Comparison of Deep Learning mutual information estimators for Neuroscience

Laura Milad

Master of Science

Data Science

School of Informatics

University of Edinburgh

2023

Abstract

The advancements of data acquisition technologies, have resulted in neuroscience experiments generating increasingly complex and multivariate datasets for which the assessment of interactions between numerous variables is required. Mutual information (MI) is a fundamental quantity for capturing the true dependence between variables, making it a valuable tool for analyzing such data. However, MI is difficult to compute directly from neuroscience recordings and very few approaches for calculating MI can scale up to the size and dimensionality encountered in modern problems. To address this issue, variational objectives utilising deep neural networks have emerged as promising MI estimators, yet their performance on neuroscience datasets remains unclear. In this work, we evaluate and compare three deep learning MI estimators - MINE, InfoNCE and FLO - across simple Gaussian and simulated neuroscience datasets. Despite the poor performance of InfoNCE in the simple Gaussian settings, it demonstrates superior performance in the simulated neuroscience framework, making it the optimal choice for calculating MI in real neuroscience data.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 685901
Date when approval was obtained: 2023-05-26

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Laura Milad*)

Acknowledgements

Words cannot express my gratitude to my supervisor for his valuable support and feedback. I deeply appreciate his efforts to explain and guide me towards the right direction so I can make the most of this project.

I am also very grateful to my family and friends for their constant support and encouragement. Very few and special people have always been by my side through all the ups and downs and i am very lucky to have them. Also, my Edinburgh friends offered me an opportunity to unwind and have a good time in between my studies. This challenging journey would be ten times harder without them.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

The brain is considered the most complex structure in existence [17]. It performs numerous different signal processing interactions, from gene networks that regulate the cell functions to neural circuits that control behavioural cues [58]. These signal processing interactions can be considered as information exchange mechanisms. The neural system acquires information in the form of sensory input, process it and adjusts its state in accordance with the change in the environment [16]. Neuroscience, the dedicated science for studying the nervous system's structure and functions, seeks to uncover these intricate mechanisms.

The brain showcases diverse functionalities that require specialized approaches for data acquisition. However, despite the diversity of these functionalities, like gene networks and neural circuits, in all cases the assessment of interactions among numerous variables is required. Due to advancements in data collection methods and computing technologies, our insights into neural processing have become more complex [29, 51]. Neuroscience experiments now generate progressively multivariate data, with simultaneous recordings of multiple neurons and can incorporate data of different types [58]. For instance, an in vivo calcium imaging experiment, which involves stimuli and behaviour, produces a dataset comprising at least three distinct types: behavioural, physiological, and stimulation data [50]. Lastly, neuroscience data is often characterized by noise and exhibits nonlinear relationships among variables, further complicating the task of capturing the underlying relationships within these experiments [58].

Mutual information (MI), an important concept in information theory, serves as a highly valuable tool for analysing such complex data as it possesses the capability

to detect both linear and non-linear relationships across multi-dimensional scenarios [59]. MI plays a fundamental role in quantifying the true dependence between variables [31]. In essence, it quantifies how much knowing the outcome of one random variable reduces the uncertainty about the outcome of another random variable. Information theory including MI finds extensive application in various fields, like neuroscience [4, 13, 46, 48, 60]. John von Neumann, for example, has emphasized through his work that information theory is essential for understanding the functionalities of the brain [38]. The use of MI helps gaining insight into the functioning of the neural system and more specifically, it reveals the ability of the system to achieve its remarkable information processing abilities [35].

However, in spite of its usefulness, it is difficult to calculate MI directly from neural recordings especially when dealing with extensive neuroscience datasets [36]. In order to calculate any information theoretic measure, precise knowledge of the joint probability distribution between the states of the stimuli and the neural population is required [59]. However, the estimation of this distribution from recordings is notoriously difficult [31, 46, 47]. Even though many parametric and non-parametric techniques [11, 18, 32] have been proposed for estimating MI, most of them work well for low-dimensional data and are not capable to scale up to the size and complexity of modern datasets.

In order to address the challenges posed by the increasing size and complexity of modern problems, variational objectives have gained significant popularity in recent years for scaling MI estimation. Variational approaches leverage mathematical inequalities to create manageable lower or upper bounds of the mutual information [44]. These bounds are parameterized by deep neural networks (DNN) [8] and achieve accurate estimations that do not make any explicit assumptions about the underlying distribution of the data. Therefore, they are characterized as general-purposed estimators and more flexible as they can scale to large datasets, like the ones recorded from neuroscience studies [3]. However, despite the theoretical potential of deep learning mutual information estimators to perform effectively with neuroscience data, it is still unclear as to which of those estimators performs best on common neuroscience datasets.

1.2 Objectives

The purpose of this paper is to evaluate the efficiency of three recently proposed deep learning mutual information estimators, MINE [3], InfoNCE [40] and FLO [20], when

applied on common neuroscience datasets. Each estimator has been shown to overcome the hurdles of previous estimators. They can effectively handle high-dimensional datasets and large sample sizes while also demonstrating increased flexibility and consistency. However, research focused on evaluating the goodness of these MI estimators specifically when applied on datasets produced during neuroscience experiments is lacking. To address this gap, we assess and compare the performance of each estimator in regard to datasets with properties similar to a common neuroscience dataset but for which the ground truth MI can be attained. By the end of the analysis, the most effective estimator will be employed to calculate the MI of real neuroscience data.

1.3 Results and Outcome

After an extensive comparative analysis of the three estimators across two Gaussian datasets of different dimensionality and one transformed Gaussian dataset similar to a real neuroscience one, it has been found that even though MINE and FLO perform best in simple Gaussian frameworks, InfoNCE outperforms them in the simulated neuroscience framework. By tuning the hyper-parameters: learning rate, number of epochs and hidden layers within its neural structure and finding its optimal negative sample parameter K, InfoNCE converges towards the ground truth MI value with the highest accuracy and stability.

1.4 Structure of Dissertation

The rest of the paper is structured in the following order. In Chapter 2, we offer comprehensive background material on important concepts and definitions from information theory alongside an introduction of the underlying structure of the three deep learning estimators. In Chapter 3 a concise review of current literature around the performance and the comparison of the three deep learning estimators is provided. Subsequently, in Chapter 4, we outline the methodology employed to generate our results which are clearly demonstrated and discussed in Chapter 5, along with which estimator is concluded to perform best on neuroscience datasets . Finally, in Chapter 6 we draw conclusions related to the results and discuss further directions of research.

Chapter 2

Background material

This chapter introduces the three deep learning mutual information estimators with a closer look on their structure and properties. In order to do so, however, we first highlight some relevant theoretical measures and definitions from information theory. Important to note that natural logarithm is used to measure the information in nats.

2.1 Entropy

Entropy is a fundamental concept in information theory, and it measures the level of uncertainty contained in a variable [58]. Entropy plays a key role in quantifying how much information is contained in a random variable. Mutual information which is introduced below uses entropy to quantify the information shared between two random variables.
Given two random discrete variables X and Y with possible states $\{x_1, x_2, \ldots, x_N\}$ and $\{y_1, y_2, \ldots, y_N\}$ respectively.

Definition 2.1.1. *The discrete entropy is defined as [9]*

$$H(X) := -\mathbb{E}[log(p_x(x))] = -\sum_{x}^{N} p_x(x) log(p_x(x)),$$

where $p_x(x)$ is the probability mass function of X, $\mathbb{E}[\cdot]$ is the expectation operator.

Definition 2.1.2. *The joint entropy of X and Y is given by*

$$H(X,Y) := -\mathbb{E}[log(p_{x,y}(x,y))] = -\sum_{x}^{N}\sum_{y}^{N} p_{x,y}(x,y) log(p_{x,y}(x,y)),$$

where $p_{x,y}(x,y)$ is the joint probability mass function of (X,Y) and it expresses the uncertainty of the combination of these two variables.

Now, given two continuous random variables $X \in \mathbb{R}^{d_X}$ and $Y \in \mathbb{R}^{d_Y}$, we can define the differential entropy as below.

Definition 2.1.3. *The differential entropy is defined as [9]*

$$H(X) := -\mathbb{E}[log(p_x(x))] = -\int_X p_x(x) log(p_x(x)) dx,$$

where $p_x(x)$ the probability density function of X.

Definition 2.1.4. *The joint differential entropy is defined as*

$$H(X,Y) := -\mathbb{E}[log(p_{x,y}(x,y))] = -\int_{X,Y} p_{x,y}(x,y) log(p_{x,y}(x,y)) dxdy$$

in which $p_{x,y}(x,y)$ is the joint density function of (X,Y) [9].

2.2 Mutual information

Mutual information was first introduced by C. E. Shannon (1948) in Mathematical theory of communication [52]. It is a quantity that measures the decrease of the uncertainty in a variable X which is resulted by knowing a variable Y. Ever since its initial introduction, it has been a powerful tool in many disciplines, including neuroscience [4, 46, 48, 60]. The definition of mutual information is as follows.

Definition 2.2.1. *The mutual information between X and Y is defined as*

$$I(X,Y) := H(X) + H(Y) - H(X,Y),$$

where H(X,Y) the joint entropy of the two variables.

Equivalently mutual information can be calculated using the probability density functions.

Definition 2.2.2. *The mutual information of two jointly discrete random variables is defined as [9]*

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p_{x,y}(x,y) log \left(\frac{p_{x,y}(x,y)}{p_x(x) p_y(y)} \right)$$

in which $p_{x,y}(x,y)$ the joint probability mass function of X and Y, and $p_x(x)$ and $p_y(y)$ are the marginal probability mass functions of X and Y respectively. In literature the integrand $log \frac{p_{x,y}(x,y)}{p_x(x) p_y(y)}$ is often known as the point-wise mutual information (PMI) [5].

Definition 2.2.3. *The mutual information of two continuous random variables is given by [9]*

$$I(X,Y) = \int_Y \int_X p_{x,y}(x,y) log\left(\frac{p_{x,y}(x,y)}{p_x(x)p_y(y)}\right) dxdy$$

where $p_{x,y}(x,y)$ the joint probability density function of X and Y, and $p_x(x)$ and $p_y(y)$ the marginal probability density functions of X and Y respectively that satisfy $p_x(x) = \int p_{x,y}(x,y)dy$ and $p_y(y) = \int p_{x,y}(x,y)dx$.

Mutual information is a non negative quantity, $I(X,Y) \geqslant 0$, where zero indicates that the two variables, X and Y, are independent.
Additionally, the invariance of mutual information under reparameterizations is a significant property and is demonstrated in **Theorem 2.2.4**. This pivotal theorem shows that the ground truth MI between two variables remains unaltered subsequent to certain transformations applied to them.

Theorem 2.2.4. *Given transformations $X' = F(X)$ and $Y' = G(X)$, the joint probability density function of the random variables X' and Y' is denoted as $p'_{x,y}(x',y')$. Thus, we obtain [32]*

$$\begin{aligned} I(X',Y') &= \int\int p'_{x,y}(x',y') log\left(\frac{p'_{x,y}(x',y')}{p'_x(x')p'_y(y')}\right) dx'dy' \\ &= \int\int p_{x,y}(x,y) log\left(\frac{p_{x,y}(x,y)}{p_x(x)p_y(y)}\right) dxdy \\ &= I(X,Y) \end{aligned}$$

2.3 Kullback-Leibler divergence

The Kullback-Leibler divergence is a measure that assesses how one probability distribution P is different from a second, reference probability distribution Q [24, 33]. This measure alongside its dual representation are the foundation from which the MINE estimator was created [3].

Definition 2.3.1. *The Kullback-Leibler divergence (KLD), also known as the relative entropy, between two probability distributions P and Q is defined as*

$$D_{KL}(\mathbb{P} \parallel \mathbb{Q}) := \mathbb{E}_{\mathbb{P}}\left[log\frac{d\mathbb{P}}{D\mathbb{Q}}\right] \geqslant 0$$

Definition 2.3.2. *The mutual information is equivalent to the Kullback–Leibler (KL) divergence between the joint distribution $\mathbb{P}_{XY}$ and the product of the marginals distributions $\mathbb{P}_X \otimes \mathbb{P}_Y$:*

$$I(X,Y) = D_{KL}(\mathbb{P}_{XY} \parallel \mathbb{P}_X \otimes \mathbb{P}_Y).$$

The dual representation of KL-divergence

The following Theorem provides a representation of the KL-divergence as outlined in the work of Donsker and Varadhan (1983).

Theorem 2.3.3. *(Donsker-Varadhan representation) The Kullback-Liebler Divergence admits the following dual representation [14]*

$$D_{KL}(\mathbb{P} \parallel \mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - log(\mathbb{E}_{\mathbb{Q}}[e^T]),$$

where the supremum is evaluated over all functions T for which both expectations are finite.

Lemma 2.3.4. *(Lower bound for the Kullback Liebler Divergence) For any F class of functions $T : \Omega \to \mathbb{R}$ that satisfy the constraints of the Theorem 2.3.3, the following inequality holds*

$$D_{KL}(\mathbb{P} \parallel \mathbb{Q}) \geqslant \sup_{T\in F} \mathbb{E}_{\mathbb{P}}[T] - log(\mathbb{E}_{\mathbb{Q}}[e^T]).$$

2.4 MI and unnormalized statistical models

In this section, we introduce some key definitions which are essential for the introduction of the FLO estimator in **Section 2.5.3**. To begin, we aim to connect the mutual information to unnormalized statistical models. Therefore, we first consider the classical MI estimator called Barber-Agarov (BA) as presented by D. Barber et al [2].

Definition 2.4.1. *(Barber-Agavok MI estimator) Considering a variational approximation $q(y \mid x)$ of the posterior $p(y \mid x) = \frac{p_{x,y}(x,y)}{p_x(x)}$ we obtain*

$$\begin{aligned} I(X,Y) = \mathbb{E}_{p_{x,y}(x,y)}\left[log\frac{p_{x,y}(x,y)}{p_x(x)p_y(y)}\right] &= \mathbb{E}_{p_{x,y}(x,y)}\left[log\frac{p(y \mid x)}{p_y(y)}\right] \\ &= \mathbb{E}_{p_{x,y}(x,y)}\left[log\frac{q(y \mid x)}{p_y(y)}\right] + \mathbb{E}_{p_x(x)}[KL(p(y \mid x) \parallel q(y \mid x))] \\ &\geqslant \mathbb{E}_{p_{x,y}(x,y)}\left[log\frac{q(y \mid x)}{p_y(y)}\right] \\ &\triangleq I_{BA}(X,Y \mid q) \end{aligned}$$

This naïve BA bound is used to estimate an estimator called unnormalized Barber-Agavok bound (UBA) that is applicable to unnormalized statistical modelling [44].

Definition 2.4.2. *(Unnormalized Barber-Agavok MI estimator) Setting* $q_\theta(y \mid x) = \frac{p_y(y)}{Z_\theta(x)} e^{g_\theta(x,y)}$ *in which* $e^{g_\theta(x,y)}$ *the tilting function and* $Z_\theta(x) = \mathbb{E}_{p_y(y)}[e^{g_\theta(x,y)}]$ *the associated partition function, we end up with the bound*

$$I_{UBA}(X,Y \mid g_\theta) \triangleq \mathbb{E}_{p_{x,y}(x,y)}[g_\theta(x,y) - logZ_\theta(x)] = \mathbb{E}_{p_x(x)}\left[\mathbb{E}_{p(y|x)}\left[log\frac{e^{g_\theta(x,y)}}{Z_\theta(x)}\right]\right]$$

This intractable UBA bound has been the foundation for several MI bounds.

2.5 Deep learning Mutual information estimators

Deep learning mutual information estimators are based on the idea of using the popular machine learning technique called artificial neural networks and specifically deep neural networks to produce more accurate estimations between multidimensional variables.

An artificial neural network (ANN) is a machine learning algorithm that draws inspiration from biological neural networks [27]. Similar to the cells in biological systems, an ANN consists of nodes that correspond to cell bodies. These nodes communicate with each other through connections, similar to the axons and dendrites in biological neurons. In biological neural networks, synapses between neurons are strengthened when their neurons exhibit correlated outputs. Similarly, in an ANN, the connections between nodes are assigned weights based on their ability to produce the desired outcome. Deep neural networks have multiple hidden layers with deep architecture and are used in estimating the MI due to its ability to learn complex patterns and relationships in data [8].

Below, we introduce three deep learning mutual information estimators, MINE, InfoNCE and FLO, which we will compare and analyse.

2.5.1 Mutual Information Neural Estimator (MINE)

Mutual Information Neural Estimator (MINE) was proposed by Belghazi et al (2018) as a general-purposed estimator which relies on the characterization of the mutual information as the Kullback-Leibler (KL-) divergence. As mentioned in **Definition 2.3.1**, the Kullback-Leibler (KL-) divergence, as introduced by Kullback in 1997, quantifies the difference between the joint distribution and the product of the marginals.

Based on the equation for mutual information (**Definition 2.3.2**) and the dual representation of KL-divergence (**Definition 2.3.3**), the approach involves selecting F to be the set of functions $T_\theta : X \times Y \to \mathbb{R}$, which is parametrized by a deep neural network with parameters $\theta \in \Theta$. This neural network is referred to as the *statistics network*. Employing the following bound,

$$I(X;Y) \geq I_\Theta(X,Y)$$

where the neural information measure, denoted as $I_\Theta(X,Y)$, is defined as follows,

$$I_\Theta(X,Y) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XY}}[T_\theta] - log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y}[e^{T_\theta}]).$$

In order for the expectations to be estimated in the above equation, empirical samples from P_{XY} and $\mathbb{P}_X \otimes \mathbb{P}_Y$ are used, or the samples from the joint distribution are shuffled along the batch axis. Maximizing the objective can be achieved through gradient ascent.

Definition 2.5.1.1. *(Mutual Information Neural Estimator) Suppose that $F = \{T_\theta\}_{\theta\in\Theta}$ the set of functions parametrized by a neural network, then MINE is defined as [3],*

$$\widehat{I(X;Y)}_n = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}^{(n)}_{XY}}[T_\theta] - log(\mathbb{E}_{\mathbb{P}^{(n)}_X \otimes \widehat{\mathbb{P}}^{(n)}_Y}[e^{T_\theta}])$$

in which given a distribution $\mathbb{P}$, we denote $\widehat{\mathbb{P}}^{(n)}$ to be the empirical distribution associated to n i.i.d. samples.

Properties

Capture of nonlinear dependencies:

An important property of mutual information is its ability to be invariant towards nonlinear transformation between random variables which have the relationship $Y = f(X) + \sigma \odot \varepsilon$ where f a deterministic nonlinear transformation and ε a random noise. MINE has been proved to capture the important property called equitability [31] which ensures the quantification dependence without bias for the relationship [3].

Consistency:

MINE relies on two main characteristics: the choice of statistical network and the number of samples of the data distribution $\mathbb{P}_{XY}$. An estimator, $\widehat{I(X;Y)}_n$, is considered to be strongly consistent if for any given value of $\varepsilon > 0$ there exists a positive integer N and a selection of a statistics network such that

$$\forall n \geq N, \quad \left| I(X,Y) - \widehat{I(X;Y)}_n \right| \leq \varepsilon, \quad a.e.$$

for which the probability is over a set of samples.

Generally, consistency is divided into two problems. An approximation problem related to the size of the family, F, which is addressed by the universal approximation theorems for neural networks [25]. As well as an estimation problem in regard to the use of empirical measures which involve classical consistency theorems for extremum estimators, as outlined in the work of Van de Geer in 2000 [12]. Based on two Lemmas that are shown in **Appendix A.1**, the MINE estimator has been proven to accurately approximate mutual information, as well as almost surely converge to a neural information measure with increasing samples.

2.5.2 InfoNCE estimator

InfoNCE information estimator was first introduced by van den Oord, et al. (2018) as part of their paper called "Representation Learning with Contrastive Predictive Coding" [40] and later formalized in the work of [44]. This estimator aims to optimize a loss function which is based on the idea of Noise Contrastive Estimation (NCE). NCE is a method for estimating the parameters of a statistical model by contrasting positive samples from the target distribution with carefully chosen "negative" samples from a noise distribution [21]. Below we introduce InfoNCE's loss function as well as its formal MI estimator definition.

Definition 2.5.2.1. *(InfoNCE Loss function) Given a set $X = x_1, \ldots, x_K$ of K random samples that contains a positive sample from $p(x_{t+k}|y_t)$ and $K-1$ negative samples from the 'proposal' distribution $p(x_{t+k})$. Then we aim to optimize the loss function*

$$\mathcal{L}_K = -\mathop{\mathbb{E}}_{X}\left[log\frac{f_k(x_{x+k}, y_t)}{\sum_{x_j \in X} f_k(x_j, y_t)}\right],$$

where $f_k(x_{t+k}, y_t)$ the density ratio which holds the following property

$$f_k(x_{t+k}, y_t) \propto \frac{p(x_{t+k}|y_t)}{p(x_{t+k})}$$

.

Definition 2.5.2.2. *(InfoNCE MI estimator) The mutual information constructed using the InfoNCE loss function (Definition 2.5.2.1) is*

$$I(x_{t+k}, c_t) \geq log(K) - \mathcal{L}_K.$$

(proof given by [40])

Definition 2.5.2.3. *InfoNCE is formally defined by B. Poole et al. (2019) as*

$$I^K_{InfoNCE}(X;Y|f) \triangleq \mathbb{E}_{p^K(x,y)}\left[log\frac{f(x_1,y_1^{\oplus})}{\frac{1}{K}\sum_{k'} f(x_1,y_{k'}^{\ominus})}\right], \quad I^K_{InfoNCE}(X;Y) \triangleq \max_{f\in F}\{I^K_{InfoNCE}(X;Y|f)\}$$

*Using its formal Definition, InfoNCE is shown to be related to the MINE estimator (**Appendix A.2**)*

In InfoNCE, a heuristic is used to differentiate positive samples from negative samples. The positive samples originate from the joint data distribution $p(x,y)$, while the negative samples are randomly paired samples derived from the corresponding marginal distributions $p(x)$ and $p(y)$. In this case, $f(x,y) > 0$ is known as the *critic function*. The notation $p^K(x,y)$ is used to represent K independent draws or the sample size (often referred to as the mini-batch size). The symbols $\oplus$ and $\ominus$ represent the positive and negative samples, respectively.

It has been shown that when the sample labels are clean, increasing the negative sampling ratio K results in a tighter lower bound of variable mutual information [40, 63]. This typically leads to improved performance, as a greater amount of information from negative samples is utilised during model training.

Lemma 2.5.2.4. *InfoNCE is an asymptotically tight mutual information lower bound [44]*

$$I(X;Y) \geq I^K_{InfoNCE}(X;Y|f), \qquad \lim_{K\to\inf} I^K_{InfoNCE}(X;Y) \to I(X;Y).$$

2.5.3 Fenchel-Legendre Optimization (FLO)

The Fenchel-Legendre Optimization (FLO) is a novel contrastive learning framework for estimating mutual information that was introduced by Q. Guo et al (2022) in their paper "Tight Mutual Information Estimation With Contrastive Fenchel-Legendre Optimization". FLO is trying to overcome the limitations that previous estimators face like the InfoNCE [40] which requires a large and very costly batch training as well as reducing variance by sacrificing bound tightness. In order to succeed this the estimator exploits the connection between MI estimation, unnormalized statistical modelling and convex optimization.

The key into the construction of the FLO estimator is the lower bound MI using the Fenchel-Legendre transform technique. The technique is defined as follows.

Definition 2.5.3.1. *(Fenchel-Legendre duality) Consider a proper and lower-semicontinuous convex function $f(t)$. We define its convex conjugate function $f^*(v) = \sup_{t\in D(f)}\{tv - f(t)\}$, where $D(f)$ represents the domain of function f . The function $f^*(v)$ is referred to as the Fenchel-Legendre conjugate of $f(t)$ and satisfies the properties of convexity and lower-semicontinuity. The Fenchel-Legendre conjugate pair (f, f^*) are dual to each other, meaning that $f^{**} = f$ giving $f(t) = \sup_{v\in D(f^*)}\{vt - f^*(v)\}$.*
For example, the Fenchel-Legendre dual for $f(t) = -log(t)$ is $f^*(v) = -1 - log(-v)$.

To begin, we take the integrand from the UBA in **Definition 2.4.2** and proceed with the following steps to structure the FLO lower bound.

$$log\frac{e^{g_\theta(x,y)}}{Z_\theta(x)} = -log\mathbb{E}_{p(y')}[e^{g(x,y')-g(x,y)}]$$

Using the Fenchel inequality of $-log(t)$ from **Definition 2.5.3.1**

$$log\frac{e^{g_\theta(x,y)}}{Z_\theta(x)} \geq \{-u - e^{-u}\mathbb{E}_{p(y')}[e^{g(x,y')-g(x,y)}]\} + 1 \quad \text{for all } u \in \mathbb{R}$$

So, the following inequality holds for any function $u_\phi(x,y) : X \times Y \to \mathbb{R}$

$$log\frac{e^{g_\theta(x,y)}}{Z_\theta(x)} \geq -\{u_\phi(x,y) + e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[e^{g(x,y')-g(x,y)}]\} + 1$$

Finally, substituting the final result into the implementation of the UBA lower bound in **Definition 2.4.2** we obtain the Fenchel-Legendre Optimization (FLO) MI lower bound.

Definition 2.5.3.2. *The FLO mutual information lower bound is*

$$I_{FLO}(X;Y|g_\theta,u_\phi) \triangleq \mathbb{E}_{p(x,y)}\left[-\{u_\phi(x,y) + e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[e^{g_\theta(x,y')-g_\theta(x,y)}]\}\right] + 1$$

Definition 2.5.3.3. *The FLO mutual information can be estimated using the following naïve empirical K estimator [20]*

$$\hat{I}^K_{FLO}(X;Y|g_\theta,u_\phi) \triangleq -\left\{u_\phi(x_i,y_i) + e^{-u_\phi(x_i,y_i)}\frac{1}{K-1}\sum_{j\neq i} e^{g_\theta(x_i,y_j)-g_\theta(x_i,y_i)}\right\} + 1$$

It is important to note that since $\hat{I}^K_{FLO}$ is not enclosed by a convex log transformation then $I^K_{FLO} \triangleq \mathbb{E}_{p^K}[\hat{I}^K_{FLO}]$ is an unbiased estimator for $I_{FLO}(X;Y|g_\theta,u_\phi)$ that does not depend on the batch size K.

Properties of FLO:

- I_{FLO} is a tight estimator, which means that the ground truth mutual information can be estimated using a specific choice of $g_\theta(x,y)$ and $u_\phi(x,y)$.
- I^K_{FLO} can be effectively optimized for any batch size K.

Chapter 3

Literature review

The MI metric was first introduced and analysed by C. Shannon in his landmark paper " A Mathematical Theory of Communication". Since then, it has been a fundamental measure in information theory. Its applications span across various domains, including neuroscience [13, 42, 58, 64], where it played a pivotal role in uncovering complex relationships between neurons.

Classical MI estimators have been extensively studied and compared in the literature [6, 15, 62]. Numerous studies have investigated the applicability and performance of these classical methods in the neuroscience domain [1, 26, 28, 45]. For example, it was shown that MI estimators outperform the commonly used Statistical Parametric Mapping technique for identifying regionally specific effects in neuroimaging data [19] with the k Nearest Neighbours estimator (k-NN) [32] being the best alternative.

Another example of the applicability of MI in neuroscience research is given by the study of B.C. Souza et al [57]. Here, the focus was given on the identification of the types of cells that are related to the hippocampal circuitry. Accurately recognizing these cells involves assessing the information carried by spikes regarding navigation characteristics. Skaggs et al. [54, 55] introduced key measures, derived from Shannon's MI [52], for estimating such information. The main task of the paper was to investigate the performance of those metrics against the Shannon's original MI metric. After using both simulated and real neuroscience data, it was observed that the existing information metrics have a weaker connection with spatial decoding accuracy compared to the original MI metric, which performed well under a variety of different scenarios.

Nevertheless, even though the valuable contribution of classical MI estimators in uncovering the functionalities of the brain, as neuroscience data have become increasingly complex and high-dimensional, their limitations are more apparent, prompting

the exploration of alternative approaches [10]. The authors of [49] had studied the effectiveness of various upper and lower bound techniques on neural codes, discovering their unreliability and their tendency on making strong assumptions about the data. Recently, deep learning MI estimators have been proposed showcasing improved accuracy and robustness [3, 20, 56]. So far, however, limited papers can be found around the comparison of deep learning MI estimators and are mainly in the context of validating the effectiveness of newly proposed estimators.

In their paper titled "Mutual Information Neural Estimation," Belghazi et al. proposed a framework based on deep neural networks to estimate mutual information. They introduced a variational lower bound on mutual information and utilised deep generative models to approximate it. The resulting estimator is called MINE. In order to demonstrate the effectiveness of their estimator, MINE was compared to the k-NN based non-parametric estimator [32], showing clear improvement when estimating the MI between 20D multivariate Gaussian random variables. Subsequently, MINE was applied on more complex settings of machine learning such as palliate mode-dropping in GANs and Information Bottleneck method, demonstrating its effectiveness.

However, MINE was one of the first deep learning estimators proposed and no comparison of MINE with other deep learning estimators is included in the paper. On the other hand, Q. Guo et al (2022) in their paper have introduced FLO which was compared with various other estimators like InfoNCE [40], NWJ [39] and TUBA [2]. The experiments conducted involved the comparison of estimators on multivariate random Gaussian variables (2D and 20D) as well as using the Bayesian optimal experiment design (BOED) and a novel meta-learning framework. All clearly showed strong empirical evidence of the superiority of the new FLO bound over its predecessors. Even though, FLO was proved to be performing better on these applications, the focus was mainly to prove the superiority of FLO in comparison to InfoNCE as both estimators are built on contrastive MI bounds. In the study, MINE estimator was only mentioned regarding its theoretical approach but was not included in the experiments. Lastly, the experiments included in [20] lack the applicability of the estimators on more complex data such as neuroscience data and their performance to estimate MI between high dimensional noisy recordings.

B. Poole et al in their paper called "On Variational Bounds of Mutual Information" provide a detailed review of existing estimators, including MINE and InfoNCE, and discuss their relationships and trade-offs. They have also conducted some experiments to evaluate the performance of the MI bounds on two tractable toy problems. The first

one followed the same 20D correlated Gaussian problem as shown in [3] to assess the estimations of MI of (x,y) over the correlation value. The second problem used samples for which a linear transformation followed by a cubic non linearity was applied on y giving $(x, (Wy)^3)$.

For both problems, a thorough analysis of the bias/variance trade-offs was conducted showing that multi-sample estimates of InfoNCE result in low variance but saturate at log(batch size). Moreover, the influence of the critic structure on the efficiency-accuracy trade-off was highlighted. The MINE estimator uses a joint critic in which x and y are fed together as an input to the network, whereas the InfoNCE estimator which is mentioned in the study but also FLO, are structured with a separable critic. Separable critics require $2N$ forward passes for a batch size N and joint ones require N^2 forward passes. Through their experiments on the two problem settings mentioned above, it was deducted that joint critics generally performed worse but separable critics required larger neural networks to provide similar performance. These two key findings will also be seen in our analysis.

Chapter 4

Methodology

The main focus of this paper is to test and assess the performance of each deep learning estimator when it comes to high dimensional continuous data recorded from neuroscience studies. A detailed description regarding the methodology to achieve that can be found in the respective subsections below.

The environment chosen to perform this analysis is Python due to the rich set of numerical computation libraries for result and graph production such as numpy [22], scipy[61] and holoviews. Moreover, all the machine learning implementations and model training tasks of the deep learning MI estimators is conducted using PyTorch [43], a highly regarded and versatile deep learning framework. To be more precise, the pytorch implementations used in the analysis below for InfoNCE and FLO are the ones generated in the original proposal paper of FLO [20] whereas for MINE can be found in `https://github.com/MasanoriYamada/Mine_pytorch.git`.

Using the above implementations we have established a baseline setup which remains constant throughout the analysis unless explicitly stated otherwise. For the FLO and InfoNCE estimators, the critic functions g(x, y), u(x, y) and u(x), use multilayer perception (MLP) network construction with 2 hidden layers (512×512) and a default batch size of 128. Similarly, MINE also employs a 2 hidden layer structure (512 x 512). In all models, the ReLU activation function and the Adam optimizer is applied.

4.1 Neuroscience dataset

For the purpose of applying and assessing the performance of the three deep learning MI estimators on real neuroscience data, we make use of the dataset that contains the results of the study conducted by Henschke et al. [23]. In their paper called "Reward

Association Enhances Stimulus-Specific Representations in Primary Visual Cortex", they describe the use of two-photon calcium imaging placed in fixed head position on various animals including mice. This way they could monitor the neuronal activity that expresses the genetically encoded calcium indicator GCaMP6 in the primary visual cortex (V1). The main task was for the subjects to be placed on the linear virtual corridor and lick on a specific spatial location that was marked by a visual cue, in order to be rewarded. During this process, neural signals were captured before, during and after the subjects acquired the ability to locate a reward on the corridor. The resulting dataset contains recordings of the V1 neural activity, alongside the position and time of the subjects on the linear corridor for different training days, combining multiple continuous variables of high dimensionality. Similar data were also produced by [41] using only mice.

After choosing a specific training day of an animal, we focus on the continuous recordings of the raw calcium signals of multiple neurons in regard to the speed $[\frac{position}{time}]$ of this animal on the linear virtual corridor. The raw calcium signal variable of each neuron of the V1 cortex contain 22,366 signal points. Similarly, the position variable contains 22,366 data points measuring the distance on the linear corridor and the time variable provides the time when each of the data points was captured.

For the purpose of our analysis, we will selectively utilise a reduced subset comprising only 5 out of the 34 neurons from which their raw calcium signal recordings are captured. Therefore, from this point on any mention of the "original/real neuroscience dataset", $X_{original}$, refers to the 22,366 x 6 matrix containing the speed and the raw calcium signal recordings of 5 V1 cortex neurons. Subsequently, using the different MI estimators under investigation we will examine the extend of the information that these 5 neuron signals provide about the speed of the subject on the linear virtual corridor.

4.2 Comparison of the deep learning estimators

The deep learning MI estimators, MINE, InfoNCE and FLO, will undergo a comparative analysis using three distinct datasets: a 2-dimensional Gaussian dataset, a 20-dimensional Gaussian dataset, and a simulated Gaussian dataset designed to emulate the properties of the original neuroscience data. Throughout the comparison, careful consideration will be given to the performance of each estimator in relation to the characteristics of the datasets, including the number of samples and the correlation.

However, the main focus will be given on which values of the different hyper-

parameters of the deep neural network (DNN) that is incorporated in each MI estimator result in better estimations. This process is called hyper-parameter tuning and refers to the iterative process of selecting the best combination of hyper-parameters that leads to significant improvement in the performance of a DNN model [34]. It is computationally expensive but a very important process for the convergence of the estimator towards the true MI value. In our analysis the learning rate (lr), the number of epochs and the number of hidden layers will be examined to discern their individual and combined impacts on the estimator's performance. Studies have shown that these hyper-parameters are generally the most influential [30, 37, 53].

The hyper-parameters of a neural network mentioned above are defined as:

- **Learning rate**: determines the step size at which the optimizer updates the model weights during training.
- **Number of epochs**: is the number of times the whole training data is shown to the network.
- **Number of hidden layers**: defines how many layers are between the input layer and the output layer of the neural network.

4.2.1 Simple Gaussian distribution

First, we test the estimators on simple 2-dimensional and 20-dimensional Gaussian models. The Gaussian models are selected as we can easily calculate their ground truth MI and therefore using that we can evaluate whether the estimators perform well.
The Gaussian model has the general form of:

$$(X,Y) \sim \mathcal{N}(0,\Sigma)$$

where Σ represents the joint covariance of X and Y. This allows us to calculate the ground truth MI using the formula:

$$I(X,Y)_{true} = \frac{1}{2} log\left(\frac{|\Sigma_X||\Sigma_Y|}{|\Sigma|}\right) \tag{4.1}$$

Specifically, the precise value of $I(X,Y)$ for the bivariate Gaussian distribution can be derived analytically using the **Definition 2.2.1** (proof in **Appendix A.3**), resulting in

$$I(X,Y)_{true} = -\frac{1}{2} log(1-\rho^2) \tag{4.2}$$

To begin, we examine the bivariate Gaussian distribution with unit variance for each marginal. Having the value of the ground truth mutual information, we can estimate

the $Error = |\hat{I}(X,Y) - I(X,Y)_{true}|$ which can be plotted against the number of samples (N) for different correlations values (ρ). This graph will show how each estimator performs under the different samples sizes when the number of epochs is fixed to 100. Additionally, for the same 2-dimensional variables we investigate the resulting estimate against a varying level of correlation $\rho \in [0, 0.9]$, as outlined in Belghazi et al [3]. The InfoNCE estimator will be additionally tested with different values of its extra parameter called negative samples (K) [20]. From the Properties of FLO in **Section 2.5.3**, the FLO estimator's performance is independent of K.

For the 20D Gaussian models we will conduct a more thorough investigation on how the different hyper-parameters that are integrated in the DNN affect the resulting estimations generated with each MI estimator. To assess that, multiple graphs featuring different values of the correlation between the variables of the dataset, as well as the hyper-parameters; learning rate and number of epochs [20] will be used. Subsequently, using the best performing hyper-parameter values, we proceed to plot the resulting estimators against the varying value of the correlation as described in the 2D case.

4.2.2 Simulated Gaussian data

The main deliverable of the analysis is the choice of the most appropriate MI estimator for neuroscience data. In order to accomplish that, an evaluation of the different deep learning MI estimators must be conducted using transformed multivariate Gaussian distribution data that hold properties and correlations as close to the ones of the original neuroscience data in **Section 4.1**. Based on the **Definition 2.2.4**, it can be established that the ground truth mutual information of the transformed (simulated) data is equivalent to the mutual information of the initial multivariate Gaussian data which can be computed using the **Formula 4.1**. The detailed process for the creation of the simulated dataset follows the pipeline shown in **Figure 4.1** and is detailed described in the below steps.

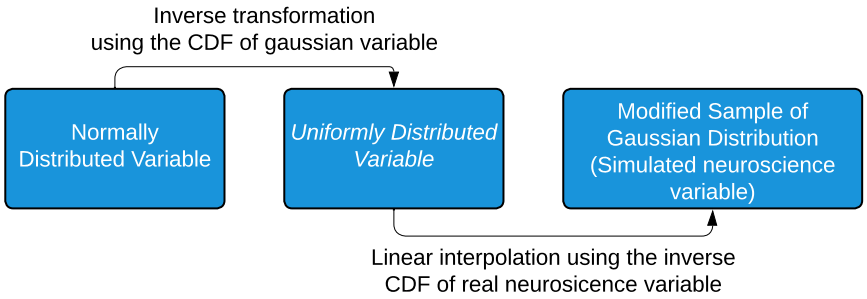

Figure 4.1: Process of generating each variable of the simulated neuroscience dataset.

First we generate six variables, one for the speed (X_{speed}) and the rest for the raw calcium signals of the five neurons ($X_{neurons}$), using the multivariate Gaussian distribution with a zero mean vector and a correlation matrix. The correlation matrix is carefully selected to minimize the sum of absolute differences between the upper triangular portion of the original correlation matrix obtained from the original neuroscience data ($X_{original}$) and the resulting correlation matrix derived from the simulated data ($X_{simulated}$). Furthermore, the size of each variable in the multivariate Gaussian matches that of the original dataset, which consists of 22,366 data points.

In accordance with the inverse transform theorem (ITT), the subsequent step involves the transformation of each Gaussian-distributed variable into a uniformly distributed variable by using its respective cumulative density function (CDF). For example, let X_{speed} be the continuous normal variable for Speed (**Figure 4.2 (A)**) with CDF equal to $F(x_{speed})$. Then $F(X_{speed}) \sim Uniform[0,1]$, which is shown in **Figure 4.2 (B)**.

Upon obtaining the uniformly distributed variables, the subsequent course of action involves determining the empirical cumulative distribution function (CDF) for each variable present in the original neuroscience dataset. Subsequently, employing linear interpolation, we can map the values of each simulated uniformly distributed variable with respect to its original inverse empirical CDF. For reference, **Figure 4.2 (C)** exhibits the empirical CDF (y) of the original speed variable. This process yields the resulting simulated variables, denoted as $X_{simulated}$, which exhibit distributions resembling those observed in the original neuroscience dataset. This concludes the steps for generating the simulated data which are the result of a transformation on the initial Gaussian ones.

For example, the process for generating the simulated speed variable is:

$$X_{speed} \xrightarrow{F(X_{speed})} U_{speed} \xrightarrow{F^{-1}_{ECDF}(U_{speed})} X_{speed'}$$

This process is also graphically illustrated in **Figure 4.2**. The resulting histogram of the simulated data for all variables are shown **Figure 4.3** where high similarity between the distributions of the simulated and the original variables can be observed.

Below, we also demonstrate the correlation matrix of the simulated data, $X_{simulated}$, and the correlation matrix of the original neuroscience data, $X_{original}$. By calculating the sum of the absolute differences between their upper triangular elements, we obtain an approximate value of 0.13564.

$$Corr(X_{simulate}) = \begin{bmatrix} 1.000000 & 0.020278 & 0.219279 & 0.409766 & 0.052962 & -0.229867 \\ 0.020278 & 1.000000 & 0.795651 & 0.102481 & 0.115814 & 0.024715 \\ 0.219279 & 0.795651 & 1.000000 & 0.284051 & 0.118855 & -0.063783 \\ 0.409766 & 0.102481 & 0.284051 & 1.000000 & -0.287832 & -0.138513 \\ 0.052962 & 0.115814 & 0.118855 & -0.287832 & 1.000000 & 0.006407 \\ -0.229867 & 0.024715 & -0.063783 & -0.138513 & 0.006407 & 1.000000 \end{bmatrix}$$

$$Corr(X_{original}) = \begin{bmatrix} 1.000000 & 0.067342 & 0.218823 & 0.409769 & 0.083801 & -0.234602 \\ 0.067342 & 1.000000 & 0.795653 & 0.109486 & 0.115823 & 70.030216 \\ 0.218823 & 0.795653 & 1.000000 & 0.284051 & 0.138580 & -0.063492 \\ 0.409769 & 0.109486 & 0.284051 & 1.000000 & -0.282916 & -0.138449 \\ 0.083801 & 0.115823 & 0.138580 & -0.282916 & 1.000000 & -0.008621 \\ -0.234602 & 0.030216 & -0.063492 & -0.138449 & -0.008621 & 1.000000 \end{bmatrix}$$

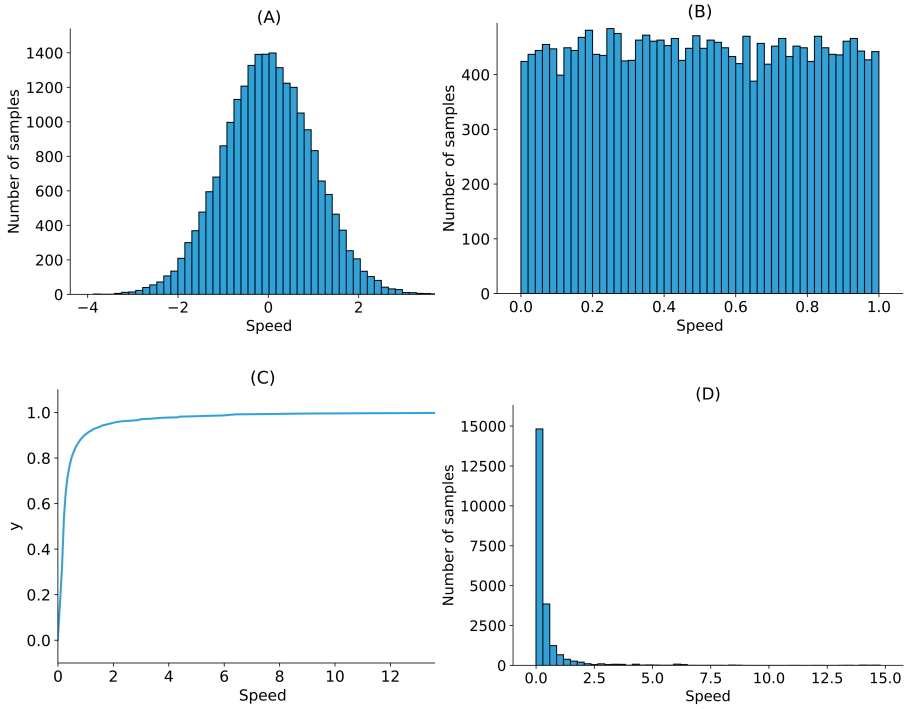

Figure 4.2: Graphical representation of the process for generating the simulated Speed variable starting from the multivariate Gaussian. (A) Gaussian distribution of Speed, X_{speed} (B) Uniform distribution of Speed, U_{speed} (C) Empirical CDF of original Speed, $F_{ECDFspeed}$ (D) Histogram of the simulated Speed, $X_{speed'}$

With the simulated dataset at hand, it remains to calculate the ground truth (GT) mutual information using the initial multivariate Gaussian data and the **Formula 4.1**. Specifically, in this case we have:

$$I(X'_{speed}, X'_{neurons})_{GT} = I(X_{speed}, X_{neurons}) = 0.5787373599348244,$$

which is the GT MI between the simulated one-dimensional speed variable and the five-dimensional simulated matrix containing the raw calcium signals of the 5 neurons.

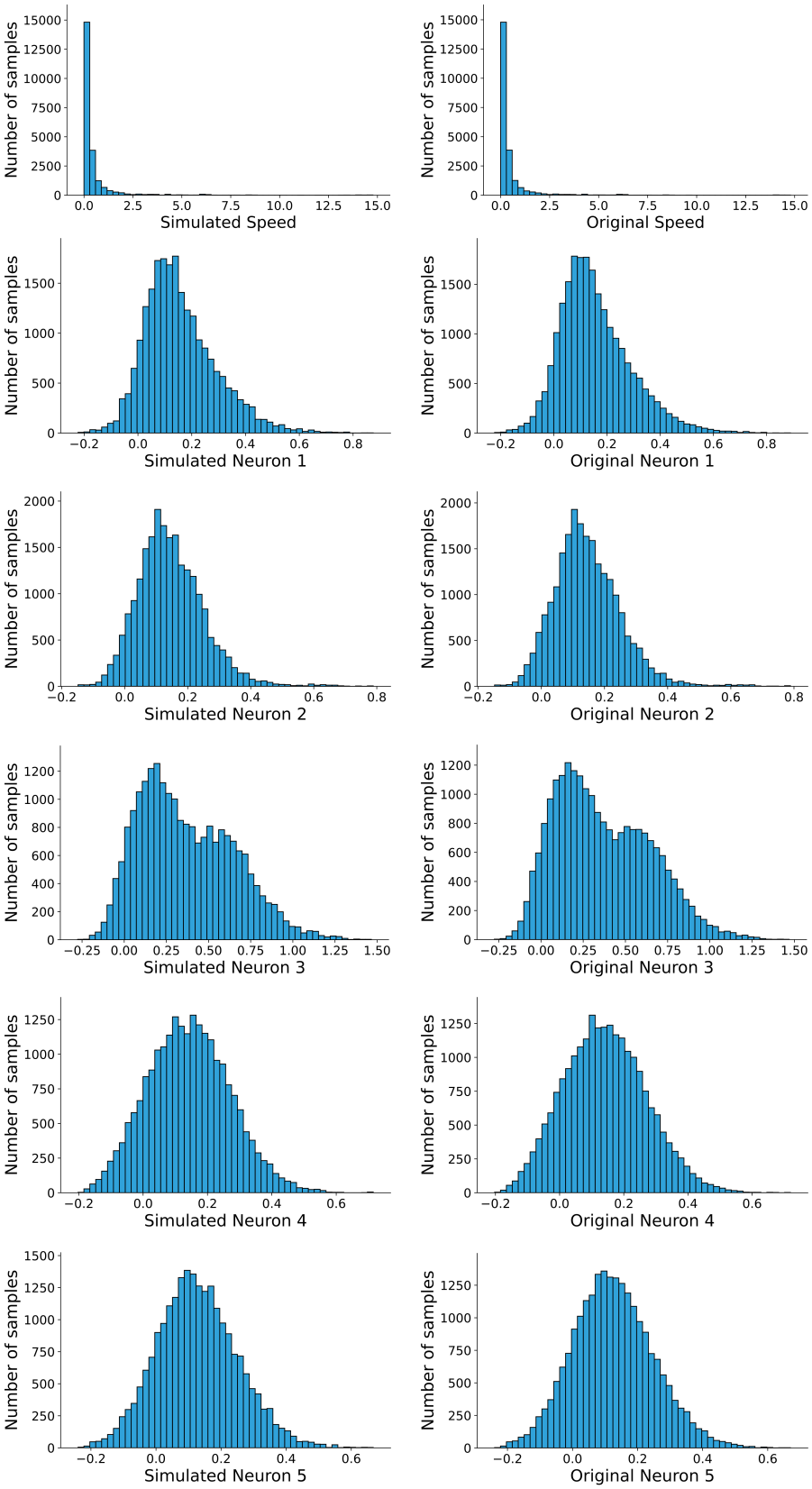

Figure 4.3: Histogram of each simulated variable side-by-side its respective histogram from the original neuroscience dataset. In comparison, the distributions are almost identical.

In order to proceed with the application of the simulated data and later of the real neuroscience data on our three deep learning MI estimators we first need to make a few alternations on the original implementations. The existing ones can only take equal dimensional X and Y inputs for estimating $I(X,Y)$. However in our case X is one dimensional whereas Y is five dimensional. To fix that a few changes were made on how the estimators' implementations take and process the matrix containing the input data. For the MINE implementation we also needed to set two extra input parameters which define the dimensionality of X and Y.

Proceeding with the evaluation of each deep earning MI estimator when applied on the simulated neuroscience data, we will adopt a comprehensive strategy for hyper-parameter tuning. By leveraging the insights gained from analyzing the results obtained from the 2D and 20D Gaussian datasets, we make targeted decisions regarding the values of the hyper-parameters to investigate.

Initially, we will construct a grid search of possible combinations of the hyper-parameters; learning rate, number of hidden layers, and number of epochs. Subsequently, we will evaluate each MI estimator against the ground truth value for each combination, employing a 3 by 3 figure to better visualize the outcomes.

Given that the estimators are trained on a neural network, it is important to acknowledge that the resulting MI estimations may not always be identical. Therefore, the next step will involve computing the Mean Absolute Error (MAE) of the hyper-parameter combinations of each estimator that yields the most promising performance during the grid search.

MAE is a metric of the average magnitude of the errors within a set of estimations. Its formula is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

where n represents the number of estimations being evaluated, y_i the ground truth value and $\hat{y}_i$ the estimated value. In our case we will execute each estimator five times, employing the best-performing combinations of hyper-parameters and show the resulting MAE values in a table for comprehensive analysis.

Furthermore, the InfoNCE estimator incorporates an extra parameter, denoted as K, which represents the size of the negative sample used for training as explained in **Definition 2.5.2.1**. While a higher negative sampling ratio K theoretically results in a more precise lower bound for variable mutual information, leading to improved performance, real datasets, such as those in neuroscience, contain noise [58]. Consequently,

incorporating a large number of noisy negative samples during model training may produce counterproductive outcomes [63]. To determine the optimal value of K for neuroscience data, a graph of InfoNCE will be generated using the best hyper-parameters identified through the previous analysis. This graph will demonstrate the estimated MI for $K = 5, 10, 20$ and 40, in order to identify the most effective value for the given dataset.

Lastly, the three estimators will be executed ten times using the best-performing hyper-parameter values obtained overall. The MAE will be extracted from these executions and recorded on a table. The estimator yielding the smallest MAE will be considered as the most suitable estimator among the three for neuroscience data.

4.3 Application on real neuroscience data

For the final step of our analysis, the resulting most promising deep learning MI estimator between MINE, InfoNCE, and FLO will be applied, using its best hyper-parameter combination, on the original neuroscience dataset mentioned in **Section 4.1**.

Chapter 5

Results

In this Section, we provide the results from testing the three deep learning mutual information estimators and comparing their performance against each other. In the first section, we provide the plots from testing the estimators on simple Gaussian generated samples. Then, we proceed to apply them to the simulated data as mentioned in **Section 4.2.2**. Lastly, the value of the real neuroscience data (**Section 4.1**) MI is showcased as estimated using the resulting best deep learning MI estimator based on the results that arises from the comparison analysis.

5.1 Gaussian distributions

5.1.1 2D Gaussian variables

Previous studies [3, 20] have demonstrated the efficacy of MINE, InfoNCE, and FLO in accurately estimating MI between two-dimensional Gaussian variables, regardless of their correlation. InfoNCE has been shown to provide improved estimations with larger negative sample sizes (K) [40, 63], while FLO's performance remains independent of K [20]. Building on these findings, our investigation aims to evaluate and compare the three estimators across various correlation values and sample sizes of 2D Gaussian variables.

To begin with, we examine the bivariate Gaussian distribution with unit variance for each marginal and estimate the *Error* for correlations $\rho = 0.3, 0.5$ and 0.8 and plot the results against the number of samples (N) for each estimator. The leaning rate hyper-parameter used for the 2D Gaussian data is 0.01 for MINE and 0.0001 for InfoNCE and FLO, as provided by the implementations. The results are shown in **Figure 5.1**.

For the correlation value of $\rho = 0.3$, as the number of samples (N) increases, the error decreases consistently for all three estimators and converges towards 0. However, this is not the case for all estimators when ρ increases.

When the correlation value increases to $\rho = 0.5$, we still observe a general trend of decreasing error as the number of samples (N) increases. However, the convergence to zero is less pronounced compared to the case of $\rho = 0.3$. Moreover, the error decrease seems to be less stable in this case especially for MINE. Overall, the FLO estimator performs favorably, producing lower errors compared to MINE and InfoNCE.

The most substantial observation can be made for the correlation value of $\rho = 0.8$. In this case, MINE and InfoNCE even with increasing numbers of samples (N), the error does not exhibit a clear convergence to zero. The errors for these two estimators remain relatively high, indicating a greater difficulty in accurately estimating the mutual information with a high correlation coefficient. Despite this observation for $\rho = 0.8$, the FLO estimator still indicates superior performance.

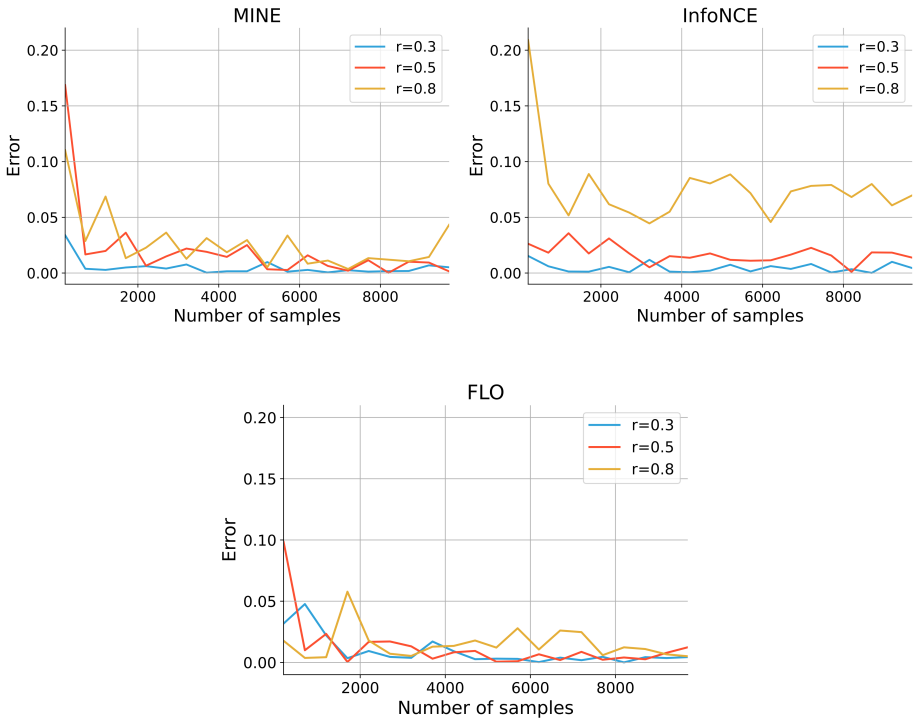

Figure 5.1: Errors of estimate for three different correlations (r) of the 2-D Gaussian data against sample size (N), where number of epochs is fixed at 100. The FLO estimator outperforms both MINE and InfoNCE for all three correlation by providing decreasing errors that converge to 0 as N increases.

Additionally, for the same 2-dimensional variables the resulting plots of the estima-

tions against the varying level of correlation $\rho \in [0, 0.9]$ are shown in **Figure 5.2**. In this case, we trained each estimator using 100 epochs and the baseline hyper-parameter values and kept the number of samples to $N = 10000$. The resulting **Figure 5.2** shows how each deep learning MI estimator perform as the correlation between X and Y increases. In order to evaluate the accuracy of each estimator we additionally include the values of the ground truth MI.

The FLO estimator consistently produces estimates that closely align with the baseline ground truth MI across the entire range of correlation values. Its estimates exhibit a high level of accuracy and are almost identical to the ground truth. This indicate that the FLO estimator can accurately capture the mutual information in the given 2D Gaussian dataset, which align with the finding in [20].

The MINE estimator also performs well, demonstrating accurate estimations, especially for low correlation values. However, it is important to note that as the correlation coefficient ρ exceeds 0.75, a very slight margin of error becomes noticeable in its estimates. This observation can also be captured in Figure 1 of [3]. Nonetheless, it overall performs well in estimating the MI for the dataset.

The InfoNCE estimator's performance indeed relies on the number of negative samples (K) used during the estimation process. As shown in **Figure 5.2** we demonstrated the performance of InfoNCE when $K = 5, 10$ and 20. It is clearly observed that as the value of K increases, the estimates provided by InfoNCE become increasingly aligned with the ground truth MI. Therefore, it is crucial to select an adequately high value of K to ensure accurate predictions from the estimator. However, it is important to note that even with an optimal value of K, the InfoNCE estimator may exhibit slightly higher estimation errors compared to FLO and MINE, particularly for higher correlation.

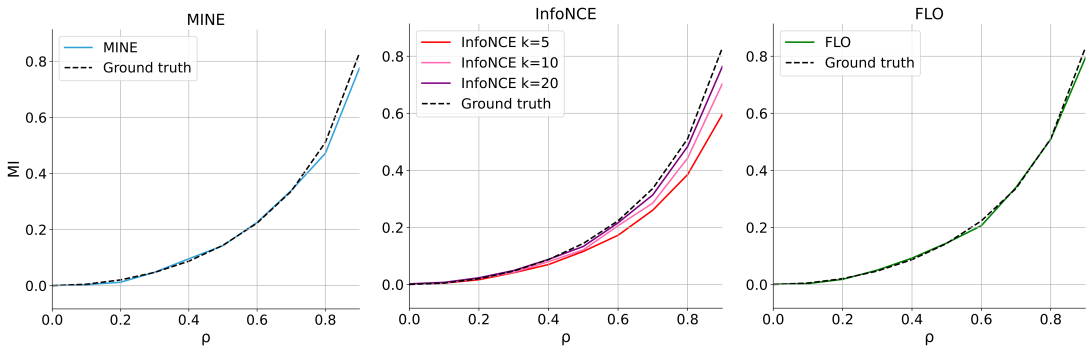

Figure 5.2: Mutual information between the 2-D Gaussian dataset with component-wise correlation $\rho \in [0, 0.9]$. The number of epochs is fixed at 100. The most recently proposed estimator FLO provides tighter estimations.

5.1.2 20D Gaussian variables

Proceeding with the more challenging 20-dimensional Gaussian distribution dataset, previous research [20] indicates that increasing the dimensionality of Gaussian variables leads to less accurate estimations for InfoNCE and FLO, particularly as their correlation value increases. However, FLO outperforms InfoNCE under these conditions. In contrast, MINE has been demonstrated to efficiently scale up to higher dimensionality and consistently provide accurate estimations across all correlation values [3]. However, it has been observed to demonstrate slower convergence when compared to other proposed estimators [7, 56].

As the estimators' behavior changes with the increase of dimensionality, we aim to further investigate and compare InfoNCE, FLO, and MINE in the context of 20D Gaussian variables with varying correlation values and different values for the hyper-parameters: learning rate and number of epochs.

First, an ablation study was conducted, as shown in the **Figure 5.3**. The aim was to investigate the impact of two very crucial hyper-parameters, the learning rate and the number of epochs on the performance of each deep learning MI estimators in achieving estimations close to the ground truth value calculated using the **Formula 4.1** . This investigation was carried out across three different correlation settings, $\rho = 0.2, 0.5$ and 0.8, in order to discern the contributions of the different hyper-parameters combinations to the performance of the estimator when the correlation between the 10D X and 10D Y increases.

It can be observed from the results that the MINE estimator overall performs best when the learning rate has a higher value, in this case 0.001, This indicates that MINE benefits from a larger step size during the optimization process. Moreover, in comparison to InfoNCE and FLO, MINE exhibits slower convergence to the ground truth MI estimation. It is evident from the **Figure 5.3** that MINE may requires a higher number of epochs to achieve accurate estimations which aligns with previous findings [7, 56].

As for the InfoNCE estimator, the number of negative samples K is set to 20, based on observations from the 2D Gaussian case, where $K = 20$ yielded more accurate estimations. The grid analysis **Figure 5.3** highlights the strong dependence of InfoNCE's performance on the correlation between X and Y. It shows that different correlations require different learning rates to achieve accurate estimations. As the correlation increases, a higher learning rate is necessary to obtain MI predictions closer to the

ground truth. Moreover, the figure indicates that 60 epochs are generally sufficient for InfoNCE to reach stable estimations.

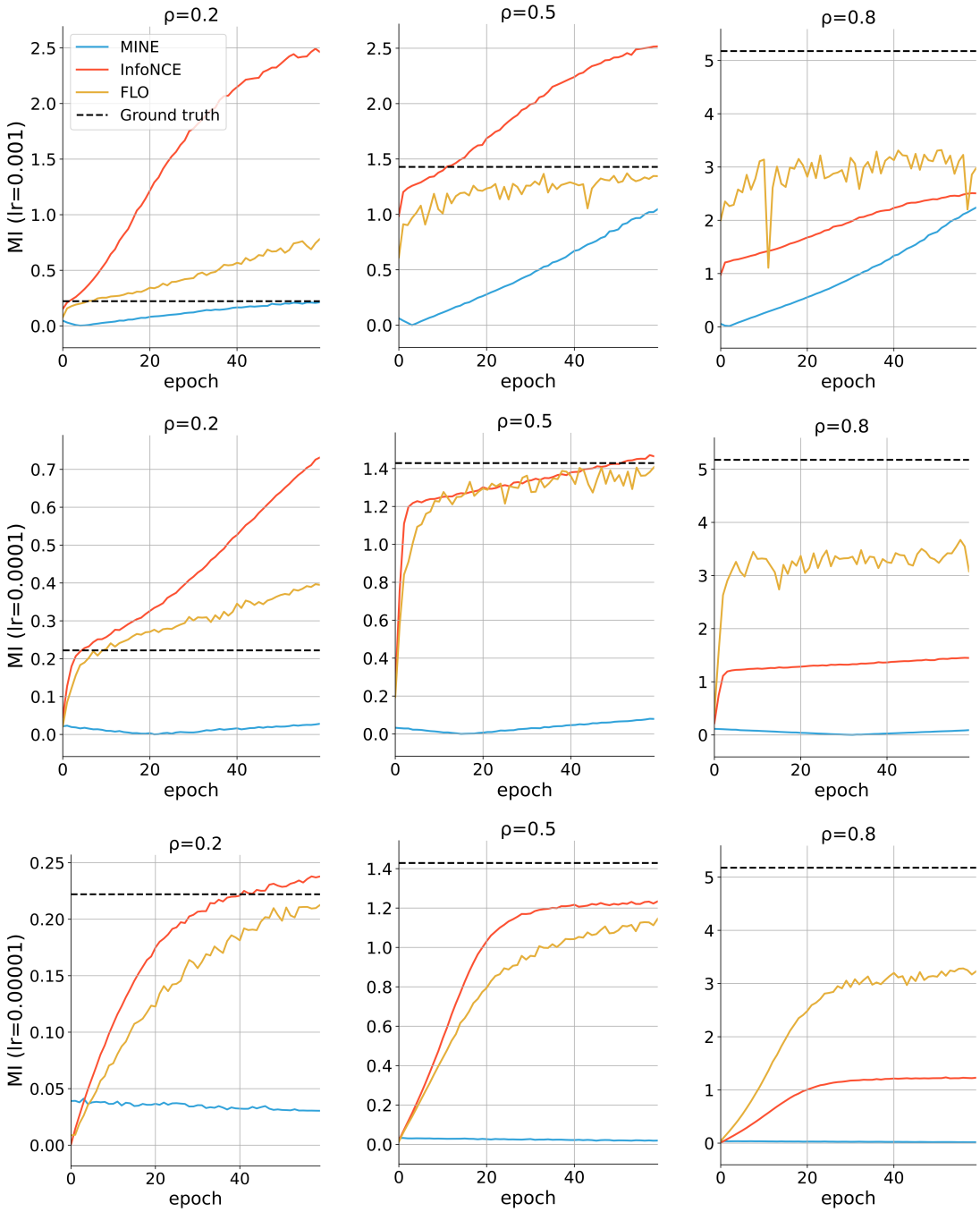

Figure 5.3: Ablation study for network complexity of each MI estimator for the 20-dimensional Gaussian variables generated using different correlations $\rho = 0.2, 0.5$ and 0.8. The hyper-parameters that change are the learning rate (0.001, 0.0001 and 0.00001) and the number of epochs (0 to 60).

Similar to InfoNCE, the FLO estimator achieves more accurate estimations by utilising different learning rate values for different correlations between X and Y. For a lower correlation like $\rho = 0.2$ a lower learning rate works best (for example $lr = 0.00001$). Conversely, for a higher correlation like $\rho = 0.5$, a learning rate of $lr = 0.0001$ performs better. Moreover, compared to MINE and InfoNCE, the FLO estimator demonstrates the highest convergence speed.

Lastly, it is important to highlight that there are notable differences in the running times of each estimator. Specifically, MINE requires approximately 10 times less time than InfoNCE and 5 times less time than FLO.

To further evaluate the impact of learning rate values on the estimators' performance, we present a plot in **Figure 5.4**, showcasing their estimations using their respective best-performing learning rates across varying correlation values ($\rho \in [0, 0.9]$). The results corroborate the observations made during the grid analysis. MINE consistently performs well with a learning rate of 0.001 and an increased number of epochs, irrespective of the correlation between X and Y. These findings align with the conclusions presented in [3].

On the other hand, InfoNCE and FLO indeed provide more accurate estimations for lower correlation values when a lower learning rate is employed. Conversely, for higher correlation values, a higher learning rate yields improved results. Notably, FLO demonstrates superior performance when compared to InfoNCE, which aligns with the initial findings reported in the original paper introducing the FLO estimator.

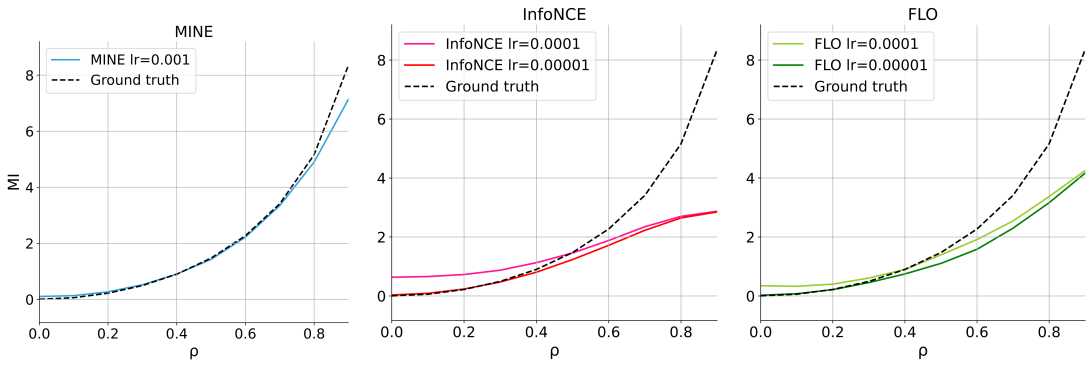

Figure 5.4: Mutual information between the 20-D Gaussian dataset with component-wise correlation $\rho \in [0, 0.9]$. For MINE the number of epochs is set at 180, while for FLO and InfoNCE is at 60. For higher dimensions, MINE scale up best and provides better accuracy as the correlation increases.

5.2 Simulated dataset

In this section, we undertake an evaluation and comparison of the three deep learning estimators using the simulated data generated in **Section 4.2.2**. These simulated data follow a multivariate Gaussian distribution which is transformed to possess very similar properties and correlations as the real neuroscience data (**Section 4.1**). Due to their Gaussian properties, we can still calculate the ground truth MI using the **Formula 4.1**.

Based on our previous analysis results and findings on relevant papers we shape the following hypothesis. First, due to low correlation values between the variables that are observed in the correlation matrix (**Section 4.2.2**), we expect that both InfoNCE and FLO estimators will perform better when using a lower learning rate. Conversely, for the MINE estimator, we hypothesize that it will demonstrate improved performance with a larger learning rate based on our findings in **Section 5.1.2**.

Additionally, extending on our previous analysis, we predict that FLO will outperform InfoNCE in the same experimental settings. Moreover, similar to the Gaussian cases, we anticipate that more epochs will be required for MINE to achieve better accuracy and convergence towards the ground truth MI. MINE's slower convergence is a known characteristic, and extending the training process with more epochs is likely to improve its accuracy in estimating the mutual information.

Lastly, based on [44], it is noted that due to the different critic architecture within the neural network between MINE and InfoNCE/FLO, it is possible that InfoNCE and FLO require a larger neural network to achieve comparable or superior results to MINE. By considering these hypotheses, we start by conducting a comprehensive grid search for three crucial hyper-parameters; number of epochs, learning rate, and the number of hidden layers. The results of this grid search are presented in **Figure 5.5**.

The top three sub-figures illustrate the performance of MINE in estimating the MI of the simulated dataset across a range of 200 epochs, utilising various combinations of learning rates and numbers of hidden layers. Subsequently, the following three sub-figures display the results for InfoNCE, and the final three sub-figures pertain to the FLO estimator across a range of 100 epochs.

The MINE estimator demonstrates optimal performance when its neural structure is relatively shallow, consisting of 2 or 3 layers, and when the learning rate is set to a larger value, specifically lr = 0.01. To validate the most effective layer structure, the Mean Absolute Error (MAE) was computed after running each configuration five times. The 3 hidden layer structure exhibited instability, resulting in NaN values. In contrast,

the 2-layer structure with lr=0.01 yielded a MAE of approximately 0.03979.

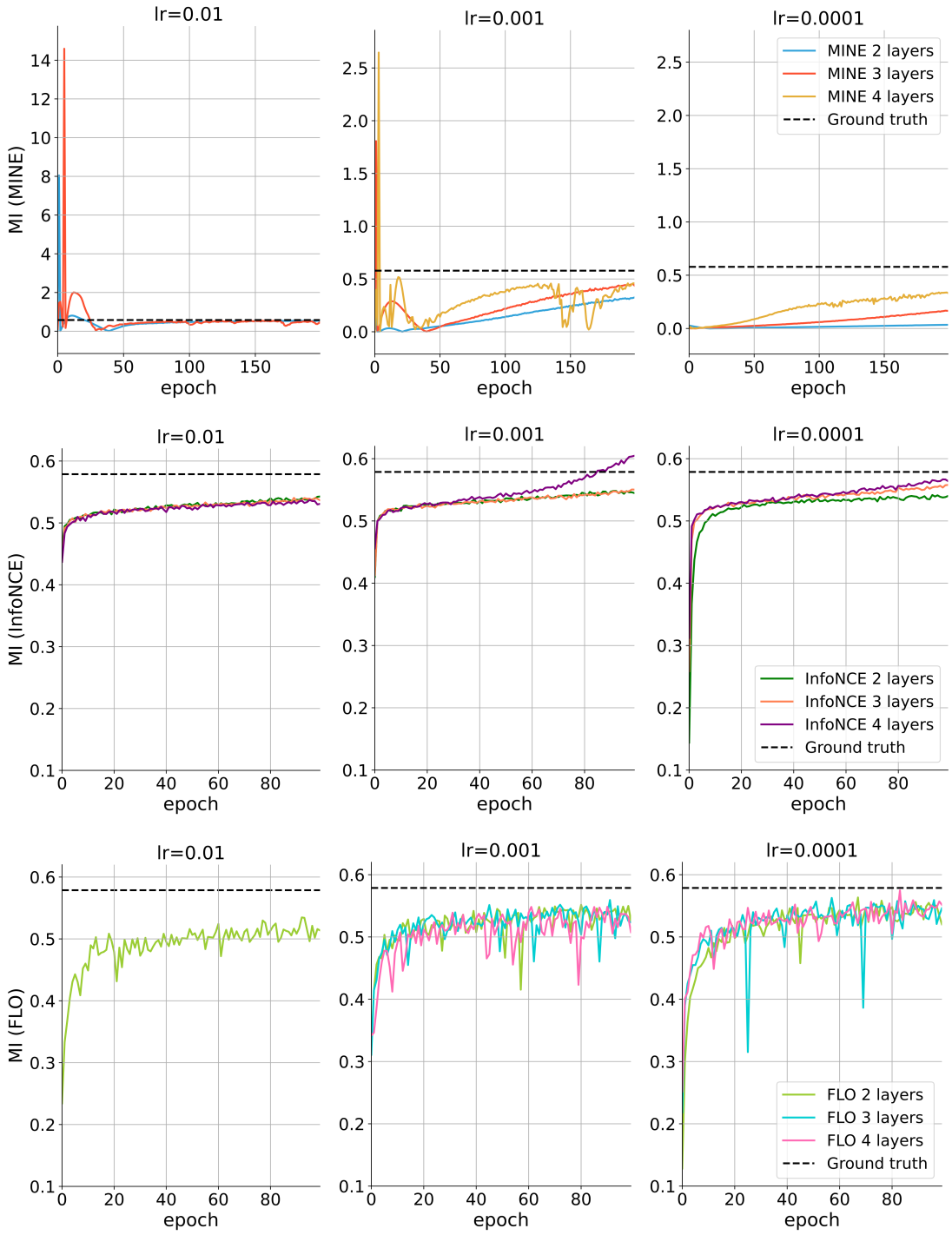

Figure 5.5: Grid search of the hyper-parameter tuning including different values for learning rate (lr = 0.01, 0.001 and 0.0001), number of epochs (range from 0 to 100) and number of hidden layers (2, 3 and 4 layers) incorporated in the DNN structure of the estimators. Any missing combinations did not result in viable estimations.

The InfoNCE estimator achieves higher accuracy as the learning rate decreases and more hidden layers are introduced to its neural structure. By analyzing **Figure 5.6 (B)**, which illustrates the error over the number of epochs for the best performing hyper-parameter combinations, the most promising combination for InfoNCE consists of a neural structure with 4 hidden layers and lr = 0.0001. This yields the most accurate estimations, as confirmed by the MAE, which is calculated for all four combinations, resulting in a value of less than 0.01 (**Table 5.1**).

The FLO estimator converges toward the ground truth mutual information, but its progress across epochs can be considered somewhat unstable. This is evidenced by sudden drops in the plots. The best performing hyper-parameter combinations for FLO all have a learning rate of 0.0001, with the 3 hidden layer configuration yielding the lowest MAE (**Table 5.1**).

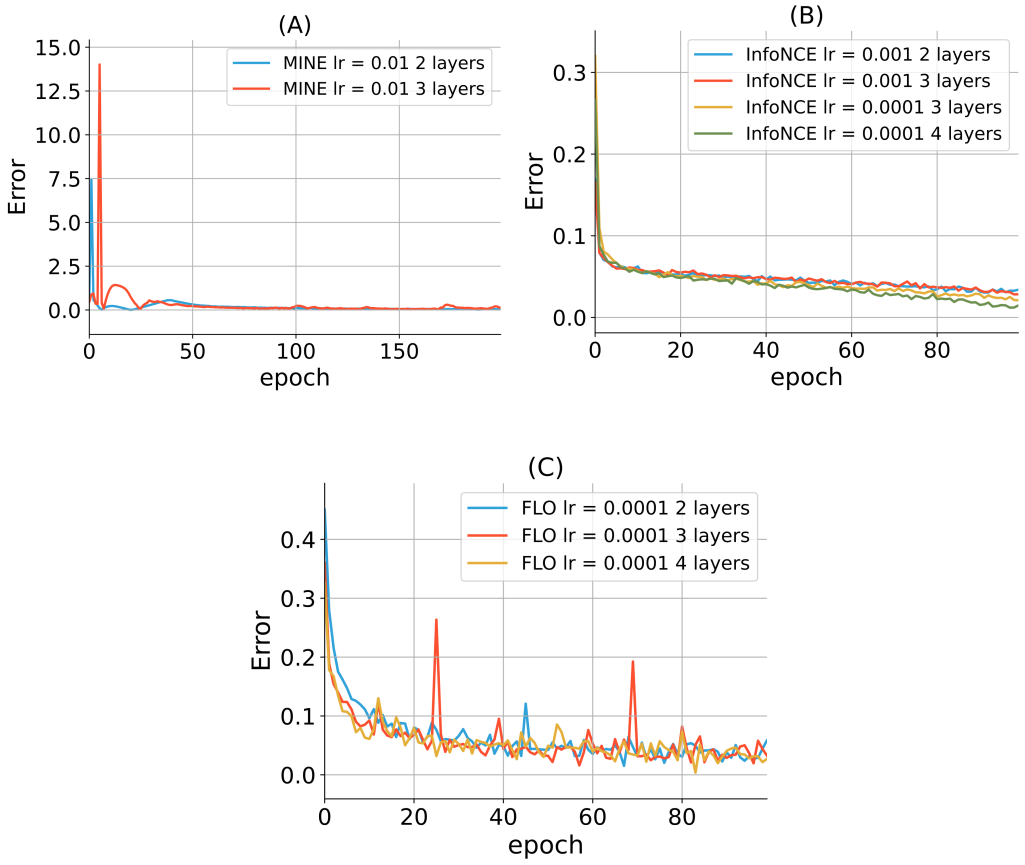

Figure 5.6: Isolated plots of the Error of the best performing hyper-parameter combinations of learning rate (lr) and number of hidden layers for (A) MINE, (B) InfoNCE and (C) FLO estimators over the number of epochs.

An important factor to also consider is the running time of each estimator. MINE

achieves an accuracy of 0.04 within a few minutes, whereas FLO requires five times more time to reach the same accuracy. This difference can be attributed to the deep neural structure and architecture of separable critics in FLO. The best combination for InfoNCE takes approximately over an hour to complete, which is considerably longer than the other two estimators, but it offers five times less error.

Estimator	Learning rate	Number of Layers	Number of epochs	Time	Mean Absolute Error
MINE	0.01	2 Layers	200	$\approx 3'$	0.039788872
	0.01	3 Layers	200	$\approx 6'$	nan
InfoNCE (K=20)	0.001	2 Layers	100	$\approx 40'$	0.029277305
	0.001	3 Layers	100	$\approx 45'$	0.027629111
	0.0001	3 Layers	100	$\approx 50'$	0.023780812
	0.0001	4 Layers	100	$\approx 80'$	0.008717371
FLO	0.0001	2 Layers	100	$\approx 12'$	0.049237186
	0.0001	3 Layers	100	$\approx 16'$	0.037322811
	0.0001	4 Layers	100	$\approx 25'$	0.044616963

Table 5.1: Best Performing Hyper-parameter Combinations and resulting Mean Absolute Error (MAE) for each of the three Deep Learning MI Estimators in the simulated neuroscience framework.

From the prior analysis conducted on a simple Gaussian dataset, it was found that setting $K = 20$ for InfoNCE yielded accurate estimations with low errors. Similarly, the current results obtained for the simulated data so far also indicated that this value of K for the negative samples led to highly accurate estimations. Nevertheless, to make sure that this value of negative sample K is the optimal one for the InfoNCE estimator within this experimental framework, we proceeded to plot the estimations of InfoNCE for different values of K, specifically $K = 5, 10, 20$ and 40.

Analyzing **Figure 5.7**, it becomes evident that as the value of K increases, the estimations tend to converge more closely towards the ground truth value over the course of epochs. However, when K is set to a larger value, such as $K = 40$ as illustrated in the plot, the estimations surpass the ground truth value once the number of epochs reaches 100. This observation aligns with the findings presented in [63], which emphasize that while theoretically, increasing K leads to improved estimator performance, in practical scenarios such as in neuroscience, the data may be noisy and therefore, cause excessive negative examples resulting in misleading gradients. Consequently, higher values of K result in estimations that do not converge towards the ground truth MI value.

Overall, setting K=20 seems to be a good choice for estimating the MI between the simple Gaussian data, as seen in our above analysis, and especially between the neuro-

science simulated data. Applying InfoNCE with $K = 20$ on the simulated neuroscience data has resulted in estimations very close to the ground truth, yielding the lowest Mean Absolute error compared to MINE and FLO as seen in **Table 5.1**.

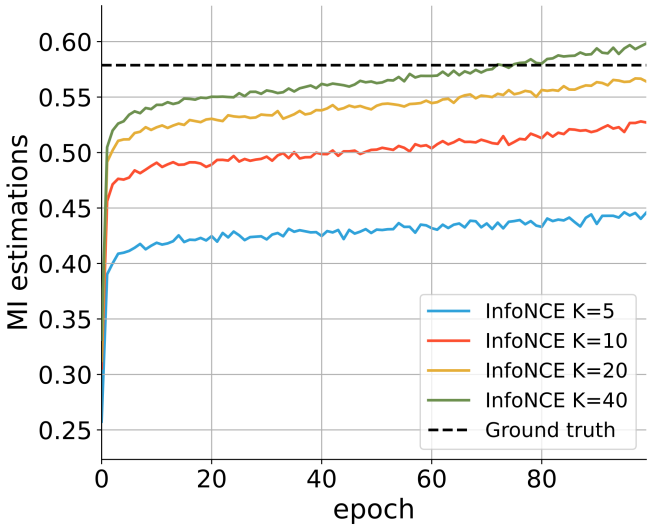

Figure 5.7: Estimation progress over the number of epochs for four different values of the negative sample parameter $K = 5, 10, 20$ and 40 of best hyper-parameter combination of InfoNCE.

In **Table 5.2**, we present the results after conducting a last calculation of the Mean Absolute Error (MAE) obtained from the one hyper-parameter combination of each estimator, which demonstrated the best performance. This time, the MAE was determined based on 10 runs of each estimator. Notably, the InfoNCE and FLO estimators exhibit improved performance when employing a larger network structure and fewer epochs compared to the MINE estimator. Additionally, the MINE estimator requires a substantially larger learning rate, whereas the InfoNCE and FLO estimators require a much smaller one.

Prior to applying these estimators to the simulated data, we initially expected FLO to outperform InfoNCE as we have seen with the simple Gaussian datasets [20]. However, contrary to our expectations, this was not the case in this practical scenario. It is evident that the InfoNCE estimator performs exceptionally well when applied to the simulated dataset, which contains properties similar to those found in real neuroscience data. Overall, all three estimators appear to perform well, but InfoNCE demonstrates significantly lower error rates, approximately four times lower than FLO and six times lower than MINE (**Table 5.2**). Furthermore, InfoNCE exhibits greater stability in its

estimation progress over the number of epochs, with a smoother reduction in error towards zero, as shown in **Figure 5.6 (B)**.

Estimator	Learning rate	Number of Layers	Number of epochs	Time	Mean Absolute Error
MINE	0.01	2 Layers	200	$\approx 3'$	0.035544258
InfoNCE (K=20)	0.0001	4 Layers	100	$\approx 90'$	0.006853117
FLO	0.0001	3 Layers	100	$\approx 35'$	0.025924197

Table 5.2: Best Performing hyper-parameter combination of each estimator, MINE, InfoNCE and FLO and the resulting Mean Absolute Error (MAE) of running each 10 times on the simulated neuroscience dataset.

However, it is important to highlight that the dependence of InfoNCE on the negative samples K introduces an additional parameter that requires regulation whose value significantly influences the estimator's capacity to deliver accurate MI estimations. While setting K=20 appears promising within this framework and in the context of the simple Gaussian data application, it should be recognized that this choice might not be optimal when applying InfoNCE to other neuroscience datasets characterized by diverse metrics and dimensionality.

5.3 Neuroscience application

Through our analysis of simulated neuroscience data, we have concluded that InfoNCE, after utilising specific hyper-parameter values such as learning rate, number of epochs, and number of hidden layers in its DNN architecture, can be deemed the most effective deep learning MI estimator for neuroscience data when compared to MINE and FLO.

Subsequently, we applied the InfoNCE estimator to the real neuroscience data as mentioned in **Section 4.1**, comprising the original Speed variable (X) and five neuron recordings (Y). The calculated estimation of the mutual information between the speed variable and these five neurons is as follows:

$$I(X,Y)_{InfoNCE} \approx 0.70777611.$$

It should be noted that the time required for InfoNCE to finalize this estimation was approximately 110 minutes.

Furthermore, estimators MINE and FLO have been applied to the same dataset in order to discern disparities in their respective estimations:

$$I(X,Y)_{MINE} \approx 0.42232007$$

$$I(X,Y)_{FLO} \approx 0.59827593.$$

Observing these outcomes, it becomes evident that there are differences among the estimations produced by each estimator. Considering InfoNCE's estimation as the approximation closest to the true MI value, it is revealed that FLO yields the second most accurate estimation, whereas MINE provides an estimation deviating the most from the true value. These outcomes are consistent with the findings derived from our comprehensive analysis conducted on the simulated neuroscience dataset.

Chapter 6

Conclusions

In this paper we aim to assess and compare the performance of three recently proposed deep learning MI estimators: MINE, InfoNCE and FLO, when applied on common neuroscience data.

In the beginning, we demonstrated their performance on simple 2D and 20D Gaussian variables. Building on prior studies [3, 20], we compared them across various correlation values and sample sizes of the 2D Gaussian variables. Our findings have shown that for simple 2D Gaussian variables all three estimators provide accurate estimations. However, the estimations of FLO are tighter and closer to the ground truth values which align with the findings of [20].

When evaluating the 20D Gaussian variables, our research indicated that the FLO estimator outperformed InfoNCE in higher dimensional spaces, particularly as the correlation increased. Similar findings were also highlighted in [20]. On the other hand, MINE showcased consistent and accurate estimations across all correlation values in this context as shown in [3]. However, despite its accuracy, MINE exhibited slower convergence compared to other estimators. This is due to MINE'S network limitation that fails to learn at the initial training phase as explained in [7]. For this setting, a detailed analysis of the hyper-parameters; learning rate and number of epochs, revealed that hyper-parameter tuning is essential for retrieving accurate results from each estimator. This analysis indicated which hyper-parameter values work better for each estimator helping us make more targeted choices in the subsequent simulated dataset framework.

Our most important findings are around the application of the three estimators on the simulated neuroscience data. In this case we explored the performance of the estimators in conditions resembling real-world neuroscience recordings but for

which the ground truth MI could still be calculated. We performed an extensive hyper-parameter tuning of the learning rate, the number of hidden layers, and the number of epochs, which are the most influential in the performance of a DNN [30, 37, 53]. Contrary to expectations, InfoNCE displayed exceptional performance on the simulated dataset when its best performing hyper-parameter values were used (4 hidden layers, 100 epochs and lr=0.0001). Despite its additional reliance on the negative sample parameter K, which needs to be determined, InfoNCE consistently delivered accurate MI estimations, outperforming both FLO and MINE in terms of accuracy and stability.

Our comprehensive evaluation highlights the significance of selecting appropriate hyper-parameter values for each estimator in various scenarios. While FLO and MINE were expected to perform best as they had indicated under the simple Gaussian settings, InfoNCE exhibited remarkable stability and precision on the simulated neuroscience data. This suggests that InfoNCE is an optimal choice for estimating the MI of real-world neuroscience data.

Nonetheless, our study showcases several limitations that need consideration. We analyzed and compared the three estimators using a singular simulated neuroscience dataset. However, to confirm the superior performance of InfoNCE within neuroscience data, it is necessary to assess its applicability on more simulated neuroscience datasets encompassing varying dimensions and sample sizes.

Furthermore, InfoNCE's dependence on the negative sample parameter K, a value that profoundly impacts the estimator's capacity to provide precise estimations, and its increased computational time are two important drawbacks. Given the diversity of neuroscience datasets, particularly those arising from extensive experimental settings, the determination of an optimal K value becomes more complex. As pointed in [63], real data, including neuroscience [58], may be noisy resulting in the inclusion of excessive negative examples for which a higher value of K may provide counter results. Therefore, not a single K value can work for all cases.

Additionally, neuroscience experiments simultaneously capture recordings from various different neurons [58], resulting in the need for the quantification of MI between numerous variables. As observed in our analysis, the time necessary for InfoNCE to generate results between a few variables is already substantially increased. Thus, in a real-world problem, each estimation may require a tremendous amount of time.

Overall, having these in mind, potentially a different estimator like FLO which is much faster, fairly accurate and independent of the negative sample parameter K be a better choice.

Collectively, these findings should motivate further in-depth investigations into the applicability of InfoNCE within the domain of neuroscience, accompanied by comprehensive comparison against the FLO estimator whose performance was also noteworthy. Future works could also extend the comparative analysis, by additionally including more proposed deep learning estimators into the framework of neuroscience research.

Bibliography

[1] S. Baillet, L. Garnero, G. Marin, and J.P. Hugonin. Combined meg and eeg source imaging by minimization of mutual information. *IEEE transactions on biomedical engineering*, 46(5):522–534, 1999.

[2] D. Barber and F. V Agakov. The im algorithm : A variational approach to information maximization. *NIPS*, 16, 2003.

[3] M.I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. *Proceedings of the 35th International Conference on Machine Learning*, pages 531–540, 2018.

[4] A. Borst and F.E. Theunissen. Information theory and neural coding. *Nature Neuroscience*, 2:947–957, 1999.

[5] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40, 2009.

[6] T. Catuogno, M. R. Camara, and M. Secondini. Non-parametric estimation of mutual information with application to nonlinear optical fibers. *IEEE International Symposium on Information Theory (ISIT)*, pages 736–740, 2018.

[7] Chung Chan, Ali Al-Bashabsheh, Hingpang Huang, Michael Lim, Da Sun Handason Tam, and Chao Zhao. Neural entropic estimation: A faster path to mutual information estimation. *ArXiv*, 2019.

[8] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell. Introduction to machine learning, neural networks, and deep learning. *Translational vision science technology*, 92, 2020.

[9] T. Cover and J. Thomas. Elements of information theory. *2nd edition Hoboken N. J.: Wiley-Interscience*, 2006.

[10] P. Czyż, F. Grabowski, J.E. Vogt, N. Beerenwinkel, and A. Marx. Beyond normal: On the evaluation of mutual information estimators. *arXiv preprint arXiv:2306.11078*, 2023.

[11] G.A. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

[12] S. Van de Geer. Empirical processes in m-estimation. *Neural NetworksCambridge University Press*, 2000.

[13] A.G. Dimitrov, A.A. Lazar, and J.D Victor. Information theory in neuroscience. *Journal of computational neuroscience*, 30:1–5, 2011.

[14] M. Donsker and S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. *Communications on Pure and Applied Mathematics*, 36, 1983.

[15] G. Doquire and M. Verleysen. A comparison of multivariate mutual information estimators for feature selection. *ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, 1:176–185, 2012.

[16] F. Effenberger. A primer on information theory with applications to neuroscience. *Computational Medicine in Data Mining and Modeling*, 2013.

[17] G.D. Fischbach. Mind and brain. *Scientific American*, 267:48–57, 1992.

[18] A.M. Fraser and H.L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2), 1986.

[19] V. Gómez-Verdejo, M. Martínez-Ramón, J. Florensa-Vila, and A. Oliviero. Analysis of fmri time series with mutual information. *Medical image analysis*, 16:451–458, 2012.

[20] Q. Guo, J. Chen, D. Wang, Y. Yang, X. Deng, J. Huang, L. Carin, F. Li, and C. Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *Advances in Neural Information Processing Systems*, 35:28319–28334, 2022.

[21] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *AISTATS*, 2010.

[22] C.R. Harris, K.J. Millman, S.J. van der Walt, and et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[23] J.U. Henschke, E. Dylda, D. Katsanevaki, N. Dupuy, S.P. Currie, T. Amvrosiadis, J. MP. Pakan, and N.L. Rochefort. Reward association enhances stimulus-specific representations in primary visual cortex. *Current Biology*, 30(10):1866–1880, 2020.

[24] J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. 4:IV–317–IV–320, 2007.

[25] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[26] R. A.A. Ince, B.L. Giordano, C. Kayser, G.A. Rousselet, J. Gross, and P.G. Schyns. A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38:1541–1573, 2017.

[27] G. James, D. Witten, T. Hastie, and R. Tibshirani. An introduction to statistical learning: With applications in r. *New York: Springer*, 2013.

[28] J. Jeong, J.C. Gore, and B.S. Peterson. Mutual information analysis of the eeg in patients with alzheimer's disease. *Clinical neurophysiology*, 112(5):827–835, 2001.

[29] J.J Jun, N.A. Steinmetz, J.H. Siegle, and D.J. Denman nd M. Bauza et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

[30] Z. S. Kadhim, H. S. Abdullah, and K. I. Ghathwan. Artificial neural network hyperparameters optimization: A survey. 18:59–87, 2022.

[31] J.B. Kinney and G.S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, pages 3354–3359, 2014.

[32] A. Kraskov, H. Stogbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69:066138, 2004.

[33] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86, 1951.

[34] L. Liao, H. Li, W. Shang, and L. Ma. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. 31, 2022.

[35] M.D McDonnell, S. Ikeda, and J.H. Manton. An introductory review of information theory in the context of computational neuroscience. *Biological Cybernetics*, 105:55–70, 2011.

[36] J. Mölter and GJ. Goodhill. Limitations to estimating mutual information in large neural populations. *Entropy*, 22, 2020.

[37] S. Nematzadeh, F. Kiani, M. Torkamanian-Afshar, and N. Aydin. Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. *Computational Biology and Chemistry*, 97, 2022.

[38] J.V. Neumann. The computer and the brain. *Yale University Press*, second edition, 2000.

[39] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56:5847–5861, 2010.

[40] A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[41] J. M P Pakan, S. P. Currie, L. Fischer, and N. L. Rochefort. The impact of visual cues, reward, and motor feedback on the representation of behaviorally relevant spatial locations in primary visual cortex. *Cell reports*, 24:2521–2528, 2018.

[42] S. E. Palmer, O. Marre, M. J. Berry, and W. Bialek. Predictive information in a sensory population. *Proceedings of the National Academy of Sciences*, 112(22):6908–6913, 2015.

[43] A. Paszke, S. Gross, F. Massa, and A. Lerer et al. Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035, 2019.

[44] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. *Proceedings of the 36th International Conference on Machine Learning*, 97:5171–5180, 2019.

[45] L. Qiang. Functional connectivity inference from fmri data using multivariate information measures. *Neural Networks*, 146:85–97, 2022.

[46] R. Quian Quiroga and S. Panzeri. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, 10(3):173–185, 2009.

[47] A. Rhee, R. Cheong, and A. Levchenko. The application of information theory to biochemical signaling systems. *Physical biology*, 9(4):045011, 2012.

[48] F. Rieke, D. Warland, R.R. de Ruyter van Steveninck, and W. Bialek. Spikes: Exploring the neural code. *MA: MIT Press, Cambridge*, 1997.

[49] C.J. Rozell and D.H. Johnson. Examining methods for estimating mutual information in spiking neural systems. *Neurocomputing*, 65:429–434, 2005.

[50] JT Russell. Imaging calcium signals in vivo: a powerful tool in physiology and pharmacology. *Br J Pharmacol*, pages 1605–1625, 2011.

[51] M. Scanziani and M. Häusser. Electrophysiology in the age of light. *Nature*, 461(7266):930–939, 2009.

[52] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[53] A. Sharma, J. N. van Rijn, f. Hutter, and A. Müller. Hyperparameter importance for image classification by residual neural networks. In *Discovery Science*, pages 112–126. Springer International Publishing, 2019.

[54] W.E. Skaggs, B.L. Mcnaughton, K.M Gothard, and E.J. Markus. An information-theoretic approach to deciphering the hippocampal code. *Advances in neural information processing systems*, 5, 1993.

[55] W.E. Skaggs, B.L. McNaughton, M.A. Wilson, and C.A. Barnes. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6(2):149–172, 1996.

[56] J. Song and S. Ermon. Understanding the limitations of variational mutual information estimators. 2019.

[57] B.C. Souza, R. Pavão, H. Belchior, and A. BL. Tort. On information metrics for spatial coding. *Neuroscience*, 375:62–73, 2018.

[58] N.M. Timme and C. Lapish. A tutorial for information theory in neuroscience. *eNeuro*, 2018.

[59] N. Veyrat-Charvillon and FX Standaert. Mutual information analysis: How, when and why? *Cryptographic Hardware and Embedded Systems - CHES*, pages 429–443, 2009.

[60] J.D. Victor. Approaches to information-theoretic analysis of neural activity. *Biological theory*, 1:302–316, 2006.

[61] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, and et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[62] J. Williams and Y. Li. Estimation of mutual information: A survey. *Proc. of the 4th Int. Conf. on Rough Sets and Knowledge Technology*, pages 389–396, 2009.

[63] C. Wu, F. Wu, and Y. Huang. Rethinking infonce: How many negative samples do you need? 2021.

[64] M. Zbili and S. Rama. A quick and easy way to estimate entropy and mutual information for neuroscience. *Front Neuroinform*, 2021.

Appendix A

Proofs

A.1 Proof of MINE estimator's consistency

The two problems in which consistency is divided (Approximation and Estimation) lead to two important Lemmas that justify that MINE is strongly consistent. Lemma A.1.1 states that the neural information estimates as defined in Definition 2.5.1.1 can accurately approximate mutual information. Lemma A.1.2 proves that with increasing samples, MINE almost surely converges to a neural information measure.

Lemma A.1.1. *(Approximation) Given an arbitrary positive value* ε*, there exists a neural network with parameters* θ *within a compact domain* $\Theta \in \mathbb{R}^k$ *that parametrizes functions* T_θ*, satisfying the following condition,*

$$|I(X,Y) - I_\Theta(X,Y)| \leq \varepsilon, \quad a.e.$$

Lemma A.1.2. *(Estimation) For a given positive value* ε*, considering a family of neural network functions* T_θ *parametrized by* θ *within a bounded domain* $\Theta \in \mathbb{R}^k$*, there exists an* $N \in \mathbb{N}$ *such that*

$$\forall n \geq N, \quad \left| \widehat{I(X;Y)}_n - I_\Theta(X,Y) \right| \leq \varepsilon, \quad a.e.$$

A.2 Proof of relation between InfoNCE and MINE

By setting $f(x,y) = e^{F(x,y)}$ to InfoNCE's formula (Definition 2.5.2.3) we get a result equivalent to the MINE estimator,

$$\begin{aligned}
\mathop{\mathbb{E}}_{X}\left[log\frac{f(x,y)}{\sum_{x_j\in X} f(x_j,y)}\right] &= \mathop{\mathbb{E}}_{(x,y)}[F(x,y)] - \mathop{\mathbb{E}}_{(x,y)}\left[log\sum_{x_j\in X} e^{F(x_j,y)}\right] \\
&= \mathop{\mathbb{E}}_{(x,y)}[F(x,y)] - \mathop{\mathbb{E}}_{(x,y)}\left[log\left(e^{F(x,y)} + \sum_{x_j\in X_{neg}} e^{F(x_j,y)}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{(x,y)}[F(x,y)] - \mathop{\mathbb{E}}_{y}\left[log\sum_{x_j\in X_{neg}} e^{F(x_j,y)}\right] \\
&= \mathop{\mathbb{E}}_{(x,y)}[F(x,y)] - \mathop{\mathbb{E}}_{y}\left[log\frac{1}{K-1}\sum_{x_j\in X_{neg}} e^{F(x_j,y)} + log(K-1)\right]
\end{aligned}$$

A.3 Proof of bivariate Gaussian MI formula

Let $(X,Y) \sim \mathcal{N}(0,\Sigma)$ be two Gaussian variables with correlation ρ, where

$$\Sigma = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}$$

Then, the entropy of both X and Y is $H(X) = H(Y) = \frac{1}{2}log(2\pi e)\sigma^2$ and their joint entropy is $H(X,Y) = \frac{1}{2}log(2\pi e)^2|\Sigma| = \frac{1}{2}log(2\pi e)^2\sigma^4(1-\rho^2)$. Therefore, using the Equation as described in **Definition 2.2.1** we derive [9]

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = -\frac{1}{2}log(1-\rho^2).$$