,\dots,h_n$ are appended to e, as illustrated in Figure 2.4. For each projection head, we aim to simultaneously solve the optimisation problems

$$\min_{e,\,h_k}\mathbb{E}\left[\, l_k(\mathbf{t}^{(k)},(h_k\circ e)(\tilde{\mathbf{x}}))\,\right]\quad\text{for } k=1,\dots,n, \tag{2.1}$$

where the expectation is over the joint distribution of *pretext inputs* $\tilde{\mathbf{x}}$ and pseudo-targets $\mathbf{t}^{(k)}$, and l_k is a loss function associated with h_k. By allowing the targets to depend on the projection head, we can consider each head to be performing a different pretext task on the common pretext input $\tilde{\mathbf{x}}$, as in VIME (Yoon et al., 2020).

To specify a method based on (2.1), we first must state an algorithm for sampling from aforementioned distribution of pretext inputs $\tilde{\mathbf{x}}$. To generate a single corrupted sample $\tilde{\mathbf{x}}^{(i)}$, we complete the following two steps.

1. Sample a mask $\mathbf{m}^{(i)}\in[0,1]^d$ from a mask distribution $\mathcal{M}$. Often, $\mathbf{m}^{(i)}\in\{0,1\}^d$ is *binary* and indicates which dimensions are to be corrupted.

2. Apply a *corruption* function C to obtain

$$C(\mathbf{m}^{(i)}, \mathbf{x}^{(i)}) = \widetilde{\mathbf{x}}^{(i)}$$

where $\widetilde{\mathbf{x}}$ is termed a *masked* or *corrupted* view of $\mathbf{x}$. In general, C may return multiple such views.

The masked view $\widetilde{\mathbf{x}}^{(i)}$ is then passed through e and each projection head, yielding

$$\widetilde{\mathbf{y}}^{(k,i)} = h_k \circ e(\widetilde{\mathbf{x}}^{(i)})$$

for $k = 1, \dots, n$. For contrastive methods, we may also compute the output for the uncorrupted sample, denoted

$$\mathbf{y}^{(k,i)} = h_i \circ e(\mathbf{x}^{(i)}).$$

The remaining question is how to specify the corresponding loss functions $\mathcal{L}_k$ and targets $\mathbf{t}_k$ for each projection head h_k. In practice, the total loss is a differentiable aggregation of the individual losses $\mathcal{L}_k$, such as a weighted sum, and is computed over the batches $\widetilde{\mathbf{X}} = \{\widetilde{\mathbf{x}}^{(i)} : i = 1, \dots, N\}$ and, if required, $\mathbf{X} = \{\mathbf{x}^{(i)} : i = 1, \dots, N\}$, with outputs denoted

$$\widetilde{\mathbf{Y}}^{(k)} = \{\widetilde{\mathbf{y}}^{(k,i)} : i = 1, \dots, N\} \quad \text{and} \quad \mathbf{Y}^{(k)} = \{\mathbf{y}^{(k,i)} : i = 1, \dots, N\}, \tag{2.2}$$

for $k = 1, \dots, n$. The joint optimisation problem (2.1) may then be tackled using backpropagation and standard optimisation techniques such as Adam (Kingma and Ba, 2015), assuming suitable architecture choices.

It is easily shown that a wide variety of tabular self-supervised methods are special cases of this framework, with variation arising from the number of projection heads n, the mask distribution $\mathcal{M}$, the corruption function C, the targets $\mathbf{t}_k$, and the loss functions $\mathcal{L}_k$. In the next two sections, we give examples of how to fill in the blanks for four techniques central to later chapters: VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), DAEs (Vincent et al., 2008), and Context Encoders (Pathak et al., 2016). For reference, in Appendix A.1 we also include complete details for the fifth method we use, SubTab (Ucar et al., 2021), which is simple but drawn out to describe and therefore omitted.

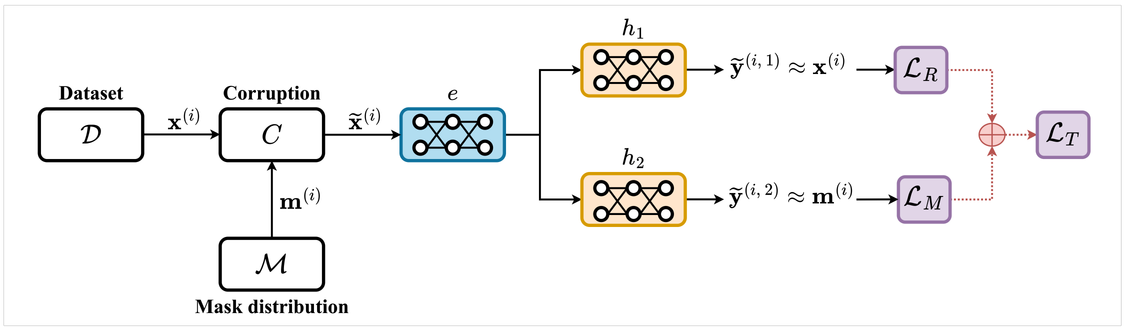

Fig. 2.5 The self-supervised component of VIME (Yoon et al., 2020). Input data $\mathcal{D}$ is first masked with swap corruption, and then passed through an encoder and two projection heads. The first is trained to reconstruct the original input and the second to predict which values were corrupted.

2.3.2 VIME

The key innovation of the self-supervised component of the VIME framework (Yoon et al., 2020)[1], is to not merely estimate a reconstruction of corrupted data, as in denoising autoencoders (Vincent et al., 2008), but to also to predict which values were corrupted. This is done using two projection heads, h_1 and h_2, as shown in Figure 2.5. The masks $\mathbf{m} \in \mathbb{R}^d$ are generated by independently sampling $m_i \sim \text{Bernoulli}(p)$, where $p \in [0,1]$ is a fixed parameter. The corruption function $C : \{0,1\}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is stochastic, and defined by

$$C(\mathbf{m},\mathbf{x}) = (\mathbf{1}-\mathbf{m}) \otimes \mathbf{x} + \mathbf{m} \otimes \mathbf{s} \tag{2.3}$$

where $\otimes$ denotes the Hadamard product, $\mathbf{1} \in \mathbb{R}^d$ is a vector of ones, and $\mathbf{s}$ is a vector formed by independently and uniformly drawing samples from empirical marginal distribution of each feature in $\mathcal{D}_u$. This form of corruption, known as *swap* corruption, prevents masked entries being trivially identified and is therefore critical to VIME.

For a batch of outputs $\widetilde{\mathbf{Y}}^{(1)}$ of size N from h_1 (see (2.2)), we compute the a reconstruction loss, but deal with continuous and categorical variables separately. That is,

$$\mathcal{L}_1 := \frac{1}{N}\left(\sum_{i=1}^{N}\sum_{j=1}^{d} \underbrace{((1-\mathbf{d}_j)\tilde{\mathbf{y}}_j^{(1,i)} - (1-\mathbf{d}_j)\mathbf{x}_j^{(i)})^2}_{\text{mean-squared error}} + \beta\, \underbrace{\mathbf{d}_j\mathbf{x}_j^{(i)} \log \mathbf{d}_j\tilde{\mathbf{y}}_j^{(1,i)}}_{\text{cat. cross-entropy}}\right), \tag{2.4}$$

[1]In this thesis we use VIME to refer to this *self-supervised* variant of VIME, which is sometimes also known as VIME-Self in the literature.

where $\beta > 0$ is tunable, and $\mathbf{d}$ is a binary indicator mask with $\mathbf{d}_i = 1$ if the j^{th} feature dimension is categorical and 0 otherwise. Throughout, subscripts, as in $\widetilde{\mathbf{y}}_j^{(1,i)}$, refer to the j^{th} component of the vector $\widetilde{\mathbf{y}}^{(1,i)}$.

The loss for output $\widetilde{\mathbf{Y}}^{(2)}$ of h_2 is a cross-entropy loss targeting the mask $\mathbf{m}^{(i)}$, given by

$$\mathcal{L}_2 := \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{d} \mathbf{m}_j^{(i)} \log \widetilde{\mathbf{y}}_j^{(2,i)}.$$

We then form the total loss as the weighted combination $\mathcal{L} = \alpha \mathcal{L}_1 + \mathcal{L}_2$ for a tunable parameter α.

2.3.3 Denoising autoencoders

A denoising autoencoder (DAE) contains a single projection head h_1. The mask $\mathbf{m}$ may be generated by uniformly choosing $\lfloor pd \rfloor$ components to be 1, and setting the remainder to be 0, or as in VIME, for some fixed parameter $p \in [0, 1]$. The corruption function $C : \{0, 1\}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is defined by

$$C(\mathbf{m}, \mathbf{x}) = (\mathbf{1} - \mathbf{m}) \otimes \mathbf{x} + \mathbf{m} \otimes \mathbf{0}, \tag{2.5}$$

where $\mathbf{0} \in \mathbb{R}^d$ is a zero vector. This is known as *zero* corruption. The loss for the single output $\widetilde{\mathbf{Y}}^{(1)}$ of $h_1 \circ e$ is a standard reconstruction loss, such as standard MSE loss or as in (2.4).

2.3.4 Context encoders

Context encoders mimic the setup of a DAE, except that the loss is computed only over the reconstructed elements. For example, with an MSE loss this is

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{d} \mathbf{m}_j (\mathbf{x}_j - (e \cdot x)(\widetilde{\mathbf{x}}^{(i)})_j)^2.$$

2.3.5 SCARF

The contrastive framework SCARF (Bahri et al., 2022) contains a single projection head h_1, as shown in Figure 2.6, and uses both corrupted and uncorrupted inputs. For some fixed parameter $p \in [0, 1]$, the masks $\mathbf{m}$ may either be generated as in VIME or by uniformly

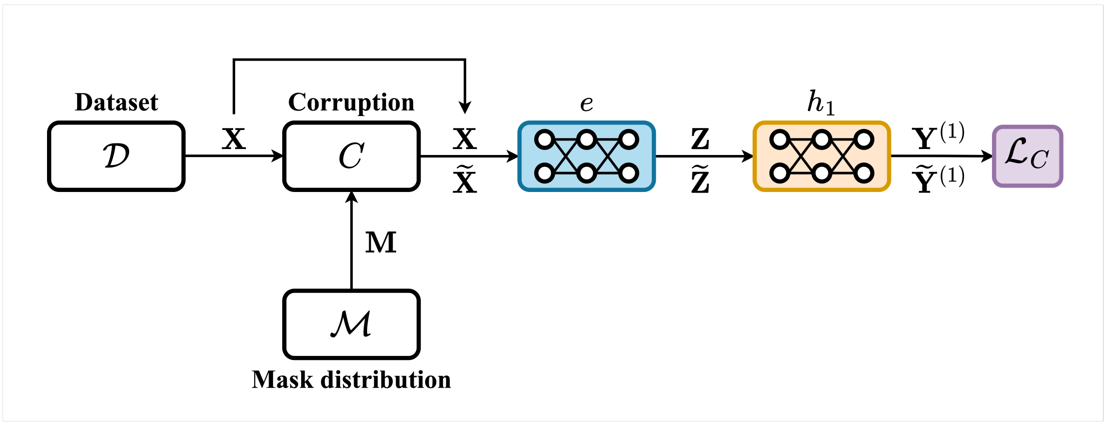

Fig. 2.6 The self-supervised framework SCARF (Bahri et al., 2022). InfoNCE loss, denoted $\mathcal{L}_C$, operates at the *batch* level, and so we consider batches denoted $\mathbf{X}$. In contrast to VIME (Yoon et al., 2020), uncorrupted and corrupted inputs flow through a pipeline that only contains a single projection head.

choosing $\lfloor pd \rfloor$ entries of each row of $\mathbf{m}$ to be 1, and setting the remainder to be 0. C is defined as in (2.3).

The loss is computed over the outputs $\widetilde{\mathbf{Y}}^{(1)}$ of corrupted inputs, but also requires the output from the corresponding uncorrupted inputs, denoted $\widetilde{\mathbf{Y}}^{(1)}$ (see (2.2)). InfoNCE loss (Gutmann and Hyvärinen, 2010; van den Oord et al., 2018) is used, and given by the formula

$$\mathcal{L}_1 := \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{\exp(\text{sim}(\widetilde{\mathbf{y}}^{(1,i)}, \mathbf{y}^{(1,i)})/\tau)}{\frac{1}{N}\sum_{j=1}^{N}\exp(\text{sim}(\widetilde{\mathbf{y}}^{(1,i)}, \mathbf{y}^{(1,j)})/\tau)}\right),$$

where $\text{sim}(\cdot)$ denotes cosine similarity and τ is a tunable *temperature* parameter that controls the sensitivity to negative samples (Wang and Liu, 2021).

2.4 Summary

This chapter lay technical foundations to prepare the reader for later sections. We formalised our problem statement and gave a conceptual overview of self-supervised learning, with a particular focus on the different families of pretext task. To clearly describe methods applicable to tabular self-supervision we developed a unified notation and framework. This brings clarity to how the device we consider in the next chapter, correlated masking, is generally applicable and may easily be injected into a broad array of current methods.

Chapter 3

Self-supervision with correlated masks

Masking processes are central to tabular self-supervision, and yet, the focus of the literature has been on developing other aspects of the pipeline. The mask defines the question posed by a pretext task, and we argue that asking better questions should ultimately lead to better representations. Specifically, we consider *correlated* masks that reflect the correlation structure of the input features. Concretely, this means that if two features $\mathbf{x}_i$ and $\mathbf{x}_j$ are highly correlated, they are likely to either both be masked, or neither. This is expected to be helpful during self-supervision, as it prevents networks learning trivial relationships, rather than meaningful structure, to solve pretext tasks.

To begin, we describe the process of generating correlated masks, using a technique based on Gaussian copulas (Lee et al., 2022). We then validate this approach across five methods: VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), denoising autoencoders (DAEs) (Vincent et al., 2008), Context Encoders (Pathak et al., 2016), and SubTab (Ucar et al., 2021). This includes experiments on a family of synthetic datasets that illustrate a significant failure mode of existing masking strategies, and on the standard benchmarks tabularised MNIST (LeCun et al., 2010), UCI Blog Feedback (Buza, 2014), and UCI Income (Kohavi and Becker, 1996). To conclude, we introduce the notion of *combination masks*. These build on the strengths of correlated and traditional independent masking, and are shown to outperform both on a family of 7 proteomics datasets for predicting the efficacy of cancer treatments (Barretina et al., 2012; Tomczak et al., 2015).

This chapter is accompanied by a software package[1] that provides a modular and extensible benchmarking suite for tabular self-supervision. It supports all of the methods we use, and

[1]github.com/stuartburrell/tabular_ssl_suite

our hope is to give other researchers convenient access to consistent baselines to aid in the development of new methods.

3.1 Building correlated masks

Existing techniques choose whether to mask each feature variable in a statistically independent fashion. For example, recall that VIME (Yoon et al., 2020) sets each element of the mask equal to an independent random draw from a Bernoulli distribution with probability p. In this section we will show how to construct masks that respect the correlation structure of the input.

It is a simple statistical problem to generate such masks, and there are multiple strategies. We opt for an approach used in Lee et al. (2022) for self-supervision enhanced feature selection that is based on Gaussian copulas, a standard tool in mathematical finance (Genest et al., 2009). This section is divided into three parts. First, we formally define this technique and describe an algorithm for obtaining samples. Second, we theoretically justify this approach and empirically validate our implementation with a simple example. Third, we show how samples from a Gaussian copula may be used to generate masks and visualise the differences between correlated and independent masking. The topics we introduce here are covered in various popular textbooks; for alternative expositions and background information see, for example, Joe (1997, 2014); Nelsen (2006).

3.1.1 Sampling process

For $d \in \mathbb{N}$, a function $C : [0,1]^d \to [0,1]$ is a copula if C is a joint multivariate cumulative distribution function (CDF) with uniform marginals. The Gaussian family of copulas are obtained by taking a d-dimensional multivariate normal distribution with *covariance* matrix $\mathbf{R}$, and applying a probability integral transform to each component. Denoted $C_{\mathbf{R}}$, this can be written

$$C_{\mathbf{R}}(U_1, \ldots, U_d) = \Phi_{\mathbf{R}}(\phi^{-1}(U_1), \phi^{-1}(U_2), \ldots, \phi^{-1}(U_d))$$

where ϕ is the CDF of $N(0,1)$ and $\Phi_{\mathbf{R}}$ of $N(\mathbf{0}, \mathbf{R})$.

To sample from $C_{\mathbf{R}}$, first decompose $\mathbf{R}$ using a Cholesky decomposition

$$\mathbf{R} = \mathbf{L}\mathbf{L}^T, \tag{3.1}$$

where $\mathbf{L}$ is a lower triangular matrix, which is viable since $\mathbf{R}$ is positive semi-definite. We then draw d samples from a univariate standard normal $N(0,1)$. These samples, denoted $\mathbf{X} \in \mathbb{R}^{d\times d}$ are then left multiplied by $\mathbf{L}$, giving

$$\mathbf{A} = \mathbf{L}\mathbf{X}.$$

Finally, to obtain a sample from $C_{\mathbb{R}}$, we compute

$$(U_1, U_2, \ldots, U_d) = (\phi(\mathbf{A}_1)), \ldots, \phi(\mathbf{A}_d)). \tag{3.2}$$

where ϕ is the CDF of a standard normal distribution $N(0,1)$.

3.1.2 Theoretical justification

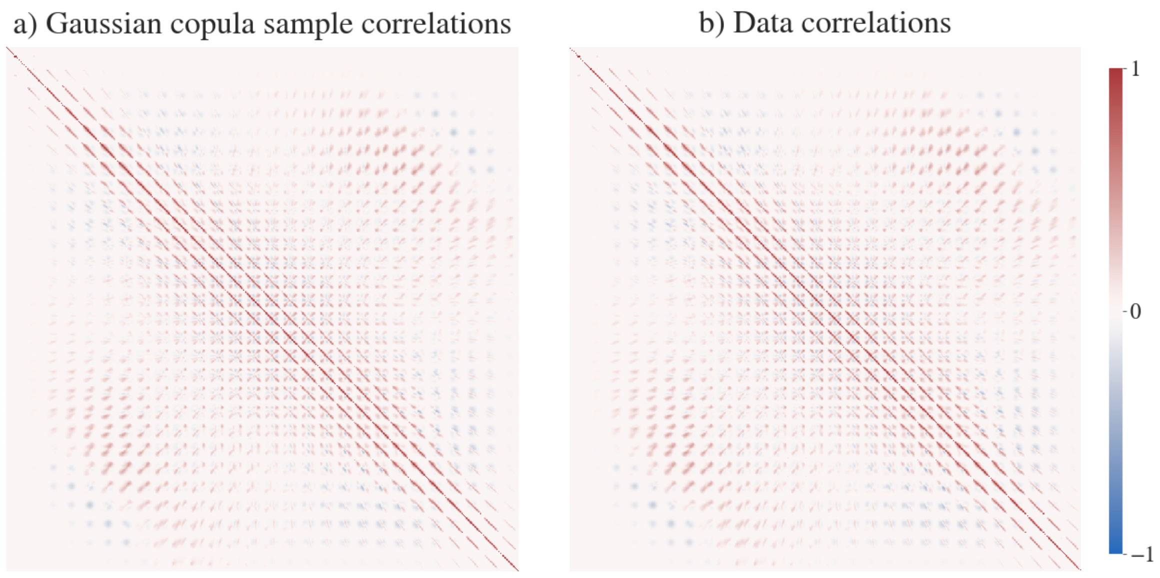

Fig. 3.1 The correlation structure of samples from a) Gaussian copula and b) the MNIST dataset (LeCun et al., 2010). As expected, the observed patterns are almost identical, with asymptotic equivalence.

The samples (3.2) are made to reflect the correlation structure of the unlabelled data $\mathcal{D}_u$ by defining $\mathbf{R}$ using the Pearson product-moment correlation coefficients given by

$$\mathbf{R}_{i,j} = \frac{\mathbf{C}_{i,j}}{\sqrt{\mathbf{C}_{i,i}\mathbf{C}_{j,j}}}, \tag{3.3}$$

where $\mathbf{C}_{i,j}$ is the covariance of features $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that it is valid to set a covariance matrix equal to a correlation matrix, since the latter is also positive semi-definite.

Correlation is carried from $\mathbf{R}$ to the copula samples, since $\mathbf{A}$ satisfies

$$\mathbb{E}(\mathbf{A}\mathbf{A}^T) = \mathbb{E}(\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L}^T) = \mathbf{L}\mathbb{E}(\mathbf{X}\mathbf{X}^T)\mathbf{L}^T = \mathbf{L}\mathbf{I}\mathbf{L}^T = \mathbf{R}$$

by (3.1). Hence, $\mathbf{A}$ has covariance matrix equal to the correlation matrix $\mathbf{R}$. It then follows from (3.3) that the correlation matrix of $\mathbf{A}$ is

$$\mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}$$

where $\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{i,i}$ equal to the variance of $\mathbf{A}_i$. Since the diagonal entries of $\mathbf{R}$ are 1 by definition, $\mathbf{D} = \mathbf{I}$, and we conclude the correlation matrix of $\mathbf{A}$ is also equal to $\mathbf{R}$. Since ϕ is positive and monotonically increasing, the final step (3.2) is *order* preserving. Moreover, ϕ is approximately linear around $0 = \mathbb{E}(\mathbf{A}_i)$ for all $i = 1, \dots, d$, and so, since correlation matrices are conserved by linear transformations with positive slope, we conclude that the correlation structure of the samples $(U_1, \dots, U_d)$ approximately reflects that of the original data, $\mathbf{R}$.

Figure 3.1 empirically validates this algorithm by visualising the correlation matrix over 1000 copula samples obtained using $\mathbf{R}$ derived from the MNIST dataset (LeCun et al., 2010). As expected, the correlation structure of the Copula samples shown in a) is similar to that in b), of the original data.

3.1.3 From copula samples to binary masks

It is simple to extend this procedure to generate binary masks. To sample a mask $\mathbf{m} \in \{0, 1\}^d$, we obtain Gaussian copula samples $(U_1, \dots, U_d)$ and set $m_i = 1$ if $U_i < p$ and 0 otherwise. Figure 3.2 illustrates the difference between correlated and independent masking on MNIST. As shown by the darker region surrounding the main diagonal in Figure 3.1, MNIST has large quantities of strong local correlations, due to the continuity of brush strokes in handwritten figures. Therefore, we see that correlated masking tends to remove larger contiguous blocks in comparison to independent masking. In the next section we validate the effectiveness of correlated masking across a wide range of current methods and datasets.

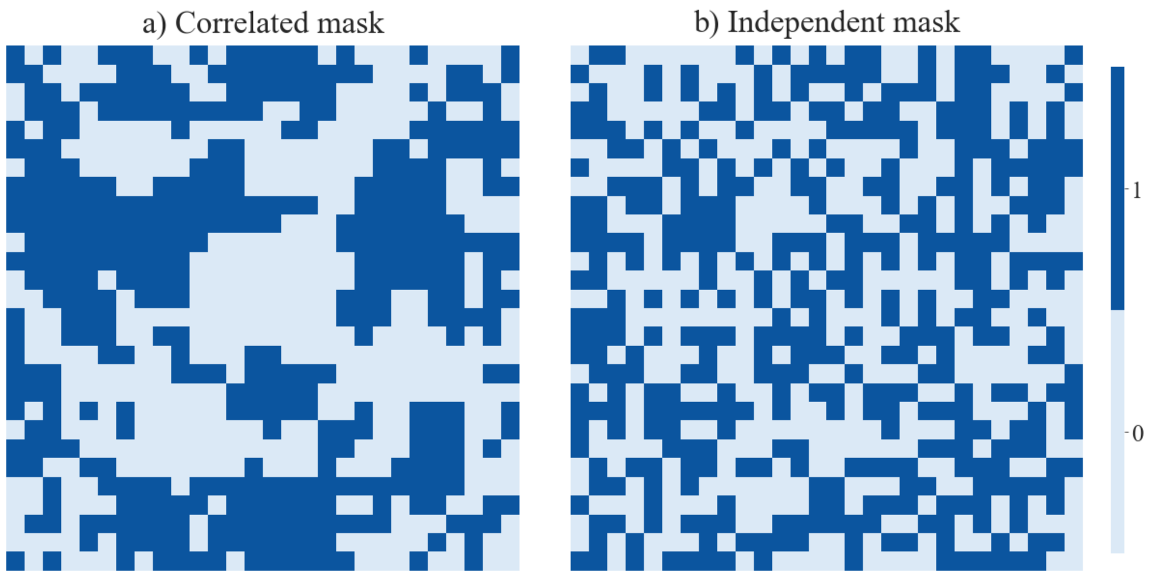

Fig. 3.2 Examples of independent and correlated masks for the MNIST dataset (LeCun et al., 2010).

3.2 Experiments

This section demonstrates through a series of experiments that correlated masking typically leads to greater downstream classification performance. First, we specify our general experimental setup and details necessary for our results to be reproduced.

3.2.1 Setup

Similar architectures were used across methods for consistency where possible. Encoders were a multi-layer perceptron with two hidden layers, while projection heads h_1 were a multi-layer perceptron with a single hidden layer. Hidden layers contained $\max\{30, d\}$ units, where d is the feature dimension which varies across datasets. Models were trained using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. For self-supervision, patience mechanisms that perform early-stopping after validation loss ceases to decrease after a pre-specified number of epochs were found to be inconsistent. Therefore, for maximum reproducibility, the number of training epochs for each dataset and model combination was fixed, and determined to be sufficient by monitoring loss on an unlabelled validation set. These quantities are given in Appendix A.2 for reference. We simulated the label-free setting where no downstream labelled validation is available, and therefore chose a typical default value of $p = 0.5$ for the corruption probability unless otherwise stated. Further model-specific hyperparameters are specified in Appendix A.

Downstream classification uses a multi-layer perceptron with 2 hidden layers. This was trained for 100 epochs with early-stopping and a patience of 20. As in (Yoon et al., 2020), 10% of downstream training data was held-out for validation, with early stopping based on validation classification performance. Accuracy or AUROC were used as performance metrics for downstream classifications tasks. The data partitions into unlabelled and labelled data are specified in the corresponding experiment. All models were implemented from scratch using PyTorch (Paszke et al., 2019), and we have made our testing suite publicly available[2].

3.2.2 Synthetic data

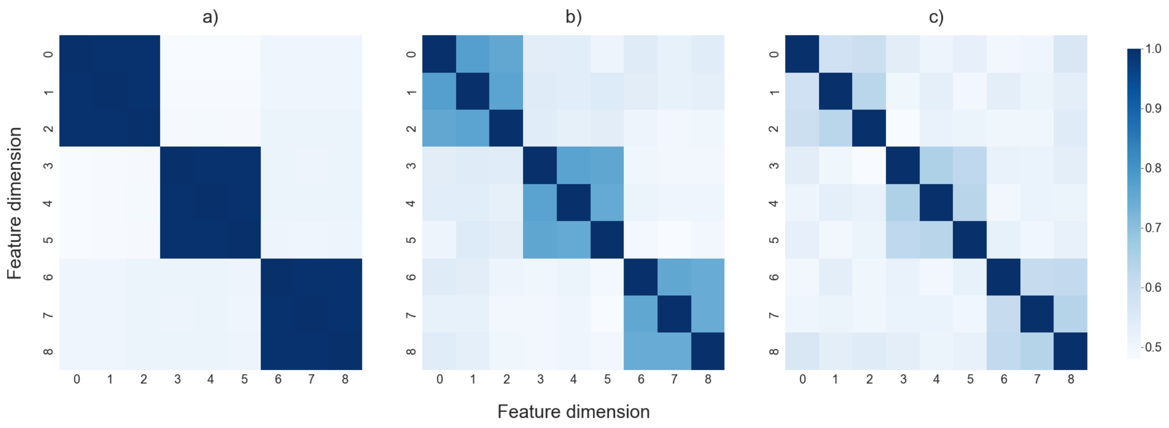

Fig. 3.3 The correlation structure of a 9-dimensional synthetic dataset built around 3 latent clusters. The within cluster variance is $\sigma = 0.1$ for a), $\sigma = 0.5$ for b), and $\sigma = 1.0$ for c). As σ increases the level of clustering dissipates.

Existing methods for tabular self-supervision may not learn informative representations if the network is able to exploit strong local correlations to solve pretext tasks while neglecting salient global structure. To illustrate this failure mode, we constructed a family of synthetic datasets. Each 9-dimensional row $\mathbf{x} \in \mathbb{R}^9$ was sampled as follows.

[2]github.com/stuartburrell/tabular_ssl_suite

Generation process

First, we independently sampled three standard normal latent variables

$$z_1, z_2, z_3 \sim N(0, 1).$$

These latents form cluster centres, around which components of $\mathbf{x}$ will lie. Specifically,

$$x_i \sim \begin{cases} N(z_1 + z_2 + \varepsilon, \sigma) & \text{if } i \in \{1, 2, 3\} \\ N(z_2 + z_3 + \varepsilon, \sigma) & \text{if } i \in \{4, 5, 6\} \\ N(z_1 + z_3 + \varepsilon, \sigma) & \text{if } i \in \{7, 8, 9\} \end{cases},$$

where $\varepsilon \sim N(0, \sigma_\varepsilon)$ is sampled once per row $\mathbf{x}$ and applied to all clusters, while σ controls the *within* cluster variance. To avoid the supervised learning task being trivial, we defined classes based on the latent variables z_1, z_2, z_3, which are discernible only by comparing the locations of clusters and implicitly solving a system of linear equations. In particular,

$$c_i = \begin{cases} 1 & \text{if } z_1 < z_2 < z_3 \text{ or } z_2 < z_3 < z_1 \\ 2 & \text{if } z_1 < z_3 < z_2 \text{ or } z_3 < z_1 < z_2 \\ 3 & \text{if } z_2 < z_1 < z_3 \text{ or } z_3 < z_2 < z_1 \end{cases}.$$

Therefore, to successfully perform classification an algorithm must be able to discern the ordering of the latents z_1, z_2, z_3 for particular example. Due to symmetry in the construction these classes are balanced, and so accuracy is considered relative to a baseline of 33%.

Our dataset contained 50000 points, of which we presumed only 100 were used for downstream training, and 5000 for downstream testing. The remaining 44900 points were considered unlabelled and used for self-supervision, with 1000 held-out as a validation set to monitor convergence during training. This lead to a training time of 200 epochs. We set the corruption parameter $p = 0.3$, as opposed to the default of value of 0.5 used elsewhere, due to the small number of features in this dataset.

Discussion

The results are given in Table 3.1. Correlated masking outperformed its independent counterpart across almost all cases when $\sigma = 0.1$ or $\sigma = 0.5$. At these noise levels, our results support the hypothesis that independent masking leads to reconstructive tasks exploiting

Table 3.1 Downstream classification performance on three synthetic datasets for a variety of reconstructive and contrastive tabular self-supervision approaches. There are 3 balanced classes, giving a random choice baseline of 33%. The best result between the two masking strategies for each method is given in bold. Mean and standard deviations are computed over 10 runs.

				Accuracy (%)		
Type	**Model**	**Encoder**	**Masking**	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1.0$
Supervised	2L-MLP	–	–	66.6 ± 7.5	56.7 ± 3.9	47.9 ± 7.7
Self-supervised	DAE	2L-MLP	Independent	59.8 ± 5.4	54.1 ± 7.9	52.7 ± 2.6
		2L-MLP	Correlated	**66.3 ± 8.8**	**59.9 ± 4.8**	**53.9 ± 2.1**
	VIME	2L-MLP	Independent	61.7 ± 8.2	56.6 ± 4.1	48.7 ± 5.8
		2L-MLP	Correlated	**67.4 ± 4.9**	**58.5 ± 3.1**	**48.9 ± 8.1**
	SCARF	2L-MLP	Independent	64.5 ± 9.2	54.5 ± 9.7	46.1 ± 7.8
		2L-MLP	Correlated	**67.8 ± 6.4**	**57.6 ± 5.4**	**47.1 ± 3.1**
	SubTab	2L-MLP	Independent	63.1 ± 6.9	**52.9 ± 2.8**	**43.9 ± 3.9**
		2L-MLP	Correlated	**67.4 ± 3.6**	51.8 ± 4.3	43.6 ± 3.7
	Context Encoder	2L-MLP	Independent	61.3 ± 6.4	57.0 ± 7.8	49.1 ± 3.8
		2L-MLP	Correlated	**66.2 ± 7.2**	**61.5 ± 2.0**	**49.7 ± 4.1**

within cluster correlations to easily reconstruct other cluster members. On the other hand, correlated masking is more likely to lead to an entire cluster being masked, forcing the model to discern the cluster values learning the hidden relationship between cluster centers.

Though to a lesser degree, correlated masking also improved the performance of the contrastive method SCARF, for which the previous argument does not apply. In this setting, correlated masking would be more prone to attracting representations between a point and a corrupted version with an entire cluster masked. The is useful, since the model learns that seemingly very distinct samples are actually similar based on the global relationship between cluster centers. On the other hand, independent masking will tend to learn only obvious similarities between points with partially masked clusters, whose similarity is evident without considering the larger scale relationships between cluster centres.

As σ increases, we see the performance discrepancies reduce. This is likely because the clustered structure begins to be obfuscated, and the within cluster correlations no longer confuse self-supervision by presenting a trivial solution to the pretext tasks. We conclude this discussion by noting that, while this dataset is synthetic, it does exhibit a structure that is plausible in many settings. For example, in a climate dataset on temperature measured at three weather stations within three regions, we would expect strong correlations between measurements in each region. This may obfuscate attempts to learn deeper patterns that occur at a larger scale *between* regions. That said, this example is specifically designed to exploit the weakness of independent masking and the strengths of correlated masking. In the next section, we level the playing field by considering three tabular benchmarks.

3.2.3 Tabular benchmarks

This section validates correlated masking on tabular MNIST (LeCun et al., 2010), UCI Blog Feedback (Buza, 2014), and UCI Income (Kohavi and Becker, 1996). These datasets are widely used to validate tabular self-supervised methods, for example, in (Bahri et al., 2022; Ucar et al., 2021; Yoon et al., 2020). Each is briefly introduced below. In all cases, the ratio of unlabelled to labelled training data we used was roughly 95-5%. 1000 unlabelled samples were held-out during self-supervision to monitor for convergence of validation loss and determine fixed stopping times, see Appendix A.2. A default value of $p = 0.5$ was used throughout. For consistency with (Bahri et al., 2022; Ucar et al., 2021; Yoon et al., 2020), we use accuracy to measure downstream performance

MNIST Handwritten Digits

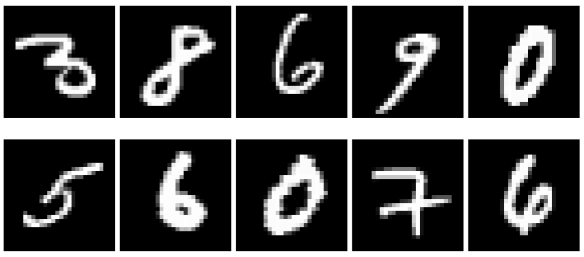

Fig. 3.4 Examples from the MNIST dataset (LeCun et al., 2010). In tabular experiments, these images are vectorised so that each forms a single row.

The MNIST dataset (LeCun et al., 2010) contains $60,000$ training and $10,000$ test grayscale 28×28 handwritten digits, see Figure 3.4. In tabular settings, each sample is converted into a 784-dimensional row vector. Each image was pre-processed by applying min-max normalization. The provided test set of $10,000$ images were used. $57,000$ training images were used for *unlabelled* self-supervision, and 3000 for downstream training.

UCI Blog Feedback

UCI Blog Feedback Buza (2014) is a tabular dataset of 60021 examples containing 280 features. These describe various attributes of blog posts, such as bag of word features and meta-data. The provided label is a regression target for the number of comments on a sample blog post in a 24 hour period (Buza, 2014), though as in VIME (Yoon et al., 2020), we convert this into a binary prediction task by predicting whether the number of comments is greater than 0. This task is slightly unbalanced, and a baseline based on predicting the mode class 0 is 63.7%. Min-max normalisation was applied to all features. We used $49,777$ unlabelled examples, 2620 downstream labelled training samples, and the provided test set of size 7264.

UCI Adult Income

UCI Income (Kohavi and Becker, 1996) is drawn from census data, and contains 30162 samples with 37 features describing attributes such as education history, job status, or age. Min-max normalisation was applied, which only impacts non-binary features. Our split contained $25,146$ unlabelled examples, 1508 labelled training examples, and 2000 testing examples.

Table 3.2 Classification performance for 2 supervised and 5 self-supervised methods on three benchmark datasets. Mean and standard deviations are computed over 10 runs.

				Accuracy (%)		
Type	**Model**	**Encoder**	**Masking**	**Blog**	**MNIST**	**Income**
Supervised	2L-MLP	None	None	76.9 ± 0.4	92.9 ± 0.6	81.7 ± 0.5
	DAE	2L-MLP	Independent	73.8 ± 0.5	93.1 ± 0.5	**83.3** ± **0.6**
		2L-MLP	Correlated	**75.6** ± **1.3**	**94.2** ± **0.5**	83.1 ± 0.7
Self-supervised	VIME	2L-MLP	Independent	77.6 ± 0.8	93.6 ± 0.6	81.6 ± 0.7
		2L-MLP	Correlated	**78.6** ± **1.1**	**94.5** ± **0.3**	**81.7** ± **0.7**
	SCARF	2L-MLP	Independent	77.6 ± 0.8	93.8 ± 0.4	**82.4** ± **0.6**
		2L-MLP	Correlated	**77.7** ± 0.8	**94.0** ± **0.4**	82.2 ± 0.9
	SubTab	2L-MLP	Independent	77.3 ± 0.4	94.9 ± 0.6	**83.4** ± **0.5**
		2L-MLP	Correlated	**77.9** ± **0.4**	**95.7** ± **0.3**	83.3 ± 0.4
	Context Encoder	2L-MLP	Independent	78.5 ± 0.6	94.1 ± 0.5	81.1 ± 1.4
		2L-MLP	Correlated	**79.2** ± **0.6**	**94.7** ± **0.2**	**81.6** ± **0.2**

Discussion

Table 3.2 presents classification accuracy for each dataset over the five methods: VIME, SCARF, SubTab, Context Encoders and DAEs. For both Blog and MNIST, correlated masking is an effective technique, consistently improving performance by around 1-2% in absolute terms across a wide range of tabular self-supervised methods. For UCI Income, both

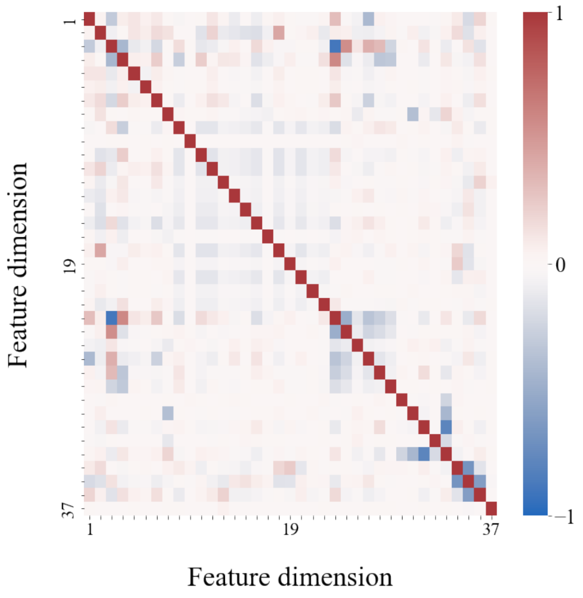

Fig. 3.5 Visualising feature correlations for the UCI Income (Kohavi and Becker, 1996) dataset.

methods were approximately comparable. This is to be expected, as given closer inspection of the correlation structure of Income features shows only very weak correlations between feature variables, see Figure 3.5. In such cases, the two masking regimes differ only negligibly and we expect comparable performance. In practice, the user may inspect or quantitatively summarise the correlation structure to determine whether there are sufficient relationships to expect correlated masking to have an impact. For example, the mean of the correlation matrix of UCI Income is 0.008, far lower than 0.02 and 0.07 for MNIST and UCI Blog, respectively. However, we never witness a significant degradation due to correlated masking, suggesting even novice practitioners may apply this technique without significant risk.

An importance difference is seen between the contrastive approach SCARF (Bahri et al., 2022) and other methods, which all rely on some form of pretext reconstruction task. For SCARF, both approaches were quite comparable asymptotically, as shown in Table 3.2. However, we did observe that on UCI Blog correlated masking converged much quicker, reaching its approximate final performance levels after just 200 epochs. At this stage however, independent masking was lagging behind by almost a 1% in accuracy, and required further 100 epochs to close the gap, see Figure A.2 in Appendix A.3. This demonstrates that correlated masking is playing a different role in contrastive settings. In contrastive frameworks, the importance of good quality *negative* examples is known, see (Chen et al., 2020; Tian et al., 2020). Therefore, we hypothesise that since correlated masks correspond to more meaningful distortion, they therefore more frequently and easily generate good negative examples. However, given sufficient training, independent masking will eventually generate such examples too, with the reduced frequency not appearing to impact performance.

3.3 Combination masks

So far we have witnessed an array of cases where correlated masking is beneficial, and would be a recommended technique across tabular self-supervision. By masking correlated collections of variables, this scheme may be thought to learn deeper, less obvious relationships. These may be characterised as *global*, since instead of pairwise dependencies they may often depend on larger numbers of variables. However, as we shall see in Chapter 4, we should not completely disregard the information provided by independent masking, either. A method that focuses more on pairwise relationships may well encode *local* information that correlated masking does not.

For many downstream tasks, we may desire representations that successfully encode information at both global and local scales. Motivated by this, we construct method that synthesising both form of mask within a single scheme. For two different masks $\mathbf{m}_1$ and $\mathbf{m}_2$ and a parameter $0 \leq \gamma \leq 1$, we set

$$\mathbf{m} = (\mathbf{u}_{<\gamma}) \otimes \mathbf{m}_1 + (1 - \mathbf{u}_{<\gamma}) \otimes \mathbf{m}_2.$$

where $\mathbf{u} \in \mathbb{R}^d$ is a binary vector with entries sampled from Uniform$(0, 1)$, and

$$(\mathbf{u} < \gamma)_i = \begin{cases} 1 & \text{if } \mathbf{u}_i < \gamma \\ 0 & \text{if } \mathbf{u}_i \geq \gamma \end{cases}.$$

In our experiments, we set $\mathbf{m}_1$ to be a correlated mask and $\mathbf{m}_2$ to be a standard independent mask. Hence, we recover independent masking when $\gamma = 0$ and correlated when $\gamma = 1$. We note that this scheme readily extends to more than two schemes, at the expense of additional parameters to define the weighting.

As a final application in this chapter, we chose to validate all three masking strategies; independent, correlated and combined, with a larger scale experiment using a family of 7 proteomics datasets on estimating the efficacy of cancer treatments. This is a more deserving test bed than arguably overused benchmarks from the literature; as the ratio of unlabelled to labelled data grows, it is for such data with rich and complex structure that tabular self-supervision will likely thrive.

3.3.1 Estimating the efficacy of cancer treatments

This validation experiment uses a dataset built from the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al., 2012) and The Cancer Genome Atlas (TCGA) (Tomczak et al., 2015). The former contains 889 samples of 196 protein expressions taken from cancer cells. We consider associated drug responses for 7 cancer treatments, which are available for 458 of these samples to form our downstream labelled data. The remaining 458 are unlabelled. This is not a scale where we expect self-supervision to be successful, and for this reason these unlabelled samples are combined with a further 7329 samples from (Tomczak et al., 2015). The resulting dataset contained 124 featured dimensions.

We created a binary classification task by labelling cell lines with responses in the top 25% as *responders* (label 1) and those in the bottom 75% as *non-responders* (label 0). This is an unbalanced task, the ratio of non-responders to responders 3 to 1. Therefore, AUROC is used as our performance metric to assess performance. As a representative example, we chose the method VIME (Yoon et al., 2020) with $\alpha = 2$ and $p = 0.5$, though others choices would be suitable. The combination mask parameter γ was set to 0.5. Table 3.3 shows that, even though thee amount of unlabelled data is modest, self-supervision yields improvements in all cases. Correlated and combination masking also outperform independent masking across all but one dataset, with the combined scheme optimum for 5 out of the 7. This supports our hypothesis that a hybrid approach that balances local and global information is often promising, and tuning of γ could potentially further increase the benefit.

Table 3.3 Downstream AUROC performance scores for estimating drug responses using the CCLE (Barretina et al., 2012) and TCGA (Tomczak et al., 2015) proteomics datasets. Mean and standard deviations are computed over ten runs.

		Drug							
Masking	γ	**Panobinostat**	**L-685458**	**AEW541**	**Topotecan**	**TAE684**	**PF2341066**	**Irenotecan**	**Average**
None	–	.747 ± .025	.660 ± .039	.641 ± .036	.672 ± .076	.652 ± .029	.649 ± .018	.599 ± .004	.660 ± .032
Independent	0.0	.749 ± .009	.696 ± .008	.613 ± .021	.716 ± .009	**.657 ± .013**	.652 ± .012	.640 ± .023	.675 ± .014
Correlated	1.0	.780 ± .005	**.705 ± .006**	.639 ± .017	.708 ± .007	.637 ± .014	**.676 ± .011**	.693 ± .027	.691 ± .012
Combination	0.5	**.784 ± .006**	**.705 ± .009**	**.646 ± .009**	**.718 ± .007**	.644 ± .012	.667 ± .018	**.705 ± .020**	**.696 ± .011**

3.4 Summary

This chapter investigates the use of correlated masking for self-supervision. Strong motivation for this technique is given by constructing a synthetic dataset which demonstrates a significant failure mode of existing techniques, showing how strong local correlations may obfuscate attempts to learn deeper latent structure. We extensively validate this method and show significant performance gains across a wide range of techniques: VIME (Yoon et al., 2020), SCARF (Bahri et al., 2022), SubTab (Ucar et al., 2021), DAEs (Vincent et al., 2008), and Context Encoders (Pathak et al., 2016). To conclude, we consider a family of 7 proteomics datasets on estimating the efficacy of cancer treatments. Correlated masking is effective in this setting, but often a novel approach based on combining masking schemes is best. This allow finer control of task difficulty, and we argue balances the learning of local and global information. However, this introduces a further hyperparameter γ that controls the weighting of each scheme, further motivating the need for effective hyperparameter optimisation techniques. In the next chapter, we address this issue by introducing methods for approximating optimal hyperparameters using explainable AI, without access to *any* labelled validation data or downstream tasks.

Chapter 4

Optimisation through explainability

Downstream performance is widely accepted as the method of choice for evaluating representation quality. If labelled examples are available during self-supervision, standard cross-validation techniques may be used to optimise network architectures and masking hyperparameters, such as p, which controls the degree of corruption. However, in such settings semi-supervised approaches (Pise and Kulkarni, 2008) that directly incorporate labelled examples within the representation learning framework might be preferred, such as the semi-supervised variant of VIME (Yoon et al., 2020). In this chapter we ask how hyperparameter optimisation may take place without access to *any* downstream labelled data.

To do this we argue representation quality may be inferred *intrinsically* from characteristics of the representations and encoder alone. We present evidence for this by performing an exploratory visual analysis of 16 self-supervised models trained on the MNIST dataset (LeCun et al., 2010), and leverage recent advances in label-free explainable artificial intelligence (Crabbé and van der Schaar, 2022). In contrast to a visualisation technique such as t-SNE (van der Maaten and Hinton, 2008), which offers a larger scale picture of an embedding space, the techniques we consider allow us to analyse properties of *individual* representations.

Based on these insights, we postulate general desirable properties of representations, and define associated quantitative metrics that may be used for concrete optimisation. These quantitative metrics are then validated on MNIST and a second dataset, UCI Blog Feedback, *unseen* during our exploratory analysis. This is an ambitious line of research, and we do not claim a single metric for optimising any representations may even exist. Instead, we take a practical approach, and suggest that over time a suite of such tools for *intrinsically* evaluating representations may be developed that may guide the machine learning practitioner during the development of self-supervised models. To begin, we introduce the reader to the field of explainable artificial intelligence (XAI), with a particular emphasis on the label-free setting (Crabbé and van der Schaar, 2022) that is directly relevant to our work.

4.1 Explainable artificial intelligence

In traditional statistical modelling, interpreting the predictions of techniques such as Generalised Linear Models (GLMs) (McCullagh and Nelder, 1989) may often reduce to simply comparing the relative sizes and roles of a handful of parameters. The success of modern deep learning, however, has come at the cost of a dependence on model complexity, with large language models such as GPT-3 (Brown et al., 2020) containing over a hundred billion parameters. This shift in scale has necessitated the development of the field of explainable artificial intelligence (XAI) (Das and Rad, 2020). This is a multi-faceted program of research, built on a need for AI that is safe, trustworthy, robust and transparent (Adadi and Berrada, 2018). It aims to unpack the decision making process of machine learning models, and understand *why* they produce the outputs they do, when at first glance they appear to be a 'black box'. This promises to open the door to high-stakes applications of AI in fields such as medicine and law (Barredo Arrieta et al., 2020), but also has the potential to aid scientific discovery, since our models may reveal reasoning for predictions that previously eluded us (Adadi and Berrada, 2018).

One approach to XAI aims to design and build models that are inherently interpretable (Caruana et al., 2015; Letham et al., 2015; Ustun, 2016; Xu et al., 2015). This approach is often too restrictive on model complexity, however, motivating the class of *post-hoc* techniques (Lipton, 2018) that are model agnostic and generally applicable. The post-hoc techniques we consider compute *feature importance*, a quantification of the importance of an input feature on a model prediction. Popular methods that fall under this category include Integrated Gradients (Sundararajan et al., 2017), Lime (Ribeiro et al., 2016), and Shap (Lundberg and Lee, 2017). See also (Crabbé and Van Der Schaar, 2021; Fong and Vedaldi, 2017; Shrikumar et al., 2017). In various ways, these techniques quantify the sensitivity of a prediction to perturbation in the input, often through approximation. For example, Lime (Ribeiro et al., 2016) uses a local linear approximation of the model around the prediction, while Integrated Gradients accumulates gradients along a path from the input to a common baseline, using the logic that a steeper gradient landscape around a feature indicates heightened sensitivity and importance. However, for each of these methods, the predictions we are aiming to interpret relate to a *supervised setting*, and are a single scalar value, such as a class. To instead consider a label-free setting with general multi-dimensional model outputs an extension was introduced in (Crabbé and van der Schaar, 2022).

Conveniently, label-free feature importance (Crabbé and van der Schaar, 2022) builds on top of a wide variety of traditional methods. To see how, let $f : \mathbb{R}^d \to \mathbb{R}^k$ denote a model of interest. To compute the feature importance scores of an input $\mathbf{x} \in \mathbb{R}^d$, denoted $a_i(f, \mathbf{x})$ for $i = 1, \ldots, d$, the model f is wrapped within a function $g_{\mathbf{x}}(f, \cdot) : \mathbb{R}^d \to \mathbb{R}$ given by

$$g_{\mathbf{x}}(f, \mathbf{t}) = \langle f(\mathbf{x}), f(\mathbf{t}) \rangle, \tag{4.1}$$

where $\langle \cdot \rangle$ denotes an inner product. Typically, this will be the standard inner product on $\mathbb{R}^d$. Importantly, note that the function $g_{\mathbf{x}}$ produces a scalar, and may therefore be passed directly to standard feature importance methods. This leads to the label-free importance scores, denoted $b_i(\mathbf{x}, f)$, defined by

$$b_i(\mathbf{x}, f) = a_i(g_{\mathbf{x}}, \mathbf{x}). \tag{4.2}$$

The motivation for $g_{\mathbf{x}}$ comes from the equation

$$\sum_{j=1}^{k} f_j(\mathbf{x}) a_i(f_j, \mathbf{x}), \tag{4.3}$$

which quantifies the overall importance of $\mathbf{x}_i$ as a weighted sum over its influence on each dimension of the output, denoted f_j. However, since this must be computed for each i and sums over j, it requires $d \times k$ evaluations of a_i per sample. However, (Crabbé and van der Schaar, 2022) observe that this may be bypassed if $a_i(f, \mathbf{x})$ is linear with respect to its first argument, meaning

$$a_i(\lambda f + \gamma g, \mathbf{x}) = \lambda a_i(f, \mathbf{x}) + \gamma a_i(g, \mathbf{x}).$$

This holds for most methods, including Integrated Gradients that we use later (Crabbé and van der Schaar, 2022). Using linearity, (4.3) trivially reduces to

$$b_i(f, \mathbf{x}) = a_i \left(\sum_{j=1}^{k} f_j(\mathbf{x}) \cdot f_j, \mathbf{x} \right) = a_i(g_{\mathbf{x}}, \mathbf{x}) \tag{4.4}$$

recovering the definition of $g_{\mathbf{x}}$ from (4.1).

By identifying f with an encoder e such as those trained in Chapter 3, this machinery allows us to probe deeper into how design choices such as correlated masking actually change representations. This identifies qualitative and quantitative properties that characterise representations with superior downstream performance.

4.2 Exploratory visual analysis

In Chapter 3, we saw how different forms of masking impact downstream performance. The story does not end there, and we may ask *why* one representation is better, or more informative, than another. We investigate this by computing and visualising feature importance scores using (4.2) with the underlying method Integrated Gradients (Sundararajan et al., 2017), though methods such as Shap (Lundberg and Lee, 2017) and Lime (Ribeiro et al., 2016) would also be suitable.

Integrated gradients considers an accumulation of gradients along a path from a baseline input $\mathbf{x}'$ to an input of interest $\mathbf{x}$. Formally, this is computed as

$$a_i(f, \mathbf{x}) = (\mathbf{x}_i - \mathbf{x}'_i) \times \int_0^1 \frac{\partial f(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial \mathbf{x}_i} \, d\alpha$$

which may be combined with (4.2) to compute the label-free version $b_i(f, \mathbf{x}) = a_i(g_{\mathbf{x}}, f)$, since linearity is immediate. For MNIST, and in later sections with UCI Blog Feedback, we found a zero baseline $\mathbf{x}' = \mathbf{0}$ to be effective, corresponding to a black image for MNIST and an empty blog post for UCI Blog Feedback, noting that the majority of the features in this latter case are word frequencies.

The impact of independent and correlated masking

Figure 4.1 visualises the feature importance of three examples taken from the MNIST dataset for two VIME derived encoders; one with independent masking and another with correlated masking. We surveyed a far larger number of examples, and chose these to exhibit three prominent trends that we discovered. Figur 4.1 a) gives an example showing that independent masking may lead to somewhat arbitrary levels of concentrated high importance, often at the edges of a digit. This was perhaps the most common pattern we found, and suggests independently masked models are more prone to being distracted by irrelevant local detail. In contrast, correlated masking typically led to a smoother landscape of importance scores, evenly spread across the essential content. Figure 4.1 b) gives an example where independent masking leads to missing patches and discontinuity in the paths of high feature importance through a digit. This suggests a degree of over-fitting, with the model placing greater importance on the central bar of the digit that happens to be present for many other numbers, such as 1 and 7. In contrast, no examples with correlated masking we saw lead to this behavior, showing a better respect for the large scale structure of the digits. Finally, in Figure 4.1 c), we see where independent masking is actually *preferable*; we often observed that minor stylistic features were picked up in the importance scores much more frequently than with correlated masks. As a result, the digits traced out for the correlated encoder often contained thinner outlines. These three examples support our conjecture from Chapter 3 that correlated masking led to a greater focus on global information while independent masking picks up more local relationships. For many downstream tasks, global information may well be more pertinent and correlated masking preferred, though when small details are relevant to the downstream task, we expect this to be reversed.

Importance collapse and task difficulty

To gain a more complete picture of an encoder $e : \mathbb{R}^d \to \mathbb{R}^k$ it may be useful to look at the typical behaviour of feature importance scores. Therefore, for a sample of inputs $\{\mathbf{x}^{(i)}, \dots, \mathbf{x}^{(N)}\}$, we define the *aggregated* feature importance to be the vector $\mathbf{A}(e) \in \mathbb{R}^d$, where

$$\mathbf{A}(e)_j = \sum_{i=1}^{N} b_j(e, \mathbf{x}^{(i)}). \tag{4.5}$$

For visualisation purposes, on MNIST we reshape $\mathbf{A}(e)$ into a 28×28 image.

Figure 4.2 visualises the aggregated feature importance for 10000 MNIST test examples (scaled by a uniform constant). For $p = 0.2$, both masking schemes show large amounts of irregularity in $\mathbf{A}(e)$, though this is more pronounced for independent masking. Specifically, large numbers of important central pixels are ignored. This shows that for small p, the self-supervision framework is able to solve the pretext tasks while completely dismissing large amounts of salient input. This shows that the pretext task in this case is too *easy*. As p increases to 0.5, regularity increases, particularly in the central region. Here, independent masking takes into account a larger portion of the input overall, suggesting a lack of focus that could explain lower downstream performance. As p increases further, scores for both collapse in a concentrated central region. This suggests the task is too hard, with the model learning to disregard the majority of the input as noise.

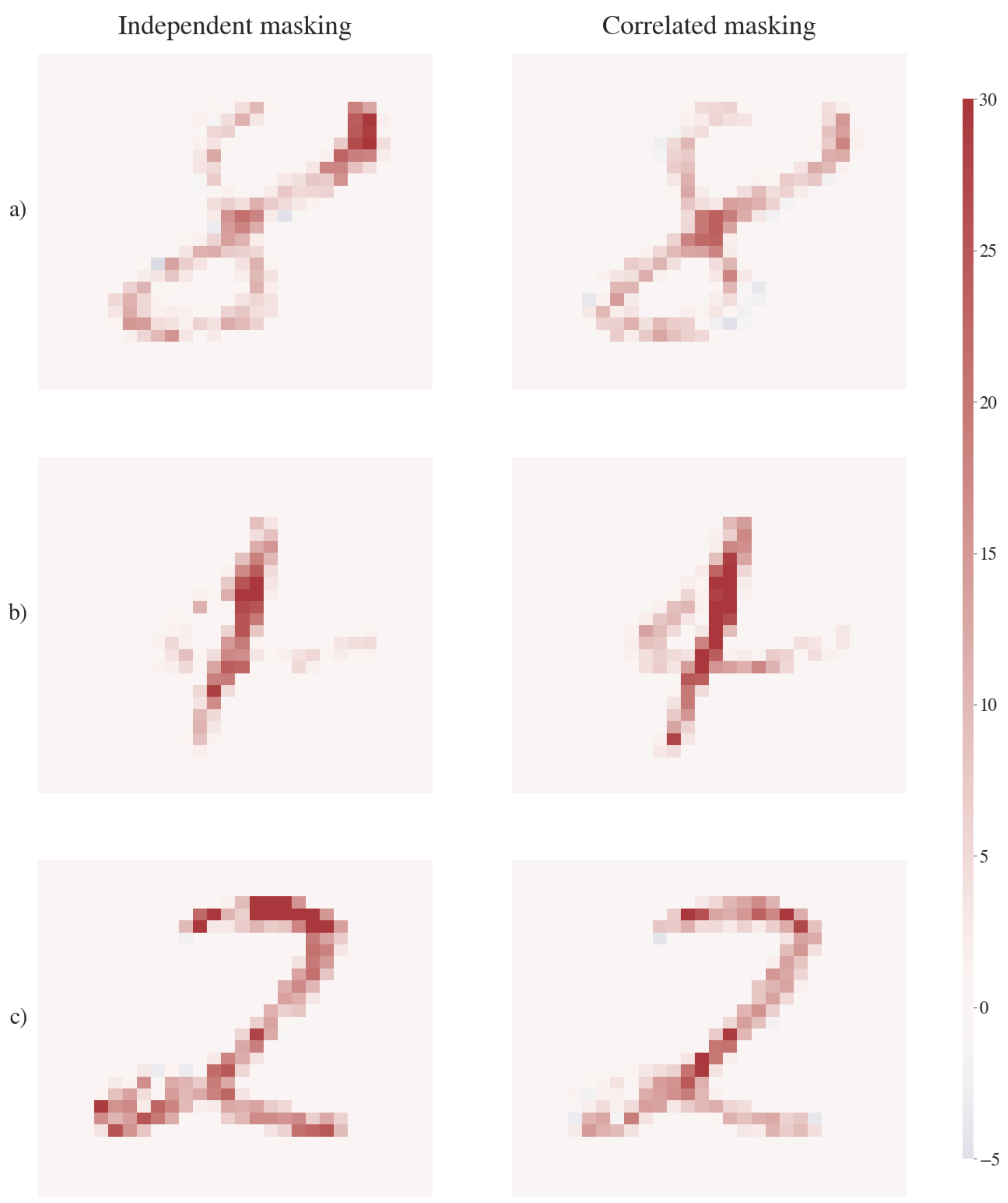

Fig. 4.1 Feature importance scores for three examples from the MNIST dataset (LeCun et al., 2010) using label-free XAI (Crabbé and van der Schaar, 2022) and Integrated Gradients (Sundararajan et al., 2017). a) and b) demonstrate how correlated masking led to smoother importance scores and higher levels of continuity. c) shows how independent masking uses more stylistic detail to build representations.

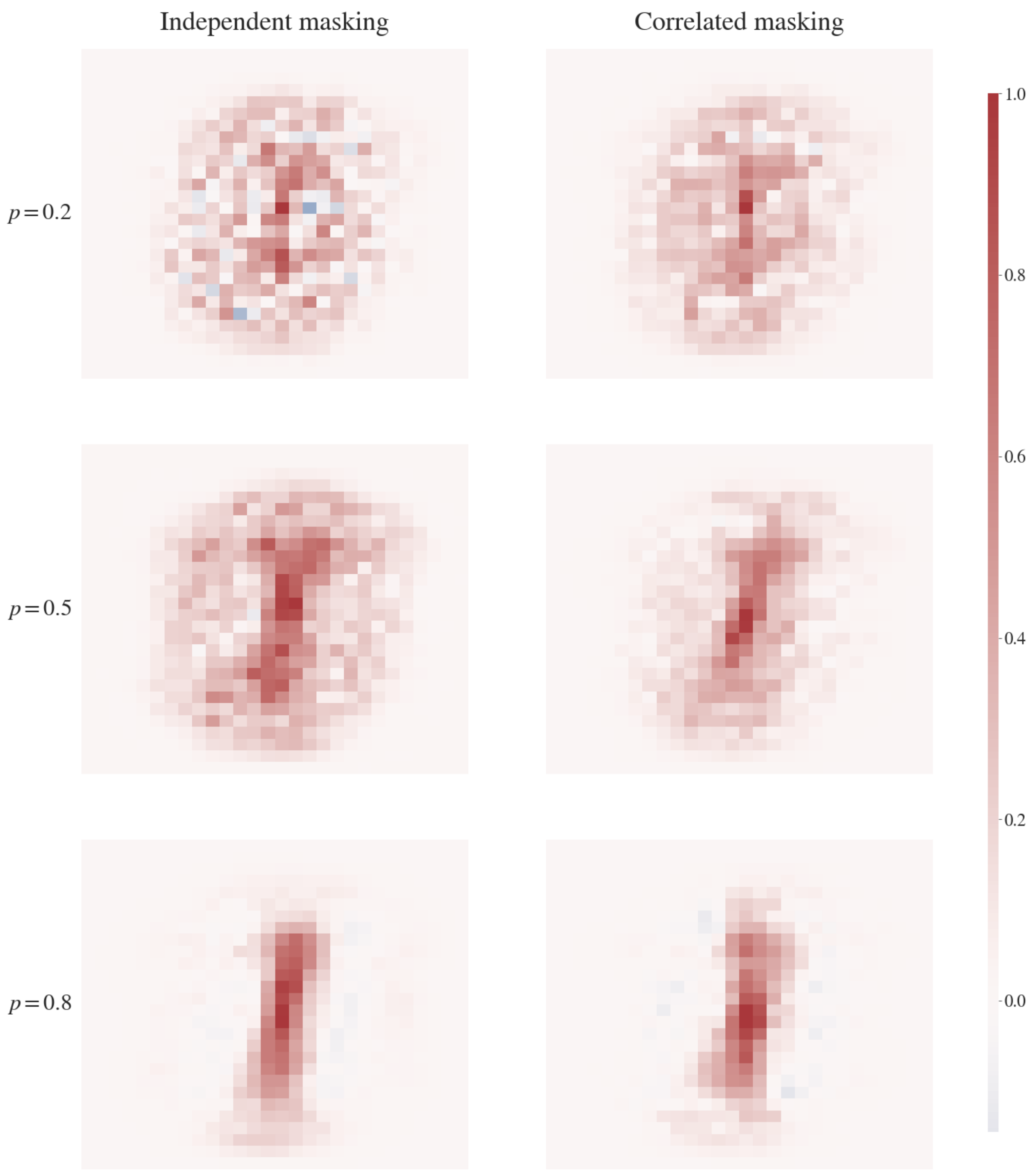

Fig. 4.2 Aggregated feature importance computed on the MNIST dataset (LeCun et al., 2010) using label-free XAI (Crabbé and van der Schaar, 2022) and Integrated Gradients (Sundararajan et al., 2017). For both forms of masking, the breadth and smoothness of the aggregated importance distribution strongly depends on the corruption parameter p, with undesirable behaviour at both extremes.

Figure 4.1 and Figure 4.2 suggest general properties that may characterise ‘better’ representations. In the next section we place this approach on a firmer footing by developing concrete quantitative metrics based on feature importance scores.

4.3 Metrics for label-free optimisation

Visual analysis of representations is a useful tool, but may be time-consuming, sensitive to bias, or difficult in situations where there are only marginal differences. Quantitative metrics go some way in remedying these challenges, and may be amenable to more automatic approaches to optimisation. In this section we propose three such metrics, and validate them on two datasets: MNIST (LeCun et al., 2010) and UCI Blog Feedback (Buza, 2014). We used the experimental setup and architectures given in Section 3.2.1 and Appendix A.2, but varied $p \in \{0.2, 0.3, \ldots, 0.8\}$ and the form of masking (independent or correlated). This required training 16 encoders on each dataset.

The first and second metrics are based on the idea that, in general, we expect good representations to take into account more of the input. This is motivated by Figure 4.2; where we saw that more extreme values of p led to degeneracy and the encoder ignoring large amounts of salient information. While aggregate feature importance is a helpful visual tool, it does not immediately translate into useful single quantity. For example, let us define the *total importance* metric to be the sum over the aggregate importance scores. That is,

$$T(e) = \sum_{i=1}^{d} \mathbf{A}_i. \tag{4.6}$$

Figure A.3 in Appendix A.3 shows that values of p leading to a better spread of importance across the input, as seen in Figure 4.2, do *not* have higher total importance. As p increases, the magnitude of the importance in the collapsed central region far outweighs the reduction in importance elsewhere, leading total importance to not correlate well with downstream performance. A logical alternative would be to compute

$$U(e) = \sum_{i=1}^{d} \mathbb{1}(\mathbf{A}_i > \varepsilon)$$

for some $\varepsilon > 0$, measuring the proportion of variables with importance scores away from zero. However, surprisingly, this also does not work (see Figure A.4 in Appendix A.3) and the reason why unearths an interesting relationship between downstream performance and positive and negative importance scores.

Motivated by this, we define

$$P(e) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\mathbb{1}(b_j(e,\mathbf{x}^{(i)}) > 0) \tag{4.7}$$

and

$$N(e) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\mathbb{1}(b_j(e,\mathbf{x}^{(i)}) < 0) \tag{4.8}$$

to be the *positive spread* and *negative spread* metrics, respectively. Figures 4.4 shows downstream performance is optimal when $P(e)$ is approximately maximised, and when $N(e)$ is approximately minimised. This mirrored behaviour is expected, since

$$d = P(e) + N(e) + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\mathbb{1}(b_j(e,\mathbf{x}^{(i)}) = 0),$$

and so the maximum of $P(e)$ is will clearly coincide with the minimum of $N(e)$. What is interesting, however, is to unpack why positive feature importance scores appears to be preferable, and not the other way around.

The magnitude of an importance score determines the degree of importance, while the *sign* corresponds to the sign of the correlation with the model output. In this case, the model output is

$$g_{\mathbf{x}}(\mathbf{x}, f) = \langle f(\mathbf{x}), f(\mathbf{x})\rangle = ||f(\mathbf{x})||,$$

recalling the definition $b_i(f,\mathbf{x}) = a_i(g_{\mathbf{x}},\mathbf{x})$. Therefore, if $N(\mathbf{A}(e))$ is large, we deduce there are many dimensions of $\mathbf{x}$ *negatively correlated* with $||f(\mathbf{x})||$. Loosely, this is indicative of degenerate behaviour of e, since if these dimensions of $\mathbf{x}$ grow while others remain fixed, the representations of large amounts of potentially very distinct inputs will collapse towards 0. This argument is of course heuristic and *asymptotic*, and just one of many potential explanations. We leave it as an interesting question for future research to determine other rigorous theoretical results that underpin these empirical observations.

Our third metric takes a different approach, and aims to measures the degree of consistency within the feature importance scores. The intuition is that we should expect correlated input dimensions, on average, to have similar importance scores.

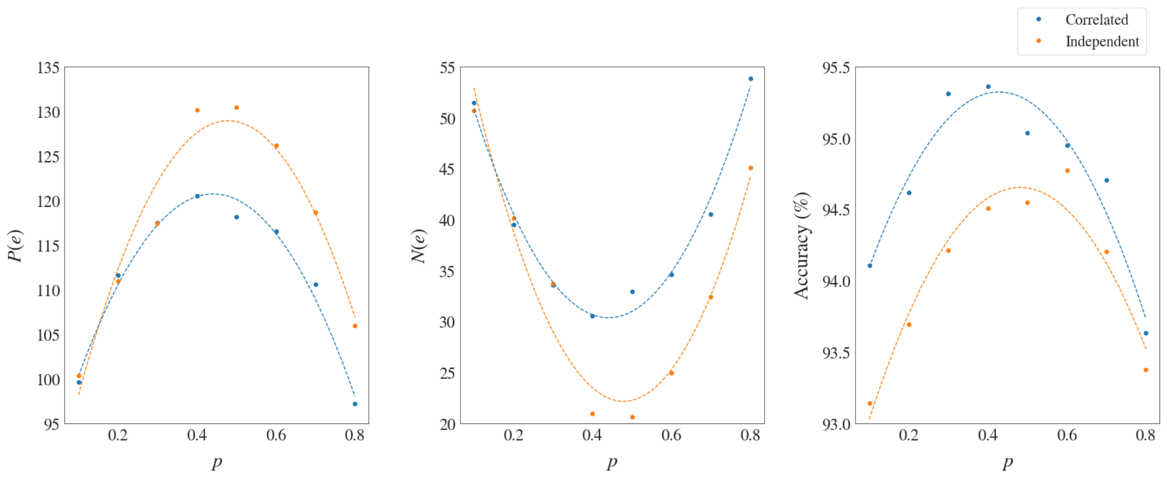

Fig. 4.3 A comparison of the positive and negative spread metrics showing that they are not sufficient to identify correlated masking as superior. This motivates the normalized consistency metric (4.9). The approximate trend lines given were computed using second-order polynomial regression.

We define the *normalized consistency* metric as

$$C(e) = -\frac{1}{T(e)} \sum_{i=1}^{d} \sum_{j=1}^{d} \mathbf{R}_{i,j} |\mathbf{A}_i - \mathbf{A}_j|, \tag{4.9}$$

where $T(e)$ is the total importance (4.6), $\mathbf{A} = \mathbf{A}(e)$ is the aggregated importance (4.5), and $\mathbf{R}$ is the correlation matrix of the input data (as in Section 3.1). Therefore, if $\mathbf{R}_{i,j}$ is large, it enforces larger penalties on deviations between $\mathbf{A}_i$ and $\mathbf{A}_j$. It is important to note that (Crabbé and van der Schaar, 2022, Proposition 2.3), which proves

$$\sum_{i=1}^{d} b_i(f, \mathbf{x}) = ||f(x)|| - b_0$$

for a baseline $b_0 \in \mathbb{R}$, is not sufficient to ensure $\mathbf{A}$ is of a comparable scale *across* models, since b_0 may depend on f. Therefore, the normalization constant $T(e)$ is essential. The positive and and negative spread metrics are *coarse* measures of degeneracy, and are not able to successfully identify correlated masking as superior, see Figure 4.3. However, Figure 4.6, shows that maximising the normalized consistency metric would lead to successfully identifying correlated masking as the stronger method. This is for both MNIST, where we expected this to be the case based on our visualisations in Section 4.2, and UCI Blog Feedback which could be considered a test case. Together, these results show that we should not rely on a single metric for optimisation, but instead take a practical approach that uses a suite of metrics and visualisations to guide design choices.

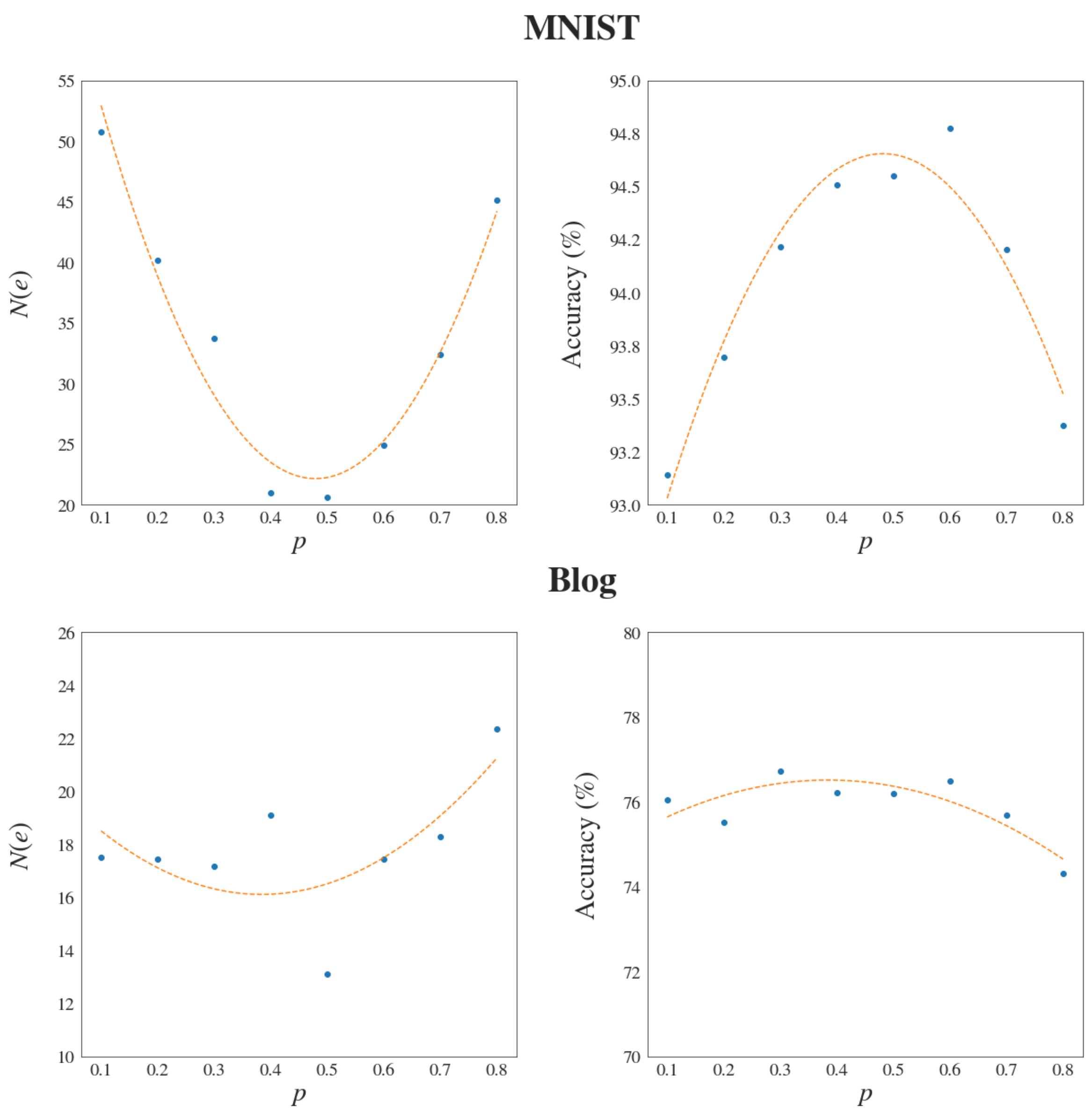

Fig. 4.4 A comparison of the negative spread metric with downstream classification performance. As expected, optimum downstream performance corresponds to the model with approximately minimal $N(\mathbf{A})$. The approximate trend lines given were computed using second-order polynomial regression.

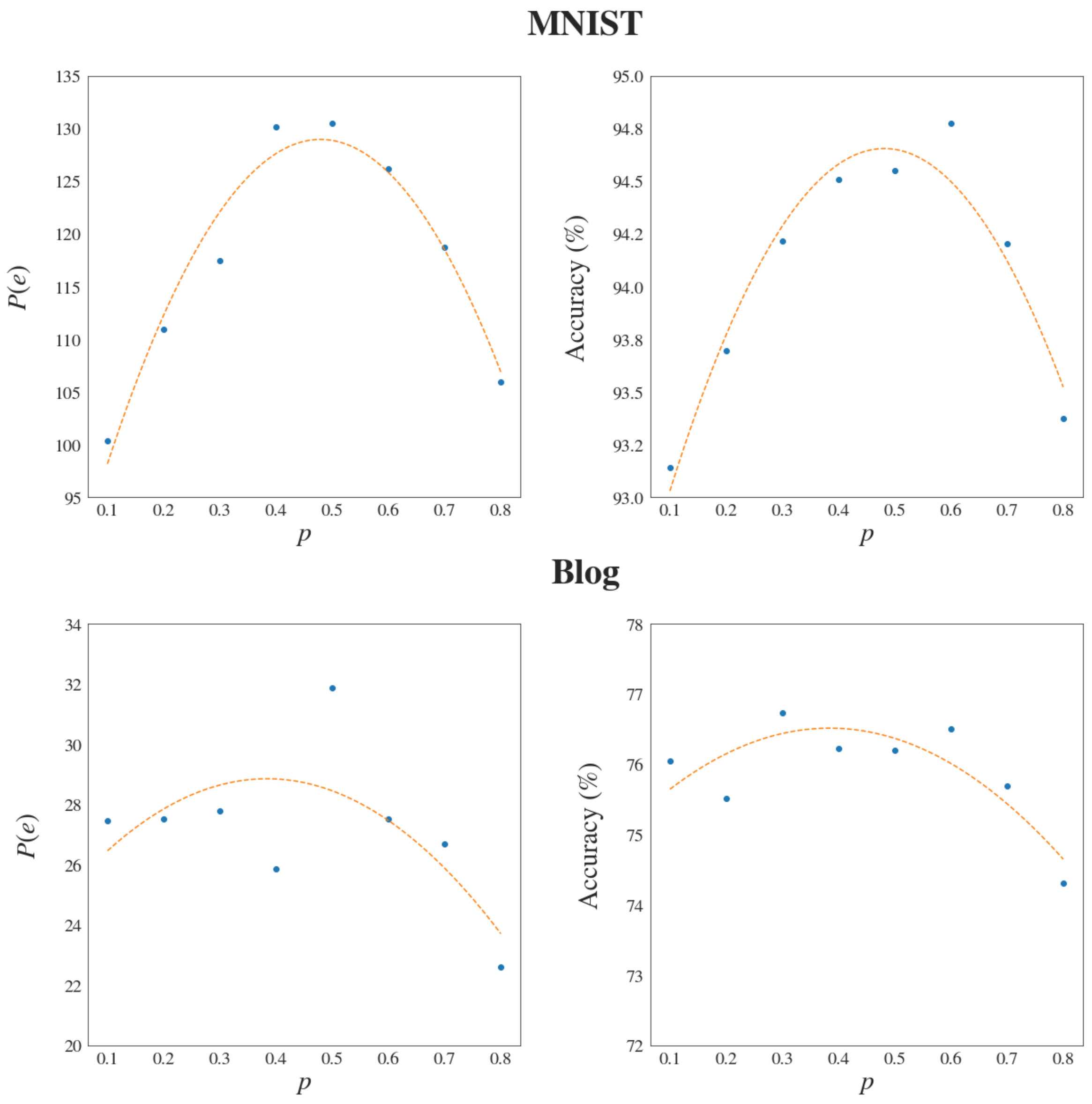

Fig. 4.5 A comparison of the positive spread metric with downstream classification performance. As expected, optimum downstream performance corresponds to the model with approximately maximal $P(\mathbf{A})$. The approximate trend lines given were computed using second-order polynomial regression.

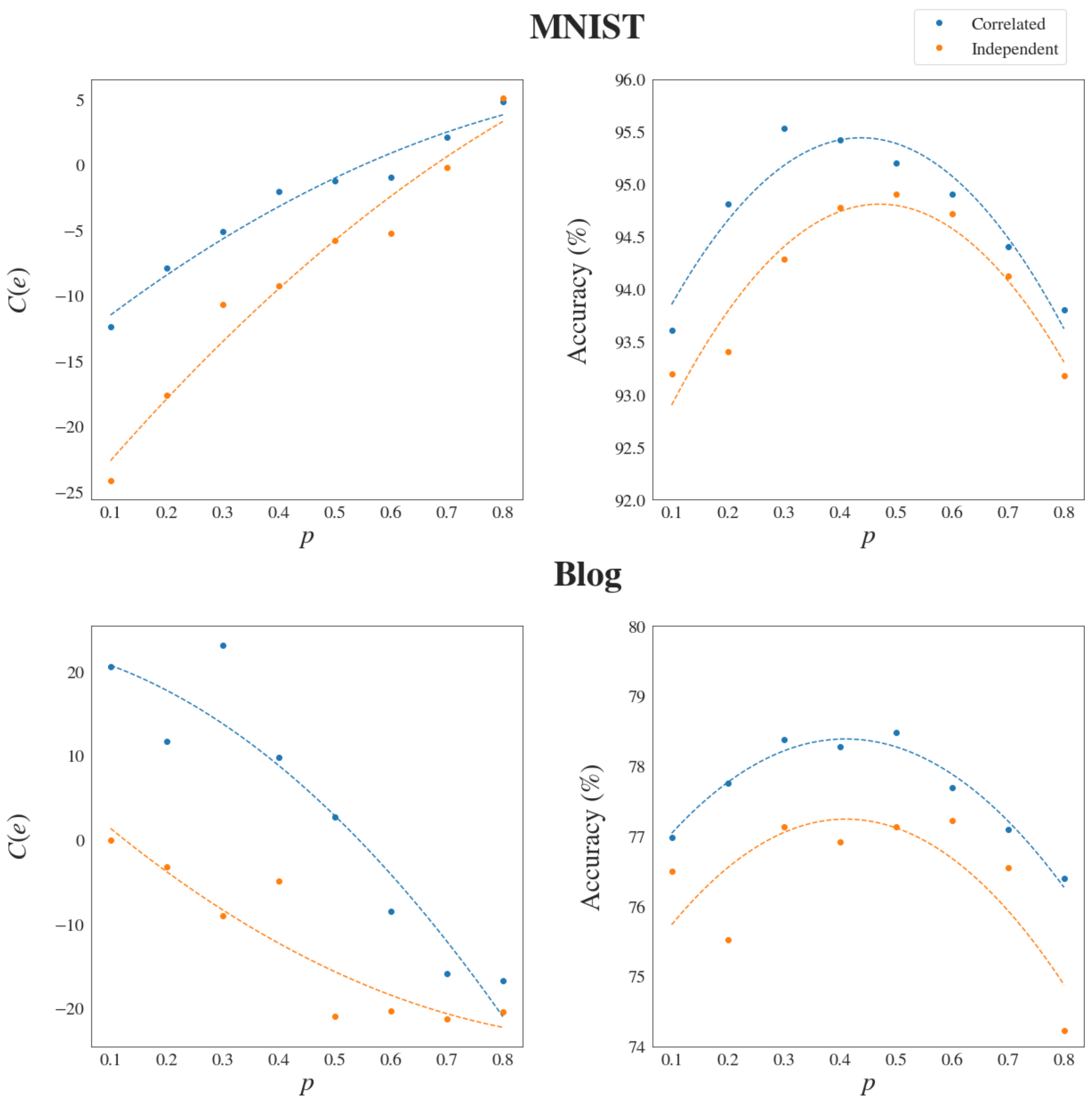

Fig. 4.6 A comparison of the normalized consistency metric with downstream classification performance. This metric, which aims to measure regularity and smoothness in the importance features, successfully identifies correlated masking as the superior model. The approximate trend lines given were computed using second-order polynomial regression.

4.4 Designing encoder ensembles

This chapter has shown that choosing correct difficulties of pretext task is critical. For example, if p is chosen to be too large the encoder may learn to disregard most of the input, see Figure 4.2. Motivated by this, we consider an alternative to trying to choose a single best p, and instead ensemble multiple encoders and concatenate their representations. At the cost of additional compute, this should increase robustness to poor hyperparameter choices, since a downstream model could learn to ignore uninformative parts of the final representation.

We demonstrate that ensembling in this standard way yields a small performance improvement. However, we also introduce a novel *hierarchical* ensemble architecture that allows encoder ensembles to collaborate to solve harder tasks. This approach is shown to perform better, and we briefly discuss how the techniques develop in this chapter explain why.

Let us begin by precisely stating the two ensemble architectures we consider. In a standard *lateral* ensemble, we train a variety of encoders with different hyperparameters independently. Downstream, we aggregate their representations using concatenation. A *hierarchical ensemble* allows certain information to flow through several encoders. As an example, let us describe a hierarchical ensemble that contains three encoders. Each encoder has associated with it a different task. Here, these different tasks will correspond to correlated masking strategies with different values of p (for example $p \in \{0, 2, 0.4, 0.6\}$). For each sample during training, we choose a masking scheme uniformly at random. If $p = 0.2$ is sampled, the input is masked accordingly and propagates through only the first encoder. If $p = 0.4$, then it passes through both the first and the second encoders, and so on. Once an input has passed through the desired number of encoders, a representation is output and passed to projection heads associated with the corresponding encoder to solve a pretext task. Loss is backpropagated as usual.

Harder tasks should be placed higher up the hierarchy, allowing more encoders to collaborate in their solution. The input to encoders above the first level is a copy of the original input *and* the representation from the encoder below it. This architecture is best visualised, see Figure 4.8.

To validate these methods, we considered ensembles with 2, 3 and 4 tasks on the MNIST dataset (LeCun et al., 2010). Each individual encoder followed the setup of Section 3.2.1, and had attached projection heads corresponding to a VIME setup, see Section 2.3. Correlated masking was used throughout.

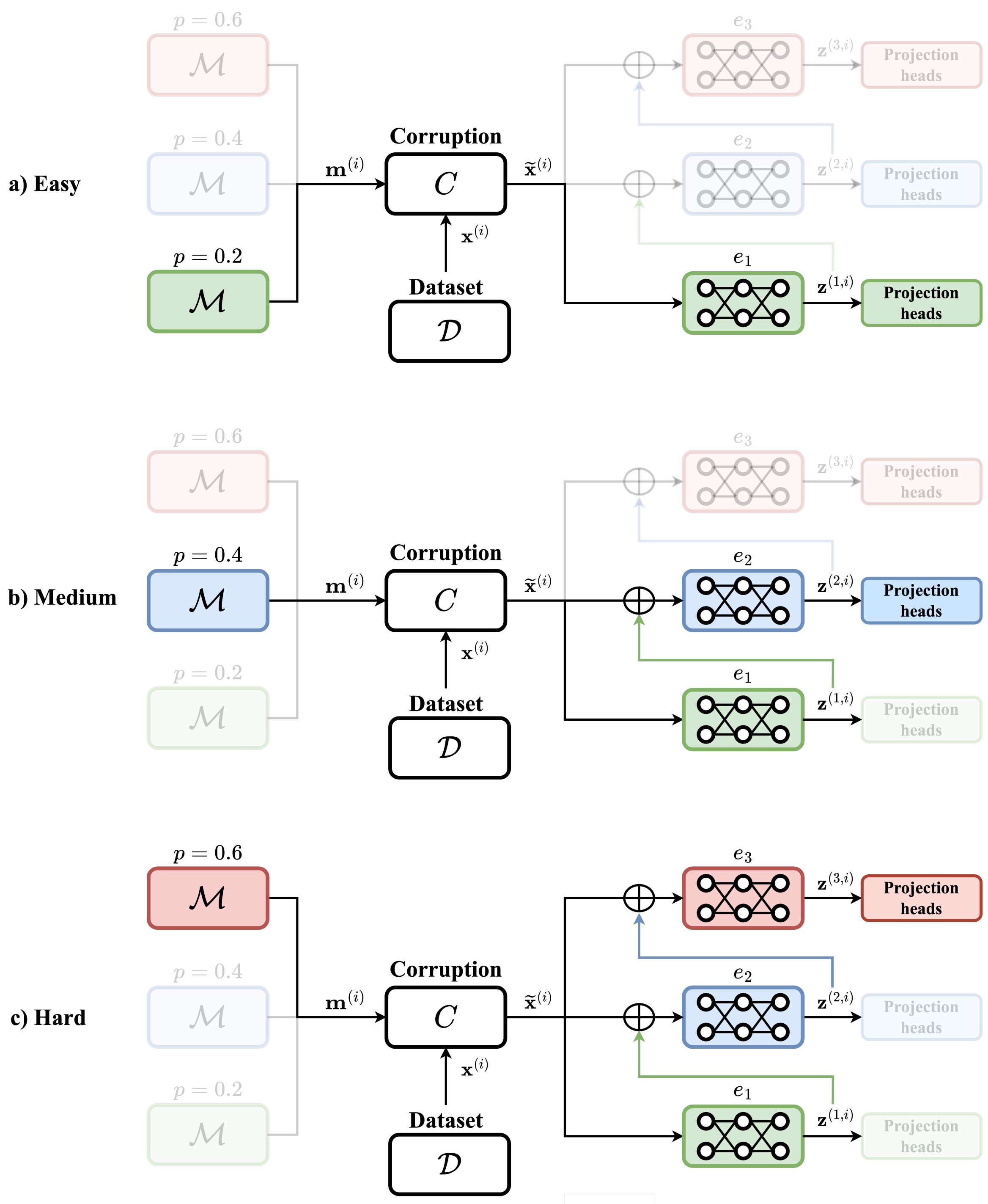

Fig. 4.7 The architecture of a hierarchical ensemble. Data corrupted with more aggressive masking passes through more encoders, and this collaboration helps regularise the base level encoder.

Results

Table 4.1 Downstream accuracy performance on MNIST for 4 VIME models, 3 lateral ensembles and 3 hierarchical ensembles. All ensembles used VIME based projection heads and pretext tasks. Mean and standard deviations are given over ten runs.

Model	**No. Encoders**	p	**Accuracy**
Supervised	–	–	92.9 ± 0.6
VIME	1	0.2	94.6 ± 0.3
	1	0.4	95.4 ± 0.4
	1	0.6	94.9 ± 0.3
	1	0.8	93.6 ± 0.3
Lateral ensemble	2	0.2, 0.4	95.4 ± 0.2
	3	0.2, 0.4, 0.6	95.7 ± 0.3
	4	0.2, 0.4, 0.6, 0.8	95.7 ± 0.3
Hierarchical ensemble	2	0.2, 0.4	96.0 ± 0.2
	3	0.2, 0.4, 0.6	96.1 ± 0.3
	4	0.2, 0.4, 0.6, 0.8	**96.2 ± 0.3**

Discussion

In Table 4.1, we see that even simple lateral ensembles are mildly effective, yielding a around a 0.3% absolute performance gain over the best single encoder. However, hierarchical ensemble improves absolute performance by a further $.5\%$ over lateral, more than doubling the overall improvement over the best single encoder. The techniques in this chapter may be used to suggest why.

Figure 4.8 visualises the aggregated feature importance of the encoder within each of the two ensembles corresponding to $p = 0.2$. In the hierarchical ensemble this the *base encoder*. Using explainability techniques on higher level encoders in the hierarchical case is non-trivial due to the complicated input structure, and we defer this to future work. However, just for the base encoder we see a striking increase in the smoothness, positivity, and magnitude of the aggregated feature importance scores for the hierarchical ensemble. Three properties that were conjectured to be characteristic of good encoders in Sections 4.3 and 4.2. Since most of the performance gain occurs in the hierarchical case when there are just 2 members of the ensemble (Table 4.1), we conjecture this effect on the base level encoder by higher level encoders could be the primary reason hierarchical ensembles perform better. This analysis is supported quantitatively by the three metrics developed in Section 4.3, see Table 4.2.

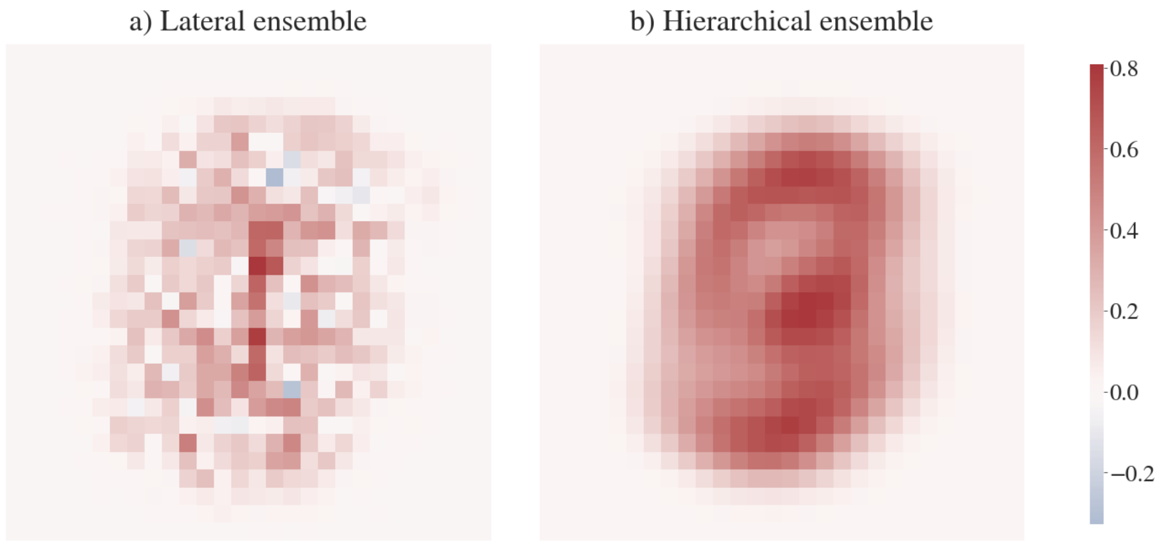

Fig. 4.8 A comparison of the aggregated feature importance computed for the base level encoder within a hierarchical ensemble and the equivalent encoder ($p = 0.2$) within a lateral ensemble.

Table 4.2 A comparison of the positivity, negativity and normalized consistency metrics, denoted $P(e), N(e), C(e)$ respectively, for the base level encoder in a hierarchical ensemble, and the corresponding encoder in a lateral ensemble ($p = 0.2$). Both ensembles used 4 encoders with $p \in \{0.2, 0.4, 0.6, 0.8\}$ and correlated masking. Other experimental parameters were identical to those in Section 3.2.1. Bold indicates superior values.

	$P(e)$	$N(e)$	$C(e)$
Lateral ensemble	111.6	39.5	-60.5
Hierarchical ensemble	**151.1**	**0.0**	**-8.0**

4.5 Summary

This chapter introduced a new program of research that aims to optimise hyperparameters within self-supervised frameworks without access to any labelled data or downstream task. We build upon recent advances in label-free XAI to analyse the intrinsic structure of representations. An exploratory analysis is given, and identifies a number of general properties representations may benefit from, such as a smooth distribution of aggregated feature importance scores. Three metrics are introduced to target these properties, and we show that optimising these metrics correlates strongly with *a-priori* known best hyperparameters. Future work should validate these techniques more widely, and expand the suite of available metrics. We hope that eventually machine learning practitioners training self-supervised models will be able to draw on a host of quantitative and qualitative tools like these to guide design choices during model development.

Chapter 5

Conclusions

This thesis has contributed to the development and understanding of masking strategies in tabular self-supervision. In this final chapter, we bring together our findings into a single narrative that allows us to clearly motivate directions for further research. We begin with a summary, before presenting a few concrete extensions and some more ambitious research questions.

5.1 Summary

In Chapter 2, we gave a conceptual overview of self-supervised learning and set up a unified notation for describing a wide variety of existing techniques. This highlighted the different components in the pipeline, and that most existing innovations have focused on developing new architectures, loss functions and pretext tasks. Even in this latter case, corruption based pretext task design has focused primarily on how to corrupt input, and not *what part* of the input to corrupt.

In Chapter 3, we saw how masking informed by the correlation structure of the input is a generally applicable technique that may enhance several state-of-the-art methods for tabular self-supervision. We construct a family of synthetic datasets to identify the failure mode of independent masking that correlated masking addresses. We then validate correlated masking across standard benchmarks. To conclude, we introduce *combination masking*, a procedure that allows us to combine multiple masking schemes to benefit from the advantages of each. This also gives finer control over task difficulty, and allows a balance between learning local and global information. We show this method outperforms the alternatives for a family of 7 proteomics datasets on the efficacy of cancer treatments.

After establishing the benefit of correlated masking, Chapter 4 asks *why* it is better, and how we may optimise hyperparameters during self-supervision without access to labelled data. This is a challenging problem but, using recent advances in label-free explainability (Crabbé and van der Schaar, 2022), we explore how distributions of feature importance scores vary across different encoders. For optimisation, we convert these qualitative insights into a collection of metrics. Two of these metrics are empirically shown to offer useful insight into issues that arise if pretext tasks become too difficult or easy. A third metric is more subtle, and measures consistency between the feature importance scores of correlated variables. This gives an abstract indicator of smoothness, and measures the tendency of the encoder to overfit to small details.

A key message of Chapter 4 is that tabular self-supervision may easily degenerate if care is not given when choosing parameters. We conclude with an application that aims to make the framework more robust by ensembling encoders and aggregating their representations. Standard ensembling like this leads to performance gains, but to improve it further, we introduce hierarchical ensembles that pass information between ensemble components to collaborate on harder tasks. Using explainability techniques, we show this acts as a form of regularisation on the base encoder and dramatically improves several of our associated metrics.

5.2 Future directions

The most immediate extension to our work is to validate the techniques we propose across even more datasets and methods, with the hope of increasing our understanding of their strengths and failure modes. However, our investigations also open the door to many new questions, and in this final section we present two that we feel are most promising.

5.2.1 From correlation to causation

Correlation leads to an increase in task difficulty and prevents trivial correlations obfuscating the learning of deeper semantic structure. However, masking in this way may lead to pretext tasks that are inconsistent from a causal perspective. A causal graph, also known as a directed acyclic graph (DAG), describes the functional dependencies between a set

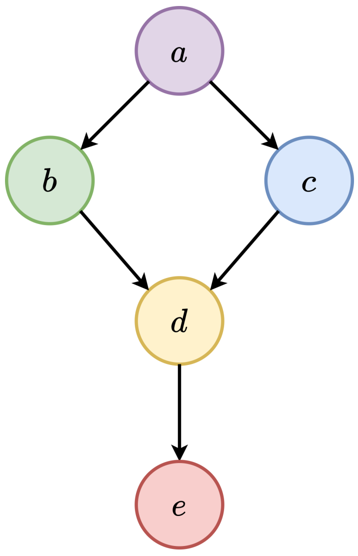

Fig. 5.1 An example of a causal directed acyclic graph (DAG) between variables a, b, c, d and e. Each variable is considered to be causally dependent on variables from which there exists an *incoming* edge. For example, b is causally dependent on a but not c.

of variables. See Figure 5.1. In this example, it would be unreasonable to task a network with predicting b from d and e, since this contradicts the causal structure, and we risk the network overfitting to particular examples in the training data in response. Whether this impacts overall representation quality will depend on the causal structure of the data, and the probability of such inconsistent pretext tasks being generated from a masking scheme in question. To begin to answer these questions, we must obtain (or approximate) the DAG. Fortunately, recent work allows a DAG to be learnt easily as a pre-processing step using a continuous optimisation process (Zheng et al., 2020). This has already been successfully applied to regularise neural networks (Kyono et al., 2020, CASTLE)), and improve imputation (Kyono et al., 2021, MIRACLE). Future work could incorporate this DAG within a masking scheme to modulate and control the nature of questions being asked.

We expect this could play a particularly important role for modelling temporal data, where existing approaches such as Temporal Neighborhood Coding (Tonekaboni et al., 2021), Neighborhod Contrastive Learning (Yèche et al., 2021) and Bootstrap Your Own Positive Sample BYOP (Wanyan et al., 2021) (not to be confused with BYOL (Grill et al., 2020)) neglect a deep modelling of causal structure. Incorporating this causal structure within pretext tasks could implicitly cause representations to encode functional dependencies that are highly sensitive to temporal relationships, and prevent the network learning spurious patterns that are not causally consistent with how events are organised with respect to an *arrow of time*.

5.2.2 Dynamic masking schemes

Should masking a scheme remain constant or evolve throughout training? In psychology, the idea of spaced repetition argues human learning benefits from seeing tasks frequently in the beginning, and then less frequently over time to revise and consolidate information (Averell and Heathcote, 2011; Cepeda et al., 2006). This idea has seen applications in natural language processing (Amiri et al., 2017), and curriculum learning (Bengio et al., 2009; Kumar et al., 2010; Soviany et al., 2022), which argues networks may benefit from task difficulty increasing over time. In our setting, this could motivate masking schemes that gradually become more difficult. This could be achieved, for example, by placing a distribution over p whose mean is a monotonically increasing function of time.

Eventually, however, this would inevitably lead to prohibitive task difficulty. This naturally asks whether the masking scheme could adapt in real-time, based on feedback. Validation loss is not a good indicator of downstream performance, but perhaps techniques building on metrics such as those in Chapter 4 could be measured during training to provide feedback and continuously fine-tune a masking scheme. This concept could be pushed further, viewing the overall model architecture and method as a hyperparameter. AutoML frameworks that automatically search over hyperparameters and neural architectures have surged in popularity in recent years (He et al., 2021), and it could be possible to design a bespoke framework for use in tabular self-supervision. Despite the inherent non-differentiability of the metrics we introduce in Chapter 4, their signal could be incorporated within automatic optimisation schemes based on evolutionary algorithms (Li et al., 2022) or reinforcement learning (Zoph and Le, 2017). The potential for this will only grow as research uncovers even more metrics that characterise what makes a good representation.

References

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Amiri, H., Miller, T., and Savova, G. (2017). Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410, Copenhagen, Denmark. Association for Computational Linguistics.

Averell, L. and Heathcote, A. (2011). The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55(1):25–35. Special Issue on Hierarchical Bayesian Models.

Bahri, D., Jiang, H., Tay, Y., and Metzler, D. (2022). Scarf: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Barretina, J., Caponigro, G., Stransky, N., and et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483:603–607.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. (2021). Deep neural networks and tabular data: A survey.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Buza, K. (2014). BlogFeedback. UCI Machine Learning Repository.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA. Association for Computing Machinery.

Cepeda, N., Pashler, H., Vul, E., Wixted, J. T., and Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132 3:354–80.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Crabbé, J. and Van Der Schaar, M. (2021). Explaining time series predictions with dynamic masks. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2166–2177. PMLR.

Crabbé, J. and van der Schaar, M. (2022). Label-free explainability for unsupervised models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4391–4420. PMLR.

Das, A. and Rad, P. (2020). Opportunities and challenges in explainable artificial intelligence (xai): A survey.

Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *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.

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. In *International Conference on Learning Representations*.

Elbers, R. (2021). Infonce pytorch. https://github.com/RElbers/info-nce-pytorch.

Fang, P.-F., Li, X., Yan, Y., Zhang, S., Kang, Q.-Y., Li, X.-F., and Lan, Z.-Z. (2022). Connecting the dots in self-supervised learning: A brief survey for beginners. *J. Comput. Sci. Technol.*, 37(3):507–526.

Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457.

Gao, T., Bhattacharjya, D., Nelson, E., Liu, M., and Yu, Y. (2022). IDYNO: Learning nonparametric DAGs from interventional dynamic data. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6988–7001. PMLR.

Genest, C., Gendron, M., and Bourdeau-Brien, M. (2009). The advent of copulas in finance. *The European Journal of Finance*, 15(7-8):609–618.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*.

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. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, X., Zhao, K., and Chu, X. (2021). Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90.

Jing, L. and Tian, Y. (2021). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058.

Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Joe, H. (2014). *Dependence Modeling with Copulas*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press.

Kaya, M. and Bilge, H. (2019). Deep metric learning: A survey. *Symmetry*, 11(9).

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kohavi, R. and Becker, B. (1996). Census Income. UCI Machine Learning Repository.

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.

Kumar, M., Packer, B., and Koller, D. (2010). Self-paced learning for latent variable models. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Kyono, T., Zhang, Y., Bellot, A., and van der Schaar, M. (2021). Miracle: Causally-aware imputation via learning missing data mechanisms. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23806–23817. Curran Associates, Inc.

Kyono, T., Zhang, Y., and van der Schaar, M. (2020). Castle: Regularization via auxiliary causal graph discovery. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1501–1512. Curran Associates, Inc.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

LeCun, Y. (2022). A path towards autonomous machine intelligence.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature Cell Biology*, 521(7553):436–444.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, C., Imrie, F., and van der Schaar, M. (2022). Self-supervision enhanced feature selection with correlated gates. In *International Conference on Learning Representations*.

Letham, B., Rudin, C., McCormick, T. H., and Madigan, D. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *CoRR*, abs/1511.01644.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, N., Ma, L., Yu, G., Xue, B., Zhang, M., and Jin, Y. (2022). Survey on evolutionary deep learning: Principles, algorithms, applications and open issues.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman Hall / CRC, London.

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., and Roth, D. (2021a). Recent advances in natural language processing via large pre-trained language models: A survey.

Min, B., Ross, H. H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., and Roth, D. (2021b). Recent advances in natural language processing via large pre-trained language models: A survey. *ArXiv*, abs/2111.01243.

Musgrave, K., Belongie, S., and Lim, S.-N. (2020). PyTorch Metric Learning.

Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer, New York, NY, USA, second edition.

Noroozi, M. and Favaro, P. (2016). Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 69–84, Cham. Springer International Publishing.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544.

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

Pise, N. N. and Kulkarni, P. (2008). A survey of semi-supervised learning methods. In *2008 International Conference on Computational Intelligence and Security*, volume 2, pages 30–34.

Python Core Team (2019). *Python: A dynamic, open source programming language*. Python Software Foundation. Python version 3.8.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Roh, Y., Heo, G., and Whang, S. E. (2021). A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. (2022). Curriculum learning: A survey. *Int. J. Comput. Vision*, 130(6):1526–1565.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 776–794, Cham. Springer International Publishing.

Tomczak, K., Czerwińska, P., and Wiznerowicz, M. (2015). The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary Oncology*, 19:A68 – A77.

Tonekaboni, S., Eytan, D., and Goldenberg, A. (2021). Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *International Conference on Learning Representations*.

Ucar, T., Hajiramezanali, E., and Edwards, L. (2021). Subtab: Subsetting features of tabular data for self-supervised representation learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Ustun, B., R.-C. (2016). Supersparse linear integer models for optimized medical scoring systems. *Mach. Learn.*, 102:349.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., 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.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML ’08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*.

Wang, F. and Liu, H. (2021). Understanding the behaviour of contrastive loss. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504, Los Alamitos, CA, USA. IEEE Computer Society.

Wanyan, T., Zhang, J., Ding, Y., Azad, A., Wang, Z., and Glicksberg, B. S. (2021). Bootstrapping Your Own Positive Sample: Contrastive Learning With Electronic Health Record Data.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Weinberger, K. Q., Blitzer, J., and Saul, L. (2005). Distance metric learning for large margin nearest neighbor classification. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, Los Alamitos, CA, USA. IEEE Computer Society.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

Yoon, J., Zhang, Y., Jordon, J., and van der Schaar, M. (2020). Vime: Extending the success of self- and semi-supervised learning to tabular domain. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11033–11043. Curran Associates, Inc.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022). Coca: Contrastive captioners are image-text foundation models.

Yèche, H., Dresdner, G., Locatello, F., Hüser, M., and Rätsch, G. (2021). Neighborhood Contrastive Learning Applied to Online Patient Monitoring. In *Proceedings of 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 11964–11974. PMLR.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 649–666, Cham. Springer International Publishing.

Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. (2020). Learning sparse nonparametric dags. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3414–3425. PMLR.

Zoph, B. and Le, Q. (2017). Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*.

Appendix A

Supporting material

A.1 SubTab

In this appendix we give a complete mathematical description of SubTab (Ucar et al., 2021).

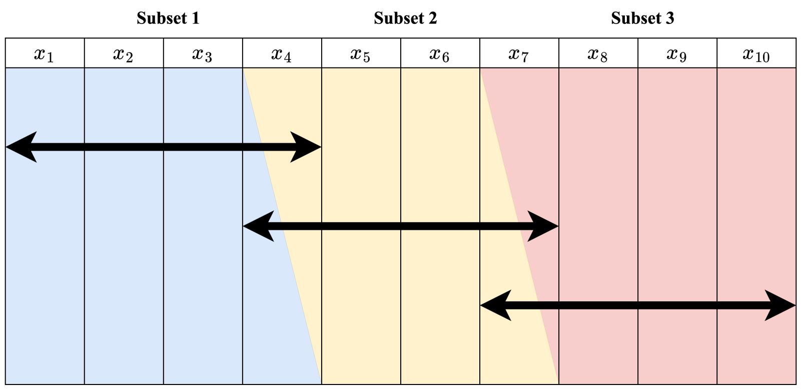

Fig. A.1 An illustration of the process to generate subset views in SubTab (Ucar et al., 2021). In this example $K = 3$, and the proportion of overlap between neighboring subsets is $r = 0.25$.

The SubTab model uses two projection heads h_1 and h_2, and a mask matrix $\mathbf{M}$ is formed by sampling rows as in the masking procedure described for VIME (see Section 2.3.2). The corruption function C contains two parts, that is $C = B \circ A$. For a batch $\mathbf{X} = \{\mathbf{x}^{(i)} : i = 1, \ldots, N\}$, A is defined by

$$A(\mathbf{M}, \mathbf{X}) = (1 - \mathbf{M}) \circ \mathbf{X} + \mathbf{M} \circ \mathbf{Z} = \widetilde{\mathbf{X}}.$$

where $\mathbf{Z}$ is either equal to a zero matrix, a matrix whose elements $Z_{i,j}$ are draws from the empirical distribution of the j^{th} dimension of $\mathbf{X}$, or additive Gaussian white noise. B then divides $\widetilde{X}$ into multiple subsets, which we denote

$$B(A(\mathbf{M}, \mathbf{X})) = B(\widetilde{\mathbf{X}}) = \{\widetilde{\mathbf{X}}^{(1)}, \dots, \widetilde{\mathbf{X}}^{(K)}\}.$$

These are obtained by dividing $\widetilde{\mathbf{X}}$ into K matrices each containing approximately

$$\left\lfloor \frac{(1+2r)N}{K} \right\rfloor \tag{A.1}$$

consecutive columns from $\mathbf{X}$, where $r \geq 0$ determines the proportion of overlap between subsets. Figure A.1 illustrates this process.

Following the notation in Section 2.3, we denote the output of $\widetilde{\mathbf{X}}^{(a)}$ from h_k by $\mathbf{Y}^{(k,a)}$. The outputs $\mathbf{Y}^{(1,a)}$ are passed to a standard MSE reconstruction loss $\mathcal{L}_1$. For the outputs $\mathbf{Y}^{(2,a)}$, define $S = \{(\mathbf{Y}^{(2,a)}, \mathbf{Y}^{(2,b)}) : a < b\}$ to be the set of K choose 2 distinct pairs of subsets, and compute

$$\mathcal{L}_C = \frac{1}{K(K-1)} \sum_{(a,b)\in S} l(\mathbf{Y}^{(2,a)}, \mathbf{Y}^{(2,b)}) + l(\mathbf{Y}^{(2,b)}, \mathbf{Y}^{(2,a)})$$

where

$$l(\mathbf{Y}^{(2,a)}, \mathbf{Y}^{(2,b)}) = -\frac{1}{N} \sum_{i=1}^{N} \log\left(\frac{\exp(\text{sim}(\widetilde{\mathbf{Y}}_i^{(2,a)}, \mathbf{Y}_i^{(2,b)})/\tau)}{\frac{1}{N}\sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp(\text{sim}(\widetilde{\mathbf{Y}}_i^{(2,a)}, \mathbf{Y}_j^{(2,b)})/\tau)} \right).$$

In addition, subsets from the same row are encouraged to have closer representations with an additional MSE distance loss given by

$$\mathcal{L}_D = \frac{2}{NK(K-1)} \sum_{(a,b)\in S} \sum_{i=1}^{N} \sum_{j=1}^{d} (\mathbf{Y}_{i,j}^{(2,a)} - \mathbf{Y}_{i,j}^{(2,b)})^2.$$

The loss for h_2 is then $\mathcal{L}_2 = \mathcal{L}_C + \mathcal{L}_D$, while the total loss is $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$.

A.2 Experimental details

In this appendix we provide further details, that may be helpful when reproducing our results. Table A.1 specifies the exact number of training epochs used for each dataset. For proteomics examples, we determined a maximum number epochs that was sufficient for convergence across all drugs. In no cases was overtraining during self-supervision found to detriment performance. Table A.2 specifies default hyperparameters used for various methods. These were chosen to be typical moderate values, and drawn from the original papers when possible (as in SubTab (Ucar et al., 2021)).

Table A.1 The number of epochs used for self-supervision when training on various datasets and methods. These numbers were conservatively determined to be sufficient by monitoring for convergence of an unlabelled validation loss.

	Dataset				
Model	**MNIST**	**Blog**	**Income**	**Synthetic**	**Proteomics**
VIME	300	500	200	200	300
SCARF	250	500	200	200	–
SubTab	400	500	200	200	–
DAE	300	500	200	200	–
Context Encoders	300	500	200	200	–

Table A.2 Default hyperparameter choices for various self-supervised methods. β is the weighting of cross-entropy loss and τ an InfoNCE contrastive loss temperature parameter. In SubTab, K is the number of subsets used and r the proportion of overlap, and our choices are based on (Ucar et al., 2021).

Model	**Defaut hyperparameters**
VIME	$\alpha = 2, \beta = 0.1$
SCARF	$\tau = 0.1$
SubTab	$K = 4, r = 0.75$ (MNIST), $K = 7, r = 0.75$ (Blog), $K = 5, r = 0.25$ (Income), $K = 3, r = 0.25$ (Synthetic).
DAE	$\beta = 0.1$
Context Encoders	$\beta = 0.1$

A.3 Additional visualisations

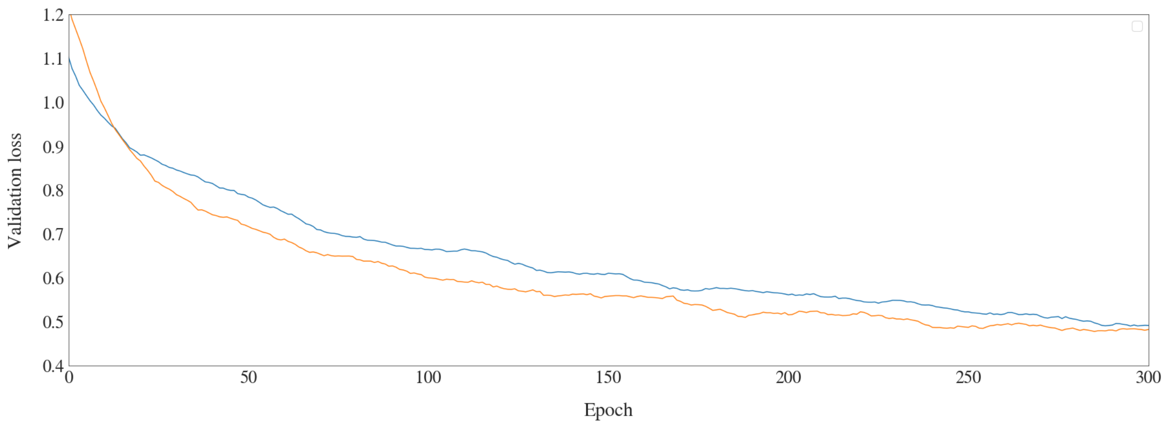

Fig. A.2 A comparison of validation and training losses for SCARF on UCI Blog. Values were smoothed by taking averages over a sliding window of size 20.

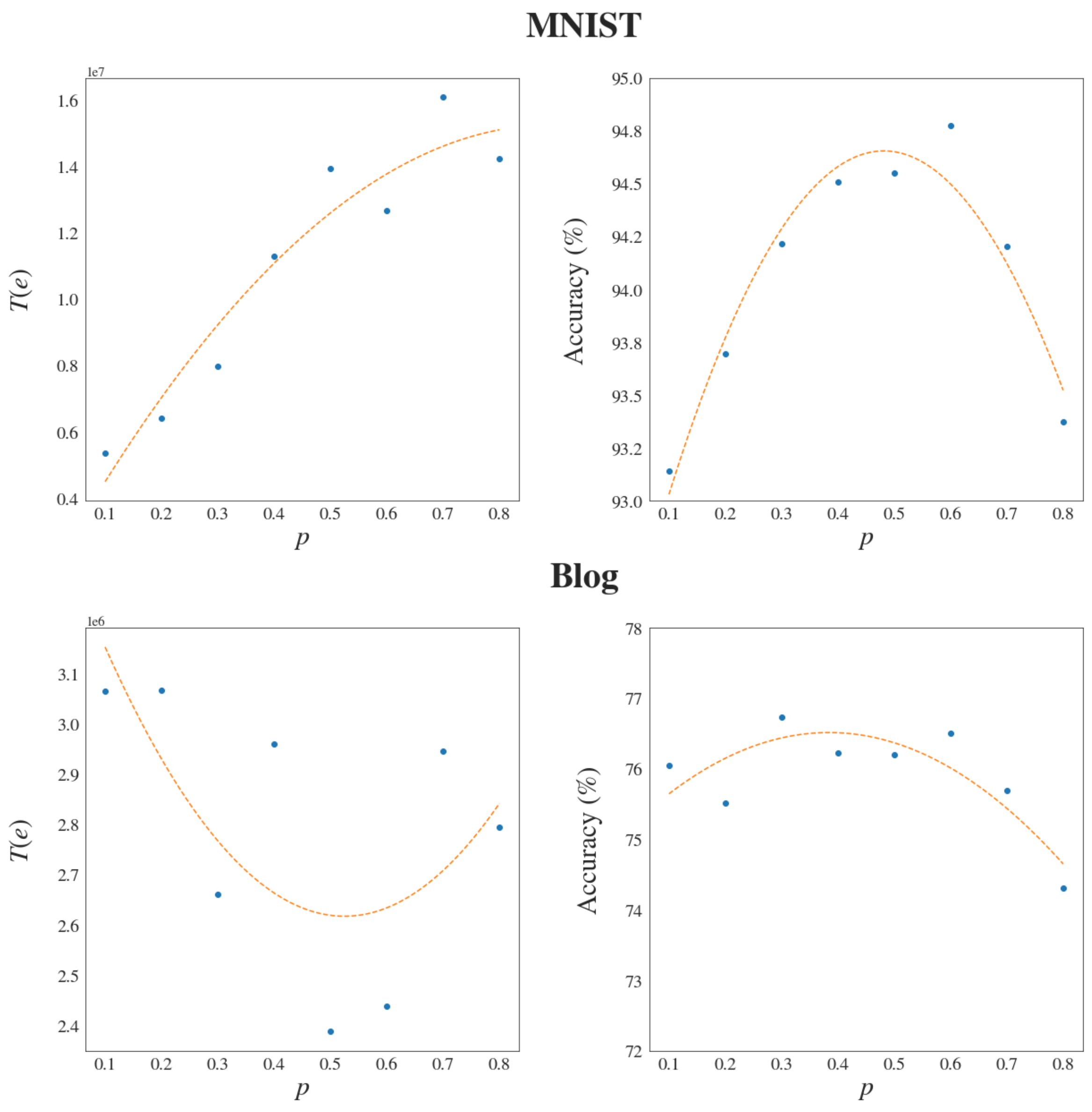

Fig. A.3 The total importance (sum over the aggregate importance vector $\mathbf{A}$) for a variety of models. This metric does not correlate well with downstream performance, due to rapidly growing feature importance scores in the central region as p increases, that dwarf reductions elsewhere.

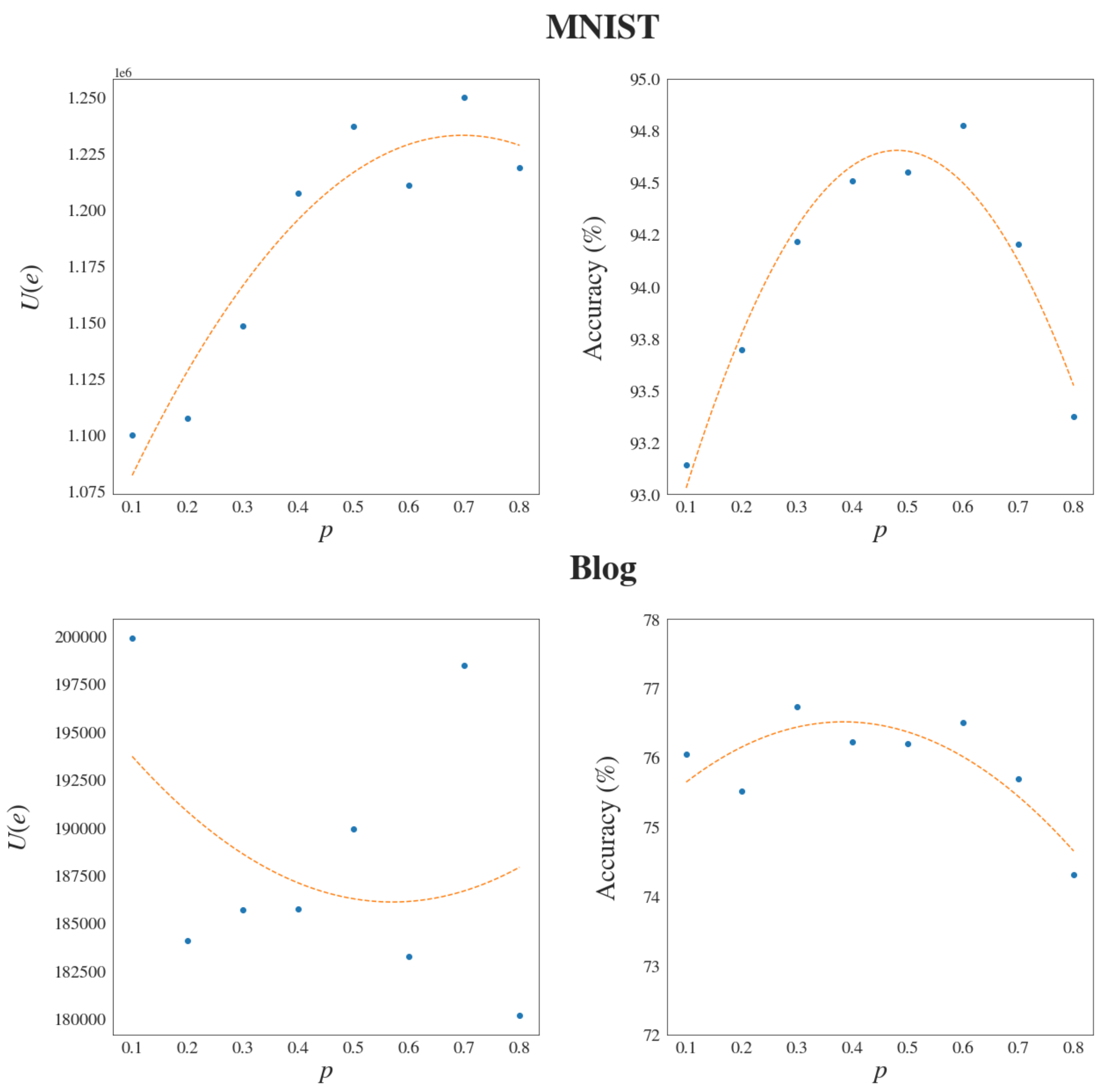

Fig. A.4 The quantity $U(e)$ with a threshold $\varepsilon = 1$. This metric measures aggregated feature importance values with magnitudes greater than ε. It does not correlate well with downstream performance, due to neglecting importance differences between positive and negative feature importance scores.