Advances in Self-Supervised Learning : applications to neuroscience and sample-efficiency

l'Émir Omar Chéhab

Advances in Self-Supervised Learning : applications to neuroscience and sample-efficiency

Avancées en apprentissage auto-supervisé : applications aux neurosciences et efficacité statistique

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : informatique mathématique.
Graduate School : Informatique et sciences du numérique.
Référent : Faculté des sciences d'Orsay.

Thèse préparée dans l'unité de recherche **Inria, Inria Saclay-Île-de-France** (Université Paris-Saclay), sous la direction de **Alexandre GRAMFORT**, Professeur, et le co-encadrement de **Aapo HYVÄRINEN**, Professeur.

Thèse soutenue à Paris-Saclay, le 24 novembre 2023, par

L'émir Omar CHÉHAB

Composition du jury

Membres du jury avec voix délibérative

Nicolas CHOPIN Professeur, ENSAE Paris	Président
Patrick GALLINARI Professeur, Sorbonne Université / Criteo AI	Rapporteur & Examinateur
Michael GUTMANN Professeur associé, University of Edinburgh	Rapporteur & Examinateur
Anna KORBA Professeur assistant, ENSAE Paris	Examinatrice
Lauri PARKKONEN Professeur, Aalto University	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASG079

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Avancées en apprentissage auto-supervisé : applications et efficacité statistique

Mots clés : Estimation contrastive bruitée, échantillonnage préférentiel, apprentissage auto-supervisé

Résumé : L'apprentissage auto-supervisé a gagné en popularité en tant que méthode d'apprentissage à partir de données non annotées. Il s'agit essentiellement de créer puis de résoudre un problème de prédiction qui utilise les données ; par exemple, de retrouver l'ordre de données qui ont été mélangées. Ces dernières années, cette approche a été utilisée avec succès pour entraîner des réseaux de neurones qui extraient des représentations utiles des données, le tout sans aucune annotation. Cependant, notre compréhension de ce qui est appris et de la qualité de cet apprentissage est limitée. Ce document éclaire ces deux aspects de l'apprentissage auto-supervisé.

Empiriquement, nous évaluons ce qui est appris en résolvant des tâches auto-supervisés. Nous spécialisons des tâches de prédiction lorsque les données sont des enregistrements d'activité cérébrale, par magnétoencéphalographie (MEG) ou électroencephalographie (EEG). Ces tâches partagent un objectif commun : reconnaître la structure temporelle dans les ondes cérébrales. Nos résultats montrent que les représentations apprises en résolvant ces tâches-là comprennent des informations neurophysiologiques, cognitives et cliniques, interprétables.

Théoriquement, nous explorons également la question de la qualité de l'appretissage, spécifiquement pour les tâches de prédiction qui peuvent s'écrire comme un problème de classification binaire. Nous poursuivons une trâme de recherche qui utilise des problèmes de classification binaire pour faire de l'inférence statistique, alors que cela peut nécessiter de sacrifier une notion d'efficacité statistique pour une autre notion d'efficacité computationnelle. Nos contributions visent à améliorer l'efficacité statistique. Nous analysons théoriquement l'erreur d'estimation statistique et trouvons des situations lorsque qu'elle peut rigoureusement être réduite. Spécifiquement, nous caractérisons des hyperparametres optimaux de la tâche de classification binaire et prouvons également que la populaire heuristique de "recuit" peut rendre l'estimation plus efficace, même en grandes dimensions.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)
université PARIS-SACLAY

Title : Advances in Self-Supervised Learning : applications and sample-efficiency

Keywords : Noise-Contrastive Estimation, Importance Sampling, Self-Supervised Learning

Abstract : Self-supervised learning has gained popularity as a method for learning from unlabeled data. Essentially, it involves creating and then solving a prediction task using the data, such as reordering shuffled data. In recent years, this approach has been successful in training neural networks to learn useful representations from data, without any labels. However, our understanding of what is actually being learned and how well it is learned is still somewhat limited. This document contributes to our understanding of self-supervised learning in these two key aspects.

Empirically, we address the question of what is learned. We design prediction tasks specifically tailored to learning from brain recordings with magnetoencephalography (MEG) or electroencephalography (EEG). These prediction tasks share a common objective : recognizing temporal structure within the brain data. Our results show that representations learnt by solving these tasks contain interpretable cognitive and clinical neurophysiological features.

Theoretically, we explore the quality of the learning procedure. Our focus is on a specific category of prediction tasks : binary classification. We extend prior research that has highlighted the utility of binary classification for statistical inference, though it may involve trading off some measure of statistical efficiency for another measure of computational efficiency. Our contributions aim to improve statistical efficiency. We theoretically analyze the statistical estimation error and find situations when it can be provably reduced. Specifically, we characterize optimal hyperparameters of the binary classification task and also prove that the popular heuristic of "annealing" can lead to more efficient estimation, even in high dimensions.

Table des matières

1 - Introduction

This thesis manuscript explores the topic of self-supervised learning, from practical applications to cognitive and clinical neuroscience, to its statistical theory.

1.1 . Self-Supervised Learning

Data collection The starting point of a learning algorithm is the dataset of observations it is given to process. A dataset is made of many $\boldsymbol{x} \in \mathbb{R}^D$ referred to as data points or inputs. The entries of the vectors are called features or covariates. Sometimes, a data point is paired with an annotation $\boldsymbol{y}$, also known as a response variable, output or target. Statistical theory assumes that the dataset has a certain structure that is described by a probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$. Pairs of data points and annotations are assumed to be independently drawn from that distribution : this is signified by the notation *i.i.d.* for independent and identically distributed draws.

Learning algorithms There are many ways to categorize learning algorithms [1, Section 1.3] : a popular criterion is what a learning algorithm is given to process.

1. A *supervised* learning algorithm processes data points and annotations together $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in [\![1,N]\!]} \overset{\text{iid}}{\sim} p(\boldsymbol{x}, \boldsymbol{y})$.

   The goal is to predict an annotation from its data point. When the annotations are discrete and finite, they are called labels and the supervised learning algorithm is known as *classification*. When the annotations are continuous, the supervised learning algorithm is known as *regression*. The term "supervised" is used because the presence of annotations "guide the learning process" [2].

2. An *unsupervised* learning algorithm processes data points only $(\boldsymbol{x}_i)_{i \in [\![1,N]\!]} \overset{\text{iid}}{\sim} p(\boldsymbol{x})$. There are no observed annotations [2, 3].

   The goal is to describe "associations and patterns" among data points [2]. This is not so much a single properly defined goal, as it is a set of possible objectives that need not be compatible [4, Goals of nonlinear ICA and unsupervised learning].

   For example, typical goals in unsupervised learning may include clustering which consists in grouping similar data points together, or density estimation which consists in approximating the distribution of data points, or representation learning which consists in projecting the data to low-dimensional spaces that can be visualized or are useful in some other way [3]. A popular way for measuring the "usefulness" of representations, is to check whether or not they correlate with a variable of interest from another dataset (*e.g.* if representations of brain activity correlate with age). This can be determined by solving a linear prediction task from the representation to the variable of interest [5] and is common practice in computer vision [6–8].

In recent years, a third category known as *self-supervised* learning (SSL) has emerged. The terminology is originally from robotics and computer vision [9, 10], although hints can almost be found earlier [11].

There is no standard definition yet, but defining traits can be pieced from the literature.

3. A *self-supervised* learning algorithm processes data points and annotations together $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in [\![1,N]\!]} \overset{\text{iid}}{\sim} p(\boldsymbol{x}, \boldsymbol{y})$.

   The goal is also to predict an annotation from its data point.

   So far, self-supervised learning shares the exact same definition as supervised learning. Yet, it is closer to unsupervised learning in two ways.

   First, annotations are not observed ; rather, they are obtained from an annotation sampling process $\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x}_{i \in [\![1,N]\!]})$ that is designed by the user. Different works described this process as obtaining annotations "automatically" [10, 12], without "any explicit effort" [9], "from the data" [12] and "often leveraging the underlying structure in the data" [13], and "without human annotators" [12].

   Second, the supervised learning task is only a "pretext" for an ulterior goal of unsupervised learning. This goal can be clustering [14], density estimation [11, 15], or learning representations that have desirable statistical properties [16–18] or that correlate with a variable of interest from another dataset "downstream".

This leads us to the following definition : *a learning algorithm is self-supervised if it solves a prediction task that is designed by the user and in so doing achieves an ulterior goal of unsupervised learning*. In this manuscript, we will consider different unsupervised learning goals : learning "useful" representations in Part I, and learning parameters of a density in Part II.

When is self-supervised learning useful ? Having described what self-supervised learning is, it is natural to ask why and when it is useful. Some claims have already been made to answer a more specific question : *why is self-supervised learning useful in opposition to supervised learning ?* "As opposed to supervised learning, which is limited by the availability of labeled data, self-supervised approaches can learn from vast unlabeled data" [19]. Or "While traditional supervised learning methods are trained on a specific task often known a priori based on the available labeled data, SSL learns generic representations useful across many tasks." [19] Yet these advantages of self-supervised learning are certainly also true of unsupervised tasks that are not prediction-based.

A question remains : *why is self-supervised learning useful in opposition to other unsupervised learning algorithms that are not prediction-based ?* Anticipating the following sections, we can start teasing out an answer for a certain self-supervised learning algorithms. Specifically, in Part II we will theoretically study self-supervised algorithms where the ulterior unsupervised goal is to estimate the parameters of a density, which is known as parametric statistical inference. This comes with a trade-off between computational efficiency and statistical efficiency (defined in Section 1.3.1). On one end of the spectrum, Maximum-Likelihood Estimation (MLE) is an unsupervised learning algorithm that is not self-supervised (it does not solve an explicit prediction task). It is known to be statistically efficient but we will see that it can be computationally inefficient (Section 1.3.2). On the other end of the spectrum, Noise-Contrastive Estimation (defined in Section 1.2.2) is a self-supervised learning algorithm that estimates parameters of a density by solving a binary classification task designed by the user. We will see that it comes with a computational advantage (Section 1.3.2) that is paid for by a statistical error

that can be exponentially large in the dimensionality of the data [20]. This trade-off between computational and statistical efficiency is one way to frame the divide between self-supervised learning and other unsupervised learning algorithms.

Equipped with a formal definition of what self-supervised learning is, we will next attempt to understand it through intuitive examples.

1.1.1 . Examples of regression tasks

Many regression tasks in self-supervised learning consist in predicting one part of the data (the output) based on another (the input). Typically, each part is obtained by applying a mask to the data, as illustrated in Figure 1.1. Different choices of masks recover well-established tasks in statistical machine learning.

Applying a deterministic mask to both input and target future values leads to *forecasting*, which consists in predicting the future of a time series based on past values. This has been successfully applied to Natural Language Processing (NLP) by predicting the next word in a sentence [21].

Applying a random mask that deletes parts of the input only, leads to missing value *imputation*, which consists in predicting the values removed by the mask. This has been successfully applied to NLP by predicting missing words in a sentence [22], or to images by predicting missing patches which is known as "inpainting" [23].

Applying a random mask which alters the input only, leads to *denoising*, which consists in predicting the original data from the noisy version. This has been successfully applied to feature extraction from images [24] or density estimation [25].

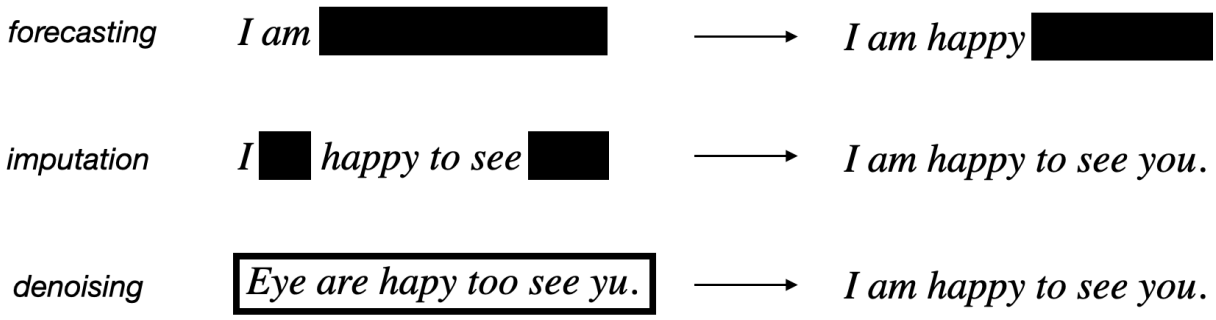


Figure 1.1 – Examples of regression tasks that are self-supervised. A mask (in black) is applied to the data.

One could imagine many more examples of regression tasks for self-supervised learning. Before moving to a more mathematical understanding in following sections, we suggest an intuitive explanation for these prediction tasks : what they have in common is altering the structure of the original data (input) and learning to reconstruct it (output). The idea is that the statistical correlations inside the data are indicative of structure : by altering the correlations, for example by changing a word in a sentence, the sentence does not "make sense" as it used to. By predicting the original data from the altered data, the bet that is made is that whatever is learnt will have to contain necessary statistical information for reconstructing the correlations in the original data ; so whatever is learnt would

therefore be meaningful in some statistical sense.

1.1.2 . Examples of classification tasks

Classification tasks in self-supervised learning consist in assigning labels to different parts of the data. We first consider binary labels $\{0, 1\}$, here referred to as "negative" and "positive" classes in the terminology of self-supervised learning.

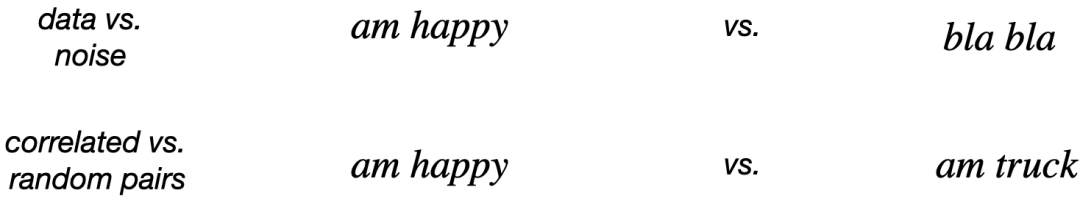


Figure 1.2 – Examples of binary classification tasks that are self-supervised.

A simple task is when the positive class refers to data, and the negative class refers to noise generated from a distribution chosen by the user [11].

A related approach is to form pairs of nearby *versus* random parts of data, and then classify them. This has been successfully applied to timeseries [16, 26], where pairs of windows that are close in time are distinguished from pairs of randomly picked windows; or to images, where pairs of patches that are close in space are distinguished from pairs of patches that are randomly picked from that image or even another [27].

We may also consider multi-class classification problems, where there are more than two labels. When the label designates a permutation of the original order of the data, "re-ordering" the data can be formulated as a prediction task. This has been applied to images [28].

While the above may appear as a broad catalogue of different tasks, this is exactly how self-supervised learning has emerged in recent literature. To our salvation, most methods by definition fall into two well-established categories of supervised learning : classification and regression. In this manuscript, we will study self-supervised learning when it is formulated as a binary classification task.

1.2 . Binary Classification

Earlier, we defined self-supervised learning as a prediction task where the annotation creating process is designed by the user. To understand what self-supervised learning actually *learns*, we must study these prediction tasks, the most basic of which is binary classification.

1.2.1 . Binary Classification as ratio-matching

Given a sample of data points and labels $(\boldsymbol{x}, y) \sim p(\boldsymbol{x}, y)$, a binary classification task consists in learning to predict a label $y \in \{0, 1\}$ from its data point $\boldsymbol{x} \in \mathbb{R}^D$. From a probabilistic viewpoint, this means learning a model $f(\boldsymbol{x}) \in [0, 1]$ of the class-predictive probability $p(y = 1|\boldsymbol{x}) \in [0, 1]$ [1].

Classification losses for the class-predictive probability A learning algorithm for $p(y = 1|\boldsymbol{x})$ is often obtained by minimizing a loss. It is common to use the logistic loss, defined as

$$\mathcal{L}_{\text{logistic}}(f) := -\mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\big[y \log f(\boldsymbol{x}) + (1 - y) \log(1 - f(\boldsymbol{x}))\big] \tag{1.1}$$

which is indeed uniquely minimized by $p(y = 1|\boldsymbol{x})$. In fact, we may consider a generalization of the logistic loss, by broadening our scope to *any* loss that is also written as an expectation and has the same minimizer — such losses are known as strictly proper [29, 30] or well-calibrated [31, Th. 16] in classification theory. They are explicitly characterized by a Bregman divergence between the model $f(\boldsymbol{x})$ and true $p(y = 1|\boldsymbol{x})$ class-predictive probability [31]

$$\mathcal{L}_{\text{Bregman}}(f; \phi_1) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\big[\mathcal{D}_{\phi_1}(p(y = 1|\boldsymbol{x}), f(\boldsymbol{x}))\big] \ . \tag{1.2}$$

Simply put, a Bregman divergence $\mathcal{D}$ between two points $a, b \in \mathbb{R}$ measures how far apart they are [32] : the divergence is zero when the points are equal and positive otherwise. Its geometrical properties are determined by a convex function ϕ which formally defines the divergence as

$$\mathcal{D}_{\phi}(a, b) = \phi(a) - \phi(b) - \phi^{'}(b)(a - b) \ . \tag{1.3}$$

For example, the special case of the $\phi(x) = x^2$ recovers the well-known Euclidean distance $(a - b)^2$. Another special case $\phi(x) = x \log(x) - (1 + x) \log((1 + x)/2)$ recovers the logistic loss in Eq. 1.1. Finally, the family of classification losses in Eq. 1.2 is described by the choice of a convex function ϕ_1 [31, Corollary 5].

From the class-predictive probability to a density ratio The class-predictive probability we wish to learn deserves further attention. Using Bayes' rule, it can be rewritten as

$$p(y = 1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = 1)p(y = 1)}{p(\boldsymbol{x}|y = 1)p(y = 1) + p(\boldsymbol{x}|y = 0)p(y = 0)} = \psi\Big(\frac{p_1}{\nu p_0}(\boldsymbol{x})\Big) \ , \tag{1.4}$$

1. In the binary setting, the probability of predicting the other class is determined by the normalization of the class-predictive distribution : $p(y = 0|\boldsymbol{x}) + p(y = 1|\boldsymbol{x}) = 1$.

so as to map to the ratio of class-conditional probabilities $p_1(\boldsymbol{x}) := p(\boldsymbol{x}|y=1)$ and $p_0(\boldsymbol{x}) := p(\boldsymbol{x}|y=0)$, reweighed by the prior-ratio of the two classes $\nu := p(y=0)/p(y=1)$. The mapping from the class-predictive probability to the density-ratio is one-to-one and is defined by $\psi(x) = 1/(1+x)$. In fact, this is the main argument relating binary classification to unsupervised learning : *learning the class-predictive probability effectively means learning (a ratio of) densities*.

Classification losses for the density-ratio Based on the identity Eq. 1.4, we may now redefine the learning problem in terms of the true (left) and model (right) density-ratio, defined as

$$r^*(\boldsymbol{x}) = \psi^{-1}(p(y=1|\boldsymbol{x})) \qquad\qquad r(\boldsymbol{x}) = \psi^{-1}(f(\boldsymbol{x})) \ .$$

The logistic loss for the class-predictive probability Eq. 1.1 can now be written in terms of the density-ratio

$$\mathcal{L}_{\text{logistic}}(r) := -\mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\left[y\log\left(\frac{1}{1+r(\boldsymbol{x})}\right) + (1-y)\log\left(\frac{r(\boldsymbol{x})}{1+r(\boldsymbol{x})}\right)\right] \ . \tag{1.5}$$

It is actually a valid loss for the density-ratio, in that it is minimized by $p_1(\boldsymbol{x})/\nu p_0(\boldsymbol{x})$. The general family of classification losses in Eq. 1.2 can be similarly expressed in terms of the density-ratio, still as a Bregman divergence

$$\mathcal{L}_{\text{Bregman}}(r;\phi_2) = \nu\mathbb{E}_{\boldsymbol{x}\sim p_0(\boldsymbol{x})}\left[\mathcal{D}_{\phi_2}\left(\frac{p_1}{\nu p_0}(\boldsymbol{x}), r(\boldsymbol{x})\right)\right] \tag{1.6}$$

with another convex function $\phi_2(x)$ [33, Proposition 3]. By expanding the integrand using Eq. 1.3, the loss is equivalently rewritten as

$$\mathcal{L}_{\text{Bregman}}(r;\phi_2) = \nu\mathbb{E}_{\boldsymbol{x}\sim p_0(\boldsymbol{x})}\Big[-\phi_2(r(\boldsymbol{x})) + \phi_2^{'}(r(\boldsymbol{x}))\times r(\boldsymbol{x})\Big] - \mathbb{E}_{\boldsymbol{x}\sim p_1(\boldsymbol{x})}\Big[\phi_2^{'}(r(\boldsymbol{x}))\Big] \tag{1.7}$$

up to an additive constant. This expression of the binary classification loss will define the self-supervised learning task in all of Part II in this manuscript.

Binary classification as self-supervised learning The takeaway from this exposition is the equivalence between

- an unsupervised learning problem, specifically learning a *density-ratio* by minimizing common losses (*e.g.* logistic, squared) in Eq. 1.7
- a supervised learning problem, specifically learning the *class-predictive probability* of a binary label by minimizing common losses (*e.g.* logistic, squared) in Eq. 1.2

This fits our definition of self-supervised learning from section 1.1 : a binary classification task can be used toward the unsupervised goal of learning (a ratio of) densities. The next section will provide examples where learning a density-ratio is useful.

1.2.2 . Application to statistical inference

In the previous section, binary classification is formulated as an estimation method for the ratio of the class-conditional distributions p_0 and p_1, from their samples. Using a parametric model of the ratio is interesting : it allows us to infer parameters of these distributions ; this way, binary classification can be used for statistical inference. In this section, we will see that estimating parameters from a ratio of distributions *e.g.* using binary classification, avoids common computational bottlenecks of estimating parameters from a single distribution *e.g.* using maximum-likelihood. The methods covered in this section are summarized in table 1.1.

Statistical inference with neural networks Statistical inference consists in estimating from data the distribution that generated them [34, Chapter 6]. It is common to assume that this distribution is part of a statistical model, that is a family of probability distributions that are identified by a parameter

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))}{Z(\boldsymbol{\theta})} \tag{1.8}$$

where $E(\boldsymbol{x};\boldsymbol{\theta})$ is the energy functional and $Z(\boldsymbol{\theta})$ is a normalizing factor

$$Z(\boldsymbol{\theta}) := \int \exp(-E(\boldsymbol{x};\boldsymbol{\theta}))d\boldsymbol{x} \ . \tag{1.9}$$

The task is then to infer the correct parameter $\boldsymbol{\theta}^*$ from data. Some statistical models have tractable energy functionals and normalizing constants : for example, a Gaussian distribution has a quadratic energy $E(\boldsymbol{x};(\boldsymbol{\mu},\boldsymbol{\Sigma})) = (\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})/2$ and the integral defining the normalizing constant can be analytically solved : $Z(\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}$. More complex energy functionals define more expressive statistical models : in modern applications, they can be parameterized using a deep neural network with weights $\boldsymbol{\theta}$. This way, a statistical model can benefit from the approximation capabilities of neural networks [35]. Yet, this comes at a heavy price : the energy functional or the normalizing factor can lead to important computational bottlenecks. These bottlenecks are specifically two quantities — the integral or the Jacobian of a neural network — that are expensive or intractable to compute, yet are necessary to evaluate a distribution from the statistical model.

One strategy to circumvent these computational bottlenecks is by engineering the neural network architectures so that the problematic quantities become easy to compute : this had led to developing specific neural networks architectures called "normalizing flows" [36] or to use numerical tricks that avoid computing the full Jacobian matrix [37].

Another strategy to circumvent these computational bottlenecks is to choose a ratio of distributions where the problematic quantities disappear, and then to estimate parameters from that ratio using binary classification, instead of from a single distribution using maximum-likelihood. This is the route taken by substantial literature on energy-based models [38, 11, 39, 18]. How exactly these ratios of distributions are chosen is the object of the following paragraphs. Importantly, estimating parameters using binary classification opens the way for neural networks to be used for statistical inference.

Computational bottleneck in the normalizing factor One way to parameterize a statistical model with a neural network, is to define the energy functional directly by a real-valued neural network $g(\boldsymbol{x};\boldsymbol{\theta})$

$$E(\boldsymbol{x};\boldsymbol{\theta}) = g(\boldsymbol{x};\boldsymbol{\theta}) \qquad\qquad Z(\boldsymbol{\theta}) = \int \exp(-g(\boldsymbol{x};\boldsymbol{\theta}))d\boldsymbol{x} \ . \tag{1.10}$$

In this case, the computational bottleneck is in the right hand side of Eq. 1.10 : the normalizing constant is an intractable integral. Numerical methods such as quadrature can approximate the integral with a given precision, but their computational cost can be exponential in the dimensionality of the data [40]. This has motivated a number of methods to estimate the parameters $\boldsymbol{\theta}$ of the statistical model without having to compute the normalizing constant. The general idea is to build a ratio model where the computationally challenging term disappears.

Conditional Noise-Contrastive Estimation and Score-Matching. For example, the following ratio can be used to cancel out the intractable normalizing factor

$$r(\boldsymbol{x},\tilde{\boldsymbol{x}};\boldsymbol{\theta}) := \frac{p(\boldsymbol{x};\boldsymbol{\theta})p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{p(\tilde{\boldsymbol{x}};\boldsymbol{\theta})p_n(\boldsymbol{x}|\tilde{\boldsymbol{x}})} = \frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta})) \times Z(\boldsymbol{\theta}) \times p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{\exp(-E(\tilde{\boldsymbol{x}};\boldsymbol{\theta})) \times Z(\boldsymbol{\theta}) \times p_n(\boldsymbol{x}|\tilde{\boldsymbol{x}})} \tag{1.11}$$

$$= \frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{\exp(-E(\tilde{\boldsymbol{x}};\boldsymbol{\theta}))p_n(\boldsymbol{x}|\tilde{\boldsymbol{x}})} \ . \tag{1.12}$$

Here, $p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ is a tractable proposal distribution of our choosing, that randomly "noises" or "augments" a point $\boldsymbol{x}$ into $\tilde{\boldsymbol{x}}$ (think of modifying an image by adding random Gaussian noise or applying a random rotation). Computing this ratio does not require the normalizing constant. Evaluating this ratio at the correct parameter $\boldsymbol{\theta}^*$ can be done by solving a binary classification task called Conditional Noise-Contrastive Estimation [39], named thusly because the proposal distribution is conditional $p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x})$. For this binary task, the class-conditional distributions are

$$p_1(\boldsymbol{x},\tilde{\boldsymbol{x}}) = p(\boldsymbol{x};\boldsymbol{\theta}^*)p_n(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \qquad\qquad p_0(\boldsymbol{x},\tilde{\boldsymbol{x}}) = p(\tilde{\boldsymbol{x}};\boldsymbol{\theta}^*)p_n(\boldsymbol{x}|\tilde{\boldsymbol{x}}) \ .$$

A pair from the first class is made of a point $\boldsymbol{x}$ from the data distribution and its perturbation $\tilde{\boldsymbol{x}}$ by the proposal distribution p_n. The second class reverses the ordering. When the proposal distribution is chosen to be Gaussian with infinitesimal variance, this recovers another estimation method, Score-Matching [41], as a special case [39, Eq.14].

Noise-Contrastive Estimation. In fact, we may not even need to cancel out the intractable normalizing factor. Consider the following ratio model

$$r(\boldsymbol{x};\boldsymbol{\theta}) := \frac{p(\boldsymbol{x};\boldsymbol{\theta})}{p_n(\boldsymbol{x})} = \frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))}{p_n(\boldsymbol{x}) \times Z(\boldsymbol{\theta})} \tag{1.13}$$

where $p_n(\boldsymbol{x})$ again denotes a tractable proposal distribution of our choosing. Here, the factor $Z(\boldsymbol{\theta})$ ensures that $p(\boldsymbol{x};\boldsymbol{\theta})$ is a normalized density for any parameter $\boldsymbol{\theta}$ encountered *during* a learning algorithm. In theory, we care only that density be normalized when the learning algorithm *terminates* at the correct parameter $\boldsymbol{\theta}^*$. This suggests using the ratio model

$$r(\boldsymbol{x};\boldsymbol{\theta},Z) := \frac{p(\boldsymbol{x};\boldsymbol{\theta},Z)}{p_n(\boldsymbol{x})} = \frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))}{p_n(\boldsymbol{x}) \times Z} \tag{1.14}$$

where the dependency of the factor on the parameters is dropped. Z is now an additional parameter of an unnormalized density $p(\boldsymbol{x};\boldsymbol{\theta},Z)$. In this ratio model Eq. 1.14, the factor Z need not be known : it is estimated from samples alongside the parameters. When the learning algorithm terminates, $r(\boldsymbol{x};\boldsymbol{\theta},Z) \approx r(\boldsymbol{x};\boldsymbol{\theta}^*,Z(\boldsymbol{\theta}^*))$ which implies $\boldsymbol{\theta} \approx \boldsymbol{\theta}^*$ and $Z^* \approx Z(\boldsymbol{\theta}^*)$ under some identifiability conditions. Note that recent work has begun to explore identifiability conditions for different parameterizations of ratios [42, 18, 43] and using different learning procedures [11, 44]. This particular ratio can be obtained by solving a binary classification task called Noise-Contrastive Estimation (NCE) [11] where the class-conditional distributions

$$p_1(\boldsymbol{x}) = p(\boldsymbol{x};\boldsymbol{\theta}^*) \qquad\qquad p_0(\boldsymbol{x}) = p_n(\boldsymbol{x})$$

are the data and the proposal distributions. More generally, Gutmann and Hirayama [38] showed that virtually all methods that are currently used to estimate parameters without computing a normalizing constant, are special cases of ratio estimation p_1/p_0 with a certain loss Eq. 1.7. This loss was later interpreted by Menon and Ong [33] as a binary classification loss where the class-conditional distributions are p_0 and p_1.

Computational bottleneck in the energy functional Another way to parameterize a statistical model with a neural network, is to write the energy functional in terms of a bijective neural network $\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta})$ as

$$E(\boldsymbol{x};\boldsymbol{\theta}) = -\log p_s(\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta})) - \log|\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x};\boldsymbol{\theta})| \qquad\qquad Z(\boldsymbol{\theta}) = 1 \ . \tag{1.15}$$

This parameterization corresponds to a statistical model with a latent variable $\boldsymbol{s} = \boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta})$ obtained by a bijective transformation of a data point $\boldsymbol{x}$ and distributed as $\boldsymbol{s} \sim p_s$. It is the basis of many representation learning methods including Independent Component Analysis (ICA) [45]. In this case, the computational bottleneck is in the left hand side of Eq. 1.15 : the energy functional is defined in terms of the Jacobian of the neural network $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x};\boldsymbol{\theta})$, where derivatives are taken with respect to $\boldsymbol{x}$. Evaluating the Jacobian [46] and its gradient with respect to the parameters [47] is computationally costly; the latter can scale cubically $O(D^3)$ with the dimensionality of the data.

Pointwise Mutual Information. The following ratio can be used to cancel out the computationally challenging Jacobian term

$$r(\boldsymbol{x},\tilde{\boldsymbol{x}};\boldsymbol{\theta}) := \frac{p(\boldsymbol{x},\tilde{\boldsymbol{x}};\boldsymbol{\theta})}{p(\boldsymbol{x};\boldsymbol{\theta}) \times p(\tilde{\boldsymbol{x}};\boldsymbol{\theta})} = \frac{p_s(\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{g}(\tilde{\boldsymbol{x}};\boldsymbol{\theta})) \times |\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x};\boldsymbol{\theta})| \times |\boldsymbol{J}_{\boldsymbol{g}}(\tilde{\boldsymbol{x}};\boldsymbol{\theta})|}{p_s(\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta}))|\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x};\boldsymbol{\theta})| \times p_s(\boldsymbol{g}(\tilde{\boldsymbol{x}};\boldsymbol{\theta}))|\boldsymbol{J}_{\boldsymbol{g}}(\tilde{\boldsymbol{x}};\boldsymbol{\theta})|} \tag{1.16}$$

$$= \frac{p_s(\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{g}(\tilde{\boldsymbol{x}};\boldsymbol{\theta}))}{p_s(\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta})) \times p_s(\boldsymbol{g}(\tilde{\boldsymbol{x}};\boldsymbol{\theta}))} \ . \tag{1.17}$$

where $p(\boldsymbol{x},\tilde{\boldsymbol{x}})$ is the probability of co-occurence of two points. This ratio is called the "pointwise mutual information" between the random variables $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$. It can evaluated at the correct parameter $\boldsymbol{\theta}^*$ by solving a binary classification task where the class-conditional distributions

$$p_1(\boldsymbol{x},\tilde{\boldsymbol{x}}) = p(\boldsymbol{x},\tilde{\boldsymbol{x}};\boldsymbol{\theta}^*) \qquad\qquad p_0(\boldsymbol{x},\tilde{\boldsymbol{x}}) = p(\boldsymbol{x};\boldsymbol{\theta}^*) \times p(\tilde{\boldsymbol{x}};\boldsymbol{\theta}^*)$$

are the probabilities of co-occurence (joint distribution) and independent occurence (product of marginals) of the pair $(\boldsymbol{x},\tilde{\boldsymbol{x}})$. This method has been used to learn representations of words [48] and of time series [16].

Extensions to Bayesian Inference From the Bayesian perspective, the distribution of interest is the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$, where the parameters $\boldsymbol{\theta}$ are now viewed as a random variable. Using the Bayes formula, we can write the posterior distribution as

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{\exp(-E(\boldsymbol{\theta};\boldsymbol{x}))}{Z(\boldsymbol{x})} \tag{1.18}$$

where the energy functional is redefined as $E(\boldsymbol{\theta};\boldsymbol{x}) = -\log p(\boldsymbol{x}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$.

Importance Sampling. In traditional Bayesian statistics, the prior $p(\boldsymbol{\theta})$ and the conditional likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ are known which means the energy functional is known as well. There remains only to compute the normalizing factor $Z(\boldsymbol{x})$. This can be achieved by estimating the ratio

$$r(\boldsymbol{\theta},\boldsymbol{x};Z) := \frac{p(\boldsymbol{\theta},\boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{p(\boldsymbol{\theta})} = \frac{\exp(-E(\boldsymbol{\theta};\boldsymbol{x}))}{p(\boldsymbol{\theta}) \times Z(\boldsymbol{x})} \quad . \tag{1.19}$$

which is the pointwise Mutual Information between two random variables $\boldsymbol{\theta}$ and $\boldsymbol{x}$. This ratio can be obtained by solving a binary classification task called Conditional Noise-Contrastive Estimation [42], named thusly because the distribution of interest $p(\boldsymbol{\theta}|\boldsymbol{x})$ is conditional (this method should not be confused with a different task with the same name [39]). For this binary task, the class distributions are

$$p_1(\boldsymbol{\theta},\boldsymbol{x}) = p(\boldsymbol{\theta},\boldsymbol{x}) \qquad\qquad p_0(\boldsymbol{\theta},\boldsymbol{x}) = p(\boldsymbol{\theta})p(\boldsymbol{x}) \quad .$$

In chapter 4, we will justify that solving this classification task is equivalent to computing the normalizing factor Z using a family of importance sampling estimators.

Conditional Noise-Contrastive Estimation In some modern Bayesian statistics, the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ is no longer known : it can be sampled but not evaluated. This is a more realistic description of the setting where data is generated from a black-box simulation. With this constraint, Bayesian Inference is known as Likelihood-Free Inference (LFI) [49], Approximate Bayesian Computation (ABC) [50] or Simulation-Based Inference (SBI) [51]. Importantly, this means the energy functional of the posterior is no longer tractable so computing the posterior means estimating the normalizing constant *as well as* the energy functional. We can do so using the same binary classification task as in the previous paragraph. The only difference with the ratio Eq. 1.19 is that the parameter $\boldsymbol{\theta}$ is estimated now (it is after the semicolon)

$$r(\boldsymbol{x};\boldsymbol{\theta},Z) := \frac{p(\boldsymbol{\theta},\boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{p(\boldsymbol{\theta})} = \frac{\exp(-E(\boldsymbol{\theta};\boldsymbol{x}))}{p(\boldsymbol{\theta}) \times Z} \quad . \tag{1.20}$$

This is known as Likelihood-Free Inference by ratio estimation [49]. This idea of estimating the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ as part of a ratio has inspired subsequent work known as Neural Ratio Estimation (NRE) [52, 53].

Table 1.1 – Ratios can be used to obtain many popular estimators of the parameters $\boldsymbol{\theta}^*$ and normalizing constant $Z(\boldsymbol{\theta}^*)$ of a statistical model.

Name	Class 1	Class 0	Ratio	Estimand
	$p_1(.)$	$p_0(.)$	$r(.)$	
NCE	$p(\boldsymbol{x};\boldsymbol{\theta}^*)$	$p_n(\boldsymbol{x})$	$\frac{p(\boldsymbol{x};\boldsymbol{\theta},Z)}{p_n(\boldsymbol{x})}$	$(\boldsymbol{\theta}, Z)$
Importance Sampling			$\frac{p(\boldsymbol{x};Z)}{p_n(\boldsymbol{x})}$	Z
Conditional NCE	$p(\boldsymbol{x};\boldsymbol{\theta}^*)p_n(\tilde{\boldsymbol{x}}\|\boldsymbol{x})$	$p(\tilde{\boldsymbol{x}};\boldsymbol{\theta}^*)p_n(\boldsymbol{x}\|\tilde{\boldsymbol{x}})$	$\frac{p(\boldsymbol{x};\boldsymbol{\theta})p_n(\tilde{\boldsymbol{x}}\|\boldsymbol{x})}{p(\tilde{\boldsymbol{x}};\boldsymbol{\theta})p_n(\boldsymbol{x}\|\tilde{\boldsymbol{x}})}$	$\boldsymbol{\theta}$
Score-Matching	— same with $p_n(\tilde{\boldsymbol{x}}\|\boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}};\boldsymbol{x},\epsilon\boldsymbol{I})$ —			$\boldsymbol{\theta}$
Likelihood-Free Inference	$p(\boldsymbol{x},\boldsymbol{\theta})$	$p(\boldsymbol{x})p(\boldsymbol{\theta})$	$\frac{p(\boldsymbol{\theta}\|\boldsymbol{x})}{p(\boldsymbol{\theta})}$	$p(\boldsymbol{\theta}\|\boldsymbol{x})$
Pointwise Mutual Information			$\frac{p(\boldsymbol{x},\boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})}$	

1.3 . Estimation Theory

1.3.1 . Estimation error

Estimators defined as minimizers of a loss In the previous section, classification theory provided us with a *population loss* in Eq. 1.7 for learning a ratio. The term "population" refers to quantities that are computed using an infinite sample. We then considered in section 1.2.2 applications where it is useful to identify a ratio model with a parameter $\boldsymbol{\theta} \to r(\boldsymbol{x};\boldsymbol{\theta})$. Minimizing the population loss then provides us with an estimand

$$\boldsymbol{\theta}^* = \arg\min \mathcal{L}(\boldsymbol{\theta}) \tag{1.21}$$

of the correct parameter. In practice, to make the minimization computationally tractable, we resort to a finite-sample version of the loss by replacing expectations with sample averages. This new loss is commonly known as an *empirical loss*, where the term "empirical" refers to quantities that are computed using a finite sample

$$\mathcal{L}_N(\boldsymbol{\theta}) = \frac{\nu}{N_0}\sum_{i=1}^{N_0} -\phi(r(\boldsymbol{x}_i;\boldsymbol{\theta})) + \phi^{'}(r(\boldsymbol{x}_i;\boldsymbol{\theta})) \times r(\boldsymbol{x}_i;\boldsymbol{\theta}) - \frac{1}{N_1}\sum_{j=1}^{N_1}\phi^{'}(r(\boldsymbol{x}_j;\boldsymbol{\theta})) \ . \tag{1.22}$$

Here, N_0 and N_1 are the sample sizes for data from p_0 and p_1 respectively, and $N = N_0 + N_1$ is the total sample budget. Minimizing the empirical loss defines an estimator

$$\hat{\boldsymbol{\theta}}_N(\boldsymbol{x}_{1:N}) = \arg\min \mathcal{L}_N(\boldsymbol{\theta};\boldsymbol{x}_{1:N}) \tag{1.23}$$

of the correct parameter. The estimator is a function of the sample $\boldsymbol{x}_{1:N}$, as shorthand notation for $(\boldsymbol{x}_i)_{i\in[\![1,N]\!]}$. It is equivalently defined by the optimality equation

$$\mathbf{0} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_N(\hat{\boldsymbol{\theta}}_N;\boldsymbol{x}_{1:N}) \ . \tag{1.24}$$

This defines the estimator implicitly. Sometimes, when the loss is "simple enough" so that the minimization is tractable, the estimator can be written explicitly. Under standard technical conditions [54, Th. 5.14], the estimator $\hat{\boldsymbol{\theta}}_N$ is *consistent* : it converges (in probability) to $\boldsymbol{\theta}^*$, mainly because it minimizes a loss which converges (pointwise) to $\mathcal{L}$. Consistency guarantees that the estimator correctly targets the estimand when given more data.

Note that this framework is in fact quite general. Estimating a parameter by minimizing a loss function that is evaluated on a random sample [55, Section 3.2.1], is central to statistical estimation and machine learning. When the loss function is written as a sample-average as in Eq. 1.22 , the relevant theory is called M-estimation [54, Chapter 5] or Empirical Risk Minimization (ERM)[56, Chapter 4]. Moreover, the parameter $\boldsymbol{\theta}$ is determined by the loss function it minimizes [54, Eq. 5.1] : it does not have to identify a statistical model, as is typically the case in classical estimation theory. For instance, the parameter may include the normalizing factor of a statistical model as in Noise-Contrastive Estimation Eq. 1.14.

In the following, we use the abbreviated notation $\mathbb{E}[\hat{\boldsymbol{\theta}}_N]$ for averaging over random samples $\mathbb{E}_{\boldsymbol{x}_{1:N}}[\hat{\boldsymbol{\theta}}_N(\boldsymbol{x}_{1:N})]$.

Measuring the estimation error A natural question is : how does the estimator $\hat{\boldsymbol{\theta}}_N$ differ from the estimand $\boldsymbol{\theta}^*$? This difference is known in the literature as the "estimation error", "sample efficiency" or "statistical complexity". Evaluating that difference can be done in a number of ways that will be familiar to different readerships : looking at the difference in the parameters themselves (top), or through the intermediary of a functional (middle) or a loss function (bottom) of the parameters

$$\mathrm{MSE}_{\text{parameters}} := \mathbb{E}[\|\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}^*\|^2_{\ell^2}] \tag{1.25}$$

$$\mathrm{MSE}_{\text{functional}} := \int \mathbb{E}\Big[\big(f(\boldsymbol{x};\hat{\boldsymbol{\theta}}_N) - f(\boldsymbol{x};\boldsymbol{\theta}^*)\big)^2\Big] d\boldsymbol{x} \tag{1.26}$$

$$\mathrm{ME}_{\text{loss}} := \mathbb{E}[\mathcal{L}(\hat{\boldsymbol{\theta}}_N) - \mathcal{L}(\boldsymbol{\theta}^*)] \ . \tag{1.27}$$

The error in the parameters (top) is usual in parametric statistical inference, where the endgoal is to infer parameters from data. It is known as the parametric Mean-Squared Error (MSE) [34, Eq. 6.6]. The error in a functional (middle) is familiar for non-parametric [2] statistical inference, where the parameters are simply a means of modelling a functional $f(.;\boldsymbol{\theta})$, for example a label-predictor or a probability density. It is known as the integrated non-parametric MSE [57, Eq.4.12]. The error in the loss (bottom) is commonly used in Machine Learning as it can easily by approximated in practice : in Eq. 1.27, the population loss evaluated at $\boldsymbol{\theta}^*$ is constant and the expectation is typically approximated using cross-validation [57, Eq. 5.29]. This distinguishes the error in the loss from other measures of estimation error where strong modelling assumptions are required, such as knowing in advance the population parameters $\boldsymbol{\theta}^*$ in Eq. 1.25 or modelling a functional (a parametric family of densities) in Eq. 1.26. The error in the loss is also known as the excess risk [56, Def. 2.3] and is related to the generalization error [56, Section 6.1]. It is here denoted as the Mean Error (ME). Of the three measures of error, we will focus on the parametric MSE. This choice for the estimation error is usually expanded into two terms that have a clear interpretation

$$\mathrm{MSE}_{\text{parameters}} := \|\boldsymbol{\theta}^* - \mathbb{E}[\hat{\boldsymbol{\theta}}_N]\|^2 + \mathbb{E}[\|\hat{\boldsymbol{\theta}}_N - \mathbb{E}[\hat{\boldsymbol{\theta}}_N]\|^2] \ . \tag{1.28}$$

The second term is the variance which measures the fluctations of the estimator. The first term is the squared bias : it quantifies how the estimator tracks the estimand on average, despite fluctations.

How the estimation error depends on task design Hidden within the estimation error in Eq. 1.25 are dependencies on design variables that are used to evaluate the loss — the sample size N, the dimensionality D of data points, and the configuration of hyperparameters. It is desirable to find cases where the estimation error is a tractable function of these design variables. Sometimes, this requires strong modelling assumptions, such as assuming that the parameter identifies a simple statistical model such as a gaussian family of densities [34, Example 9.11]. Alternatively, finding limits where the estimation error becomes a tractable function of these hyperparameters has a rich history in statistics. For example, the limit of an infinite sample size $N \to \infty$ if often considered in classical statistics [54], the limit of infinite dimensions $D \to \infty$ is often considered in contemporary statistics [58] and combinations of these are even today an active field in deep learning [59]. Finding such

2. the term "non-parametric" does not indicate the absence of parameters : they are simply a means to model a functional. The endgoal is to recover the correct functional.

cases where the estimation error is tractable, allows us to explicitly analyze how the error grows with hyperparameters and how best to choose them. For example, suppose the estimation error scales exponentially with the dimensionality D of the data : this is a common situation known as the "curse of dimensionality". Such an estimation error is fatal in high dimensions and cannot in theory be fixed by the choice of optimization algorithm : tooling with the Adam optimizer [60] in high dimensions which is sometimes a reflex in practical deep learning, will not change the fact that the resulting global minimizer $\hat{\boldsymbol{\theta}}_N$ can be off from the estimand $\boldsymbol{\theta}^*$ by an error that is exponentially large. This suggests a perhaps unconvential approach to deep learning [61], where attention is payed to the choice of loss (so that the global minimizer is sample-efficient) as much as the choice of optimizer. Ideally, studying the estimation error can inform us if a certain configuration of hyperparameters can bring the error down from an exponential growth in the dimensionality to a polynomial growth, which is more acceptable.

The large sample limit from classical statistics, $N \to \infty$ The limit of a large sample will simplify the estimator and its error as a function of the sample size N. In this limit, the estimator which was implicitly defined in Eq. 1.24 can now be explicitly written

$$\hat{\boldsymbol{\theta}}_N = \boldsymbol{\theta}^* + \boldsymbol{\varepsilon}(\boldsymbol{x}_{1:N}) + O(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2) \tag{1.29}$$

where the first-order term

$$\boldsymbol{\varepsilon}(\boldsymbol{x}_{1:N}) = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_N(\boldsymbol{\theta}^*)^{-1} \times \nabla_{\boldsymbol{\theta}} \mathcal{L}_N(\boldsymbol{\theta}^*) \tag{1.30}$$

is an error whose randomness comes from the sample $\boldsymbol{x}_{1:N}$. The error's probability law describes the dispersion of the estimator around the estimand $\boldsymbol{\theta}^*$. This law is asymptotically Gaussian. This can be understood by a simple argument. The random matrix (empirical Hessian) in Eq. 1.30 converges in probability to a constant matrix (population Hessian). Hence, the first-order error $\boldsymbol{\varepsilon}(\boldsymbol{x}_{1:N})$ is equivalent in probability to a constant matrix multiplying a random vector : this is a sum of random variables and is asymptotically Gaussian, using the Central Limit Theorem [54]

$$\boldsymbol{\varepsilon}(\boldsymbol{x}_{1:N}) \sim \mathcal{N}(\mathbf{0}, N^{-1}\boldsymbol{\Sigma}) \ . \tag{1.31}$$

This means the estimator $\hat{\boldsymbol{\theta}}$ is centered at the estimand $\boldsymbol{\theta}^*$ (it is asymptotically unbiased) and is scattered in directions and magnitudes given by the covariance matrix

$$\boldsymbol{\Sigma} := \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*)^{-1} \times \mathrm{Var}_{\boldsymbol{x}_{1:N}}[\nabla_{\boldsymbol{\theta}} \mathcal{L}_N(\boldsymbol{\theta}^*)] \times \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*)^{-1} \tag{1.32}$$

which depends on two terms : the population Hessian, and the covariance matrix of the empirical gradient. For a scalar parameter, the magnitude of the estimation error given by Eq. 1.32 is smaller with

- a smaller variance the gradient of the empirical loss
  This means that for different samples $\boldsymbol{x}_{1:N}$, we have roughly the same gradient near the optimum, so the estimator defined in Eq. 1.24 is "robust" across datasets.

- a higher curvature of the population loss
  This means that the estimator is easier to compute in that the landscape defines the minimum more sharply, which increases the convergence speed of many optimization algorithms for a convex loss [62].

In the following, we suppose the standard technical conditions of van der Vaart [54, Th. 5.23] apply so that the remainder term $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2$ can indeed be written independently of the parameterization, as $o(N^{-1})$.

Finally, writing the estimation error in the limit of a large sample $N \to \infty$ [54, Eq. 5.20] simplifies the dependency on N from 1.25 to

$$\mathrm{MSE}_{\text{parameters}} := \frac{1}{N}\mathrm{trace}(\boldsymbol{\Sigma}) + o\left(\frac{1}{N}\right) \ . \tag{1.33}$$

Note that the trace operator naturally arises from the definition of the parametric MSE in Eq. 1.25. It sums the variances of each component of the estimator $\hat{\boldsymbol{\theta}}$. This scalar quantity will be the centerpiece of the statistical analysis in chapters 4 and 5.

1.3.2 . Computing the estimation error for binary classification

In the previous section, we established that the estimation error, measured by the parametric MSE, is described by a covariance matrix defined in Eq. 1.32.

Asymptotic covariance matrix for maximum-likelihood estimation To build intuition on what that matrix looks like, we first consider an important setup where the parameter identifies a statistical model $\boldsymbol{\theta} \to p(\boldsymbol{x}; \boldsymbol{\theta})$, the population loss is the Kullback-Leiber divergence $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x};\boldsymbol{\theta}^*}}\left[\log \frac{p(\boldsymbol{x};\boldsymbol{\theta}^*)}{p(\boldsymbol{x};\boldsymbol{\theta})}\right]$, and the empirical loss replaces the expectation with a finite sum. Recall that the covariance matrix Eq. 1.32 is computed using the population Hessian and the variance of the empirical gradient. Here, both are equal to the same matrix

$$\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^*) = \boldsymbol{I}_{\text{Fisher}} \qquad\qquad \mathrm{Var}_{\boldsymbol{x}_{1:N}}[\nabla_{\boldsymbol{\theta}}\mathcal{L}_N(\boldsymbol{\theta}^*)] = \boldsymbol{I}_{\text{Fisher}} \ .$$

known as the Fisher Information matrix

$$\boldsymbol{I}_{\text{Fisher}} := \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x};\boldsymbol{\theta}^*)}\left[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}^*) \times \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}^*)^\top\right] \ . \tag{1.34}$$

It follows that the asymptotic covariance matrix from Eq. 1.32 is the inverse Fisher matrix

$$\boldsymbol{\Sigma} = \boldsymbol{I}^{-1}_{\text{Fisher}} \tag{1.35}$$

and the estimation error Eq. 1.33 is

$$\mathrm{MSE}_{\text{parameters}} = \frac{1}{N}\mathrm{trace}(\boldsymbol{I}^{-1}_{\text{Fisher}}) + o\left(\frac{1}{N}\right) \ . \tag{1.36}$$

This is known as the Cramer-Rao lower bound : it is the minimum MSE achievable by an unbiased estimator $\hat{\boldsymbol{\theta}}$ of the correct parameter of a statistical model. Beyond this specific situation, it is still useful to intuitively think of the population Hessian and gradient covariance as being roughly of the same order of magnitude [20].

Asymptotic covariance matrix for binary classification We now consider a setup closer to self-supervised learning. Here, the parameter identifies a ratio model $\boldsymbol{\theta} \to r(\boldsymbol{x};\boldsymbol{\theta})$, the population loss is the binary classification loss defined in Eq. 1.7, and the empirical loss replaces the expectation with a finite sum in Eq. 1.22. Again, we can write the population Hessian and the variance of the empirical gradient, borrowing their formulae from previous works [63, Equation 8]; these formulae were derived for a specific binary classification task but we checked that the proof holds more generally.

$$\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^*) = \boldsymbol{I}_w \qquad\qquad \mathrm{Var}[\nabla_{\boldsymbol{\theta}}\mathcal{L}_N(\boldsymbol{\theta}^*)] = \boldsymbol{I}_v - \big(1+\frac{1}{\nu}\big)\boldsymbol{m}_w\boldsymbol{m}_w^\top \ .$$

Both quantities are expressed in terms of the vector $\boldsymbol{m}_w$ and matrices $\boldsymbol{I}_w$ and $\boldsymbol{I}_v$, which are the reweighted mean and covariances of the parameter-gradient of the log model $\nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*)$

$$\boldsymbol{m}_w = \mathbb{E}_{\boldsymbol{x}\sim p_1}\big[w(\boldsymbol{x})\,\nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*)\big] \tag{1.37}$$

$$\boldsymbol{I}_w = \mathbb{E}_{\boldsymbol{x}\sim p_1}\big[w(\boldsymbol{x})\,\nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*) \times \nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*)^\top\big] \tag{1.38}$$

$$\boldsymbol{I}_v = \mathbb{E}_{\boldsymbol{x}\sim p_1}\big[v(\boldsymbol{x})\,\nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*) \times \nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*)^\top\big] \ . \tag{1.39}$$

Note that the reweighting of points from p_1 is done by

$$w(\boldsymbol{x}) = r(\boldsymbol{x};\boldsymbol{\theta}^*)\phi''(r(\boldsymbol{x};\boldsymbol{\theta}^*)) \tag{1.40}$$

$$v(\boldsymbol{x}) = r(\boldsymbol{x};\boldsymbol{\theta}^*)^2\phi''(r(\boldsymbol{x};\boldsymbol{\theta}^*))^2(1+r(\boldsymbol{x};\boldsymbol{\theta}^*)) \ . \tag{1.41}$$

We can now write the asymptotic covariance matrix

$$\boldsymbol{\Sigma} = \boldsymbol{I}_w^{-1}\Big(\boldsymbol{I}_v - \big(1+\frac{1}{\nu}\big)\boldsymbol{m}_w\boldsymbol{m}_w^\top\Big)\boldsymbol{I}_w^{-1} \tag{1.42}$$

and the estimation error Eq. 1.33 is

$$\mathrm{MSE}_{\mathrm{parameters}} = \frac{1}{N}\mathrm{trace}\Big(\boldsymbol{I}_w^{-1}\big(\boldsymbol{I}_v - \big(1+\frac{1}{\nu}\big)\boldsymbol{m}_w\boldsymbol{m}_w^\top\big)\boldsymbol{I}_w^{-1}\Big) + o\Big(\frac{1}{N}\Big) \ . \tag{1.43}$$

A few simple observations can be made to make sense of these equations. Similar to maximum-likelihood estimation, the main mathematical object is the parameter gradient of the log model. There are two differences here : first, the parameter identifies a ratio model $r(\boldsymbol{x};\boldsymbol{\theta})$, not a statistical model $p(\boldsymbol{x};\boldsymbol{\theta})$; second, the samples are reweighted by functions $w(\boldsymbol{x})$ and $v(\boldsymbol{x})$. Consider for a moment that these differences are erased. Specifically, suppose that ratio model is parameterized by one of the class-conditional distributions $r(\boldsymbol{x};\boldsymbol{\theta}) = p(\boldsymbol{x};\boldsymbol{\theta})/p_0(\boldsymbol{x})$, so that the parameter gradient of the log model is the same as in maximum-likelihood estimation : $\nabla_{\boldsymbol{\theta}} \log r(\boldsymbol{x};\boldsymbol{\theta}^*) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}^*)$. This quantity is known as the Fisher score vector. Also, suppose there is no reweighting : $w(\boldsymbol{x}) = v(\boldsymbol{x}) = 1$. Then, the asymptotic covariance matrices are the *same* for maximum-likelihood estimation (Eq. 1.35) and binary classification (Eq. 1.42). Part II of this manuscript pursues this line of inquiry by finding situations where we can make definitive conclusions about the estimation error in binary classification Eq. 1.43. For example, chapter 5 considers situations where this estimation error's dependency on the dimensionality of the problem is explicit, and chapter 4 considers situations where the optimal hyperparameters can be written explicitly.

Table 1.2 – Comparison of three non-invasive brain imaging technologies (MEG, EEG, and fMRI) in the year 2023. Positive (green) and negative (red) assessments are made by the author.

	MEG	EEG	fMRI
Temporal Resolution	ms	ms	s
Spatial Resolution	mm^3	cm^3	mm^3
Portability	no	yes	no
Cost	million $	thousand $	million $

1.4 . Brain Imaging Data

In this document, self-supervised learning is used on both real and synthetic data. Namely, Part I uses self-supervised learning on human brain recordings, which prompts the fundamental question : *What can we hope to gain from understanding the human brain ?*

In terms of scientific inquiry, recording brain activity provides a window into our cognitive processes, that is, how we think and process information from our environment. From the viewpoint of a machine learning researcher, the brain can be thought of as a biological neural network that simultaneously and efficiently processes different data modalities, such as images, audio, and language. This has been the case historically, where brain science has inspired, however loosely, concepts in Machine Learning [64]. Notable examples include Convolutional Neural Networks (CNN) inspired by the human visual system [65], or Independent Component Analysis (ICA) partially inspired by the fly's visual system [66, 67].

On a more practical level, understanding the brain is relevant in a clinical setting where the goal is to improve a patient's health, for example by monitoring and diagnosing their brain state. Applications include personalized anesthesia, improving sleep, or predicting epileptic seizures, which we will later discuss in more detail.

What does brain activity look like ? The human brain contains an estimated 86 billion neurons that communicate using electrical currents [68]. Imaging technologies allow us to observe the activity generated by populations of thousands of these neurons. At the population level, the average activity of individual neurons firing produces an oscillation that is often categorized by its frequency band [69] and is sometimes associated with a certain brain state [70]. For example, in chapter 2 of this manuscript, we will explore transient oscillations triggered by visual stimuli and slower oscillations in the alpha band (8-12 Hz), which modulate brain responses [71]. In this document, "brain activity" will refer to these oscillations.

How is brain activity measured using imaging technologies ? Imaging technologies do not record brain activity itself : they instead detect signals that serve as proxies for these brain oscillations.

For instance, in functional Magnetic Resonance Imaging (fMRI) [72], the proxy signal is the rapid blood delivery to neural cells, known as the haemodynamic response. In the case of Magnetoencephalography and Electroencephalography (M/EEG) [73, 74], the proxy signals are variations in the magnetic and electric fields near the scalp, produced by active neuronal populations. Because this manuscript uses MEG measurements in chapter 2 and EEG measurements in chapter 3, we now describe these two modalities in more detail.

Using M/EEG for brain science is very much a development of the 20th century. The first recording of the human brain with EEG dates back to the 1930s [75, 76], and was followed by MEG some 40 years later [77, 76]. As their usage became more widespread, M/EEG have led to an exponentially increasing number of publications [78, Fig. 1]. Today, EEG recordings are the preferred modality to detect sleep stages [79] and MEG recordings are precise enough for reconstructing speech a user is exposed to [80]. This warrants a better understanding of how these technologies actually work. The practical setup in M/EEG is to place a headset of sensors over the scalp of a human brain and record variations in the electromagnetic field. It is worth noting that typical magnetic and electric signals from M/EEG are of the order 10^{-13} Tesla and 10^{-4} Volt, respectively. For comparison, these are a billion times smaller than the steady magnetic field at the earth's surface, and a million times smaller than the electric tension of a phone charging in the same room or a city train passing nearby [76]. This makes M/EEG measurements particularly corruptible by environment "noise" : for this reason, the recordings are often conducted in a Faraday cage that shields the room from local variations in the electromagnetic field and may justify a cost in the millions of dollars [81]. For MEG, this is a necessary cost for an already sophiscated device, where sensors are cooled using liquid helium. MEG scanners are not portable : they are scarce and expensive, with an estimated 200 in use worldwide [81]. On the other hand, EEG technology can be portably used outside a shielded room, which makes it a less precise but cheaper option.

Trade-offs between brain imaging technologies Having explained how different brain imaging technologies work, there remains to understand their trade-offs in terms of invasiveness, spatial resolution, temporal resolution, and practicality. These trade-offs are summarized in Table 1.2. Specifically, the main argument for M/EEG [78, Fig. 1] among different brain imaging modalities is the high temporal resolution in the millisecond range that it offers. This precision is suitable for measuring brain oscillations and explains why, among non-invasive imaging technologies, M/EEG have become the indispensable tool to study brain dynamics. However, their spatial resolution is limited by the physical setup of tracking the electromagnetic field only at the surface of the scalp. Assessing the spatial resolution of these methods requires mapping back the scalp measurement unto the three-dimensional brain which is called an "inverse problem". When there is a single, dominant source of neural activity, M/EEG methods can localize it with a precision at a scale of the mm^3 that is on par with fMRI. However, when brain activity originates from different sources, possibly close to each other and with different depths and orientations, the localization error may increase [82]. This is particularly an issue for EEG due to a lateral blurring of the signal caused by different conductivities in the layers of the scalp [76], so the spatial resolution may reach the order of the cm^3 [83]. To overcome this limitation, invasive technologies like Stereoelectroencephalography (sEEG) and Electrocorticography (ECoG) place electrodes beneath the scalp and in/on the brain cortex, ensuring accurate measurements but

at the cost of convenience and safety as there may be a risk of infection [84].

Supervised learning from M/EEG M/EEG techniques record brain activity in real-time, non-invasively, and importantly with a temporal granularity that is fine enough to capture brain oscillations that would go otherwise undetected.

MEG is often used for research, in universities and hospitals, into the human brain's cognitive processes. The market for MEG is expanding [81] : although the scanner costs in the millions [81], the combined temporal and spatial resolution of MEG makes it a prime choice among non-invasive imaging technology for studying cognitive processes. A typical experiment in MEG may consist in stimulating the brain with a visual prompt and measuring the brain response. One way to model cognitive processes in the brain is to predict the brain response from the stimulus (known as "encoding"), or the stimulus from the brain response (known as "decoding") [70]. These constitute supervised tasks in MEG.

EEG is also used for research in cognitive neuroscience, but unlike MEG it is often the preferred imaging modality for clinical neuroscience. EEG headsets can be worn during sleep to monitor brain oscillations such as sleep spindles that are sometimes associated with memorization [85]. They can also be used to track sleep disorders such as apnea and narcolepsy [86]. In a clinical setting, EEG headsets can be used to measure a patient's reactivity to anesthesia, by tracking how the anesthetic drug alters waveforms in the brain that mark the patient's cognitive state. EEG headsets can also be used to screen for neurological pathologies such as epilepsy or dementia [87, 88]. Once the brain activity is recorded with EEG, a typical task is to predict the user's condition — one of five stages of sleep or the presence of a pathology — opening possibilities for personalized prevention and treatment. These constitute supervised tasks in EEG.

While supervised learning is an important part of research using M/EEG data modalities, it requires annotated datasets. Yet it is time-consuming [89] and expensive for experts to manually annotate recorded M/EEG signals. Increasingly, datasets of M/EEG recording are being shared in the public domain but with few annotations. This limits the scope of supervised learning and motivates where unsupervised learning, including self-supervised learning, has its mark to make.

1.5 . Outline and contributions

This manuscript studies self-supervised learning in a broad way. Part I is more practical : it applies self-supervised methods to learn representations from brain imaging data. Part II is more theoretical : it analyzes the estimation error of a prototypical self-supervised learning task called Noise-Contrastive Estimation.

Part I, Applications to Brain Activity The contributions in this part are practical. We apply two self-supervised tasks based on recognizing temporal structure in brain imaging data — regression tasks in chapter 2 and classification tasks in chapter 3 — and interpret what is being learnt.

Part I, Chapter 2 : Regression tasks on MEG data This work studies regression tasks based on recognizing temporal structure of MEG data. These tasks consists in predicting the brain reponse

from two covariates : a visual stimulus and past brain activity ; this is known as "neural encoding" and "forecasting" in the neuroscience and statistics literatures. We highlight the following contribution :

- *This work is among the first to use deep neural networks for forecasting brain activity using MEG data.* Forecasting is traditionally done using linear models ; a few works only have investigated deep models and it remains to be understood what statistical information they capture that linear models do not. We are among the first to methodologically investigate this question. Our analysis showed that our deep model used the interaction between the inputs (stimulus and past brain activity) to modulate the forecast, which linear models did not.

Personal contributions. I wrote the main text and co-wrote the code with co-first author Alexandre Defossez whose contribution to the project was important. Alexandre proposed using permutation feature importance as a tool for interpreting a “black-box" model and helped with the practical part of running experiments with deep neural networks using different hyperparameters. The text was heavily edited by my supervisors and the original idea of combining deep learning and neural decoding with a focus on interpretability was proposed by J.R. King and J.C. Loiseau.

Publication :

O. Chehab*, A. Defossez*, J.C. Loiseau, A. Gramfort, J.R. King. *Deep Recurrent Encoder : A scalable end-to-end network to model brain signals*. Journal of Neurons, Behavior, Data analysis, and Theory, 2022.
(* means shared first-authorship)

Part I, Chapter 3 : Classification tasks on EEG data This work studies a classification tasks based on recognizing temporal structure of EEG data. They consist in predicting if snapshots of brain activity are temporally adjacent or not, or if they are ordered or not. We highlight the following contribution :

- *This work is among the first to apply self-supervised learning to EEG data.* Little prior work existed for using these classification tasks based on temporal ordering on brain imaging data. This meant having to find which design choices worked. For example, through trial-and-error, we came up with a certain parameterization of the classifier which empirically worked and that we analyze in further detail in section 3.5.

Personal contributions. Here, my contributions were more minor : I helped design the experiments and worked on their theoretical analysis using basic simulations. At the time, the theoretical analysis was not mature enough to make it to publication ; it has since been revisited and is included in section 3.5 of this thesis as a starting point for future research.

Publication :

H. Banville, O. Chehab, A. Hyvärinen, D. Engemann, A. Gramfort. *Uncovering the structure of clinical EEG signals with self-supervised learning*. Journal of Neural Engineering, 2021.

Part II, Statistical Analysis The setup here is to infer the parameters or normalizing constant of a statistical model of the data. This comes with a trade-off between computational efficiency and

statistical efficiency, here measured by the parameteric MSE defined in Section 1.3.1. On one end of the spectrum, inferring parameters from a single distribution using Maximum-Likelihood Estimation (MLE) is statistically efficient but can be computationally inefficient (Section 1.3.2). On the other end of the spectrum, estimating parameters from a ratio of distributions using binary classification, comes with a computational advantage (Section 1.3.2) that is paid for by a statistical error that can be exponentially large in the dimensionality of the data [20, 90]. This second part of this thesis is dedicated to achieving a better trade-off for binary classification by provably reducing the statistical error.

Part II, Chapter 4 : Noise-Contrasive Estimation This work studies the estimation error of binary classification when the model is identified by the parameters of the target distribution. We highlight the following contribution :

- *We analytically optimized the estimation error.* The estimation error (here defined as the asymptotic variance of the estimated parameters of the classifier model) is usually provided when proposing a new estimation method. However, a theoretical analysis of this error [63, 11, 91] is rarely provided. The entirety of Part II which is half of this thesis, is dedicated to finding situations where we can derive an interpretable formula of the estimation error and analytically optimize it with respect to the choice of the proposal distribution.

Personal contributions. I wrote the proofs, code, and text, which were reread and edited by my supervisors. The initial idea of studying the estimation error of noise-contrastive estimation as a prototype for self-supervised learning was suggested by Aapo Hyvärinen.

Publications :

O. Chehab, A. Gramfort, A. Hyvärinen. *The Optimal Noise in Noise-Contrastive Learning Is Not What You Think*. Uncertainty in Artificial Intelligence (UAI), 2022.

Part II, Chapter 5 : Annealed Noise-Contrasive Estimation This work focuses on one a specific parameterization of the classifier model, using the normalizing constant of the target distribution. We prove that the estimation error can be reduced by annealing, which consists in introducing a path of distributions between the two original ones to classify. We highlight the following contributions :

- *We provide the first proof that annealing (introducing a path between two distributions) can make noise-contrastive estimation amenable to high-dimensional problems.* Our theory provided insights that could not necessarily be obtained from intuition alone. For example, we proved the optimal annealing path in high dimensions is made of distributions that are mixtures of the target and the proposal. We also proved that traversing that path at different speeds can lead to an estimation error that is exponential, polynomial, or constant in the dimensionality of the problem.
- *We help unify literatures for estimating the normalizing constant and parameters of a statistical model, and self-supervised learning.* We show that the same self-supervised task of classifying between data and noise, is at the root of estimating the parameters (using noise-contrastive estimation) and normalizing constant (using importance sampling) of a statistical model. We also show that annealing the classification task recovers many Monte-Carlo estimators of the normalizing constant from the past 50 years.

Personal contributions. Building on prior work of mine [92], I wrote the blueprint, proofs, code, and text for this project, which were reviewed by my co-authors. Aapo Hyvärinen and Andrej Risteski helped frame this paper. Andrej helped with many creative discussions, suggesting research directions and namely with making the right assumptions (*e.g.* exponential family distributions with bounded partition functions) to obtain general results. Andrej also helped review proofs.

Publications :

O. Chehab, A. Hyvärinen, A. Risteski. *Provable benefits of annealing for estimating normalizing constants*. Neural Information Processing Systems (NeurIPS), 2023. Spotlight.

Première partie

Applications to Brain Activity

2 - Regression Tasks on MEG data

This chapter presents the work published in :

O. Chehab*, A. Defossez*, J.C. Loiseau, A. Gramfort, J.R. King. *Deep Recurrent Encoder : A scalable end-to-end network to model brain signals*. Journal of Neurons, Behavior, Data analysis, and Theory, 2022.
(* means shared first-authorship)

Only minor changes have been made : additional text is included to explain how the model we obtained in this paper, relates to self-supervised learning.

2.1 . Summary

Understanding how the brain responds to sensory inputs from non-invasive brain recordings like magnetoencephalography (MEG) can be particularly challenging : (i) the high-dimensional dynamics of mass neuronal activity are notoriously difficult to model, (ii) signals can greatly vary across subjects and trials, and (iii) the relationship between these brain responses and the stimulus features is non-trivial. These challenges have led the community to develop a variety of preprocessing and analytical (almost exclusively linear) methods, each designed to tackle one of these issues. Instead, we propose to address these challenges through a specific end-to-end deep learning architecture, trained to predict the MEG responses of multiple subjects at once. We successfully test this approach on a large cohort of MEG recordings acquired during a one-hour reading task. Our Deep Recurrent Encoder (DRE) reliably predicts MEG responses to words with a three-fold improvement over classic linear methods. We further describe a simple variable importance analysis to investigate the MEG representations learned by our model and recover the expected evoked responses to word length and word frequency. Lastly, we show that, contrary to linear encoders, our model captures modulations of the brain response in relation to baseline fluctuations in the alpha frequency band. The quantitative improvement of the present deep learning approach paves the way to a better characterization of the complex dynamics of brain activity from large MEG datasets.

2.2 . Context and Contributions

A major goal of cognitive neuroscience consists of identifying how the brain responds to distinct experimental conditions. While descriptive statistics and statistical tests are classically used to analyze neural data [93], this approach is not suited to predict how the brain should react to new conditions. The resulting models of the brain can thus be particularly challenging to compare. By contrast, predictive encoding models [94, 70] can be directly trained to predict brain responses to various experimental conditions, and compared on their ability to accurately predict novel conditions. For example, encoding models allow the estimation of integration constants in the brain [95, 96], the hierarchical organization of visual [97] and speech processing [98, 99]. Beyond MEG, predictive models have

enabled automatic segmentation [100] and dynamical system identification [101, 102]. In functional Magnetic Resonance Imaging, predictive encoding models are starting to emulate complex neural processing [103] and are a step towards discovering new phenomena [104, 105]. Yet, this general objective of developing encoding models faces three major challenges when working with non-invasive and time-resolved signals collected by magneto- and electro-encephalography (M/EEG).

Challenge 1 : rich response dynamics M/EEG signals are known and promoted for their excellent temporal resolution. While this ability to measure cognitive processes at a millisecond timescale offers unique opportunities for fine chronometry of neural responses in humans, it also makes such signals notoriously difficult to analyze. For example, brain responses to audio streams overlap in time making their identification difficult. To address this issue in the context of encoding models, it is standard to employ a Temporal Receptive Field (TRF) model [106–115] [1]. TRF models are commonly designed to predict neural responses to exogenous stimulation by fitting a linear regression model with a fixed time-lag window of past sensory stimuli. By doing so, the predictions derived from TRFs are only influenced by stimuli descriptors, enabling them to modulate their response based on previous brain activity. Consequently, unless the basal activity from previous time points is introduced as an exogenous feature, TRF cannot learn to capture neuronal adaptation responses [118], nor can it learn to vary an evoked response as a function of the pre-stimulus alpha power [119, 71].

Challenge 2 : inter-trial and inter-subject variability Neuronal recordings in general, and M/EEG in particular, can be extremely variable across trials and subjects [82, 70, 120, 71, 121]. To reduce the nuisance factors behind these variations such as eye blinks, head movements, cardiac, and face muscle activity which corrupt MEG recordings, it is common to make use of multiple sessions and subjects within a study. For example, several methods based on spatial filtering [122–126] or "hyper-alignment" use linear models such as canonical correlation analysis (CCA), partial least square regression (PLS), multi-view ICA and back-to-back regressions (B2B) [127–132] to isolate the brain responses shared across trials and/or individuals. However, these denoising techniques can also remove relevant signals. For example, VanRullen [71] have repeatedly shown that evoked responses to sensory input can be modulated by pre-stimulus alpha activity in a predictable way. Averaging trials, or filtering out this variability during preprocessing would therefore prevent the identification of such phenomenon.

Challenge 3 : identifying the relationship between brain responses and stimulus features A large part of cognitive neuroscience aims to identify *how* the brain responds to stimulus features. For example, are V1 neurons tuned to respond to luminance, contrast, oriented lines, or faces? When and where does this elicit a response? To tackle this issue, it is common to present many stimuli to the subject, and fit a general linear model (GLM) to predict brain responses given a set of hypothetical features [94]. This approach can be limited, as GLMs only reveal brain responses to features predetermined by the analyst [133, 116, 134, 135] and understanding interactions between

1. also referred to as Finite Impulse Response (FIR) analysis in fMRI [116], and Distributed Lag modeling in statistics [117]

features often requires explicitly modeling these interactions (e.g. as cross-terms to form a quadratic polynomial) and feeding them to a linear regression [136].

From supervised to self-supervised learning Here, we propose to simultaneously address these three core challenges with a unique end-to-end "Deep Recurrent Encoding" (DRE) neural network trained to robustly predict brain responses from both (i) past MEG activity and (ii) current experimental conditions. We test DRE on 99 subjects recorded with MEG during a one-hour long reading task, and show that our model (1) better predicts MEG responses than standard models, (2) efficiently captures inter-trial and inter-subject variability, and (3) identifies feature-specific responses as well as interactions between basal activity and these evoked responses.

While forecasting is a traditionally supervised task, it is here also used to learn useful, even interpretable, representations of brain activity which is an unsupervised goal. There are many ways to characterize useful representations, as discussed in section 1.1. Here, we show that representations of brain activity learnt by our DRE model capture meaningful information from a cognitive neuroscience perspective. Specifically, these representations capture the interaction between past brain activity and external stimuli, which is not the case for representations obtained from other models in this paper. This is an example of *self-supervised learning*, where a supervised objective such as forecasting is used to achieve an unsupervised goal such as learning useful features.

2.3 . Self-Supervised Regression Tasks

We next present, with consistent and self-contained mathematical notations, a methodological progression from linear to nonlinear encoding models of neural dynamics as observed with MEG. We also discuss the statistical and computational benefits of recurrent models, as well as the novel methodological ideas proposed with the DRE model.

Problem formalization In the case of MEG, the measured magnetic fields x reflect a tiny subset of brain dynamics h — specifically a partial and macroscopic summation of the synaptic input to cortical pyramidal cells. Given the physics of electromagnetic fields propagation, it is standard to assume that these neuronal magnetic fields have a linear, stationary, and instantaneous relationship with the magnetic fields measured via MEG sensors [137]. We refer to this "readout operator" as C, a matrix which is subject-specific because it depends on the location of pyramidal neurons in the cortex and thus on the anatomy of each subject. Furthermore, the brain dynamics governed by a function f evolve according to their past and to external stimuli u [138]. In sum, we can formulate the problem as follows :

$$\begin{cases} x_{\text{current}} &= Ch_{\text{current}} \\ h_{\text{current}} &= f(h_{\text{past}}, u_{\text{current}}) \end{cases}$$

Operational Objective Here, we aim to parameterize f with θ, and subsequently learn θ and C to obtain a statistical (as opposed to biologically constrained as in [139]) generative model of observable

brain activity that accurately predicts MEG activity $\hat{x} \in \mathbb{R}^{d_x}$ given an initial state and a series of past stimuli.

Notations We denote by $u_t \in \mathbb{R}^{d_u}$ the stimulus with d_u encoded features at time t, $x_t \in \mathbb{R}^{d_x}$ the MEG recording with d_x sensors, $\hat{x}_t$ its estimation, and by $h_t \in \mathbb{R}^{d_h}$ the underlying brain activity. Because the true underlying brain dynamics are never known, h will always refer to a model estimate. To facilitate the parametrization of f, it is common in the modeling of dynamical systems to explicitly create a "memory buffer" by concatenating successive lags. We adopt the bold notation $\boldsymbol{h}_{t-1:t-\tau_h} := [h_{t-1}, ..., h_{t-\tau_h}] \in \mathbb{R}^{d_h \tau_h}$ for flattened concatenation of $\tau_h \in \mathbb{N}$ time-lagged vectors. With these notations, the dynamical models considered in this paper are described as :

$$\begin{cases} x_t &= Ch_t \\ h_t &= f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \boldsymbol{u}_{t:t-\tau_u}) \end{cases} \tag{2.1}$$

where

- $f : \mathbb{R}^{d_h \tau_h + d_u(\tau_u+1)} \to \mathbb{R}^{d_h}$ governs brain dynamics given the preceding brain states and external stimuli
- $C \in \mathbb{R}^{d_h \times d_x}$ is a linear, stationary, instantaneous, and subject-specific observability operator that makes a subset of the underlying brain dynamics observable to the MEG sensors.

We next list the models considered in this study.

Temporal Receptive Field (TRF) Temporal receptive fields (TRF) [106] are arguably the most common model for predicting neural time series in response to exogeneous stimulation. The TRF equation is that of control-driven linear dynamics :

$$h_t = f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \boldsymbol{u}_{t:t-\tau_u}) = B\boldsymbol{u}_{t:t-\tau_u} \ , \tag{2.2}$$

where $B \in \mathbb{R}^{d_h \times d_u.(\tau_u+1)}$ is the convolution kernel that maps the stimuli to the brain response and $\theta = \{B\}$. By definition, the TRF kernel encodes the input-output properties of the system, namely, its characteristic time scale, its memory, and thus its ability to sustain an input over time. A computational drawback is that the TRF kernel size scales linearly with the duration of the neural response to the stimulus. For example, a dampened oscillation evoked by the stimulus could last one hundred time samples ($\tau_u = 99$) and would require $B \in \mathbb{R}^{d_h \times 100 d_u}$ to reach 100 steps in the past, even though oscillatory dynamics can be compactly written as a second-order differential equation expressing h_t in terms of only two of its own past states (h_{t-1}, h_{t-2}) [2]. Emulating this, we will introduce a recurrent component to the TRF model to tackle the issue of dimensionality.

Recurrent Temporal Receptive Field (RTRF) A Recurrent Temporal Receptive Field (RTRF) is a linear auto-regressive model. The RTRF with exogenous input can model time-series from its own past (e.g., past brain activity) *and* from exogenous stimuli. Unrolling the recurrence reveals that current brain activity can be expressed in terms of past activity. This corresponds to recurrent dynamics with control :

$$h_t = f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \boldsymbol{u}_{t:t-\tau_u}) = A\boldsymbol{h}_{t-1:t-\tau_h} + B\boldsymbol{u}_{t:t-\tau_u} \ , \tag{2.3}$$

2. A sine wave can be produced by a simple linear auto-regressive (AR) model of order 2

where the matrix $A \in \mathbb{R}^{d_h \times (d_h \cdot \tau_h)}$ encodes the recurrent dynamics of the system and $\theta = \{A, B\}$.

The dependency of h_t on h_{t-1} in Eq. 2.3 means we need to unroll the expression of h_{t-1} in order to compute h_t. However, it has been shown that linear models perform poorly in this case, as terms of the form A^t (A to the power t) will appear, with either exponentially exploding or vanishing eigenvalues. This rules out optimization with first order methods due to the poor conditioning of the problem [140], or using a closed-form solution. To circumvent unrolling the expression of h_{t-1}, we need to obtain it from what is measured at time $t-1$. This however assumes the existence of an inverse relationship from x_{t-1} to h_{t-1}, which we assume here to be linear by using the pseudo inverse of $C : h_{t-1} = C^{\dagger} x_{t-1}$. As a result, h_t and x_t are identifiable to one another, and Eq. 2.3 can be solved in closed form as a regular linear system [141]. Initializing the RTRF dynamics with the pre-stimulus data can be written as :

$$h_t = C^{\dagger} x_t \qquad \forall t \in \{0, ..., \tau_h - 1\} \ , \tag{2.4}$$

where τ_h is chosen to match the pre-stimulus duration τ.

Though the recurrent component of the RTRF is able to reduce the receptive field τ_u of TRF, it is nevertheless constrained to maintain a 'sufficiently big' receptive field τ_h to initialize over τ_h steps. The following model, DRE, will avoid this issue, and will also not require that h_t and x_t are identifiable via linear inversion.

Deep Recurrent Encoder (DRE) DRE is an architecture based on the Long-Short-Term-Memory (LSTM) computational block [142]. It is useful to think of the LSTM as a "black-box nonlinear dynamical model", which composes the RTRF building block with nonlinearities and a memory module which reduces the need for receptive fields, so that $\tau_h = 1$ and $\tau_u = 0$. It is employed here to capture nonlinear dynamics evoked by a stimulus. A single LSTM layer can be formulated as [142] :

$$\begin{cases} h_t &= f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \boldsymbol{u}_{t:t-\tau_u}) = o_t \odot \tanh(m_t) \\ m_t &= d_t \odot m_{t-1} + i_t \odot \tilde{m}_t \\ \tilde{m}_t &= \tanh(Ah_{t-1} + Bu_t) \end{cases} , \tag{2.5}$$

where the $\tanh$ nonlinearity is applied element-wise, $\odot$ is the Hadamard (element-wise) product, and $(d_t, i_t, o_t) \in (\mathbb{R}^{d_m})^3$ are data-dependent vectors with values between 0 and 1 modeled as forget (or drop) input and output gates, respectively. The memory module $m_t \in \mathbb{R}^{d_m}$ thus interpolates between a "past term" $m_{t-1} \in \mathbb{R}^{d_m}$ and a "prediction term" $\tilde{m}_t \in \mathbb{R}^{d_m}$, taking h_t as input. The "prediction term" (See Eq. 2.5 last equation) resembles that of the previous RTRF model except that it is here composed with a tanh nonlinearity which conveniently normalizes the signals.

Again, the dependency of h_t on h_{t-1} in Eq. 2.3 meant that we needed to unroll the expression of h_{t-1} to compute h_t. While this is numerically unstable for the RTRF, the LSTM is designed such that h_t and its gradient are stable even for large values of t. As a result, h_t and x_t do not need to be identifiable to one another. In other words, contrary to RTRF, the LSTM allows h_t to represent a hidden state containing potentially more information than its corresponding observation x_t.

We now motivate three modifications made to the standard LSTM.

First, we help it recognize when (not) to sustain a signal, by augmenting the control u_t with a mask embedding $p_t \in \{0, 1\}$ indicating whether the provided MEG signal generates the current brain

response (i.e. 1 before word onset and 0 thereafter). Second, we automatically learn to align subjects with a dedicated subject embedding layer. Indeed, a shortcoming of standard brain encoding analyses is that they are commonly performed on each subject separately. However, this implies that one cannot exploit potential similarities across subjects. Here, we adapt the LSTM in the spirit of Défossez et al. [143] so that a *single* model is able to leverage information across multiple subjects. We do this by augmenting the control u_t with a "subject embedding" $s \in \mathbb{R}^{d_s}$, that is learned for each subject. Note that this amounts to learning a matrix in $\mathbb{R}^{d_s \times n_s}$ that is applied to the one-hot-encoding of the subject number. In order words, each subject has a vectorized representation that is one column of the embedding matrix. Setting $d_s < n_s$ allows us to use the same LSTM block to model subject-wise variability, and to train across subjects simultaneously while leveraging similarities across subjects.

Third, for comparability purposes, RTRF and LSTM should access the same pre-stimulus MEG information $\boldsymbol{x}_{\tau:1}$. Incorporating the initial MEG, before word onset, is done by augmenting the control with $p_t \odot x_t$. The extended control reads : $\tilde{u}_t = [u_t, s, p_t, p_t \odot x_t]$, and the LSTM with augmented control $\tilde{u}_t$ finally reads :

$$h_t = f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \tilde{\boldsymbol{u}}_{t:t-\tau_u}) = \mathrm{LSTM}_\theta(h_{t-1}, \tilde{u}_t) = \mathrm{LSTM}_\theta(\mathrm{LSTM}_\theta(h_{t-2}, \tilde{u}_{t-1}), \tilde{u}_t) \ . \tag{2.6}$$

In practice, to maximize expressivity, two modified LSTM blocks are stacked on top of one another (Figure 2.1).

Having introduced a nonlinear dynamical system for the brain response h_t, we can also extend the model Eq. 2.1 by challenging the linear instantaneous mixing from the brain response h_t to the measurements x_t. Introducing two new nonlinear functions d and e, respectively parametrized by θ_2 and θ_3, a more general model formally reads :

$$\begin{cases} \boldsymbol{x}_{t:t-\tau_x+1} & = d_{\theta_2}(h_t) \\ h_t & = f_{\theta_1}(\boldsymbol{h}_{t-1:t-\tau_h}, e_{\theta_3}(\tilde{\boldsymbol{u}}_{t:t-\tau_u})) \end{cases} , \tag{2.7}$$

where τ_x allows us to capture a small temporal window of data around x_t, and τ_u is taken to be much larger than τ_x. Indeed Eq. 2.7 corresponds to Eq. 2.1 if one sets $\tau_x = 1$ and $d_{\theta_2}(h_t) = Ch_t$, as well as $e_{\theta_3}(\tilde{\boldsymbol{u}}_{t:t-\tau_u}) = \boldsymbol{u}_{t:t-\tau_u}$. In more intuitive terms, the DRE model generalizes the linear instantaneous measurement of the previous models with a "convolutional autoencoder" [144]. The e (encoder) function is formed by convolutions and the d (decoder) function uses transposed convolutions, where both functions are two layers deep (Figure 2.1) [3].

In practice, we use a kernel size $K = 4$ for the convolutions. This impacts the receptive field of the network and the parameter τ_x. Equation Eq. 2.7 implies that the number of time samples in h and x are the same. However, a strong benefit of the convolutional auto-encoder is to perform a reduction of the number of time steps by using a stride S larger than 1. By using a stride of 2, one reduces the temporal dimension by 2. Indeed it boils down to taking every other time sample from the output of the convolved time series. Given that the LSTM module is by nature sequential, this reduces the number of time steps it has to consider when learning, which accelerates both training and evaluation. Further, there is evidence that LSTMs can only pass information over a limited number of time steps [146]. In practice, we use d_h output channels for the convolutional encoder.

3. While "encoding" typically means outputting the MEG with respect to the neuroscience literature, we use "encoder" and "decoder" in the context of deep learning auto-encoders [145] in this paragraph.

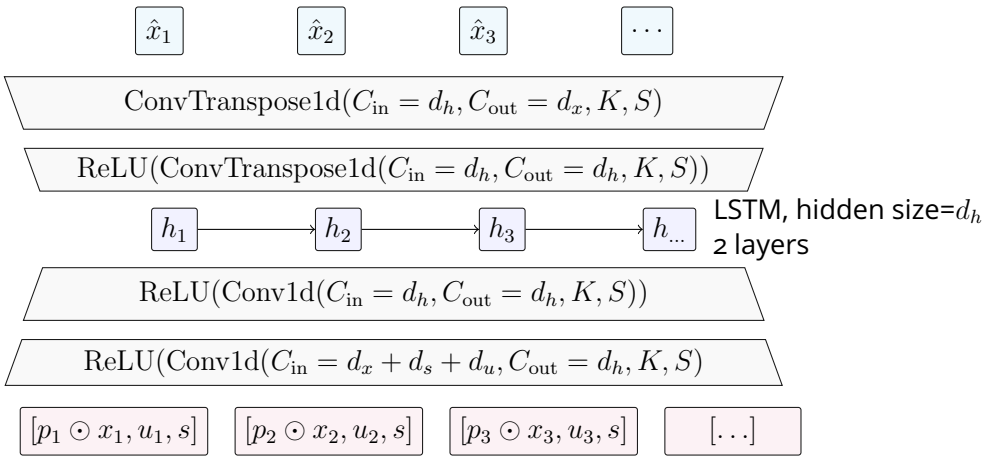


Figure 2.1 – Representation of the Deep Recurrent Encoder (DRE) model used to predict MEG activity. The masked MEG $p_t \odot x_t$ enters the network from the bottom, along with the control representation u_t and the subject embedding s. The encoder transforms the input with convolutions and ReLU nonlinearities. Then, the LSTM models the sequence of hidden states $\boldsymbol{h}_t$, which are converted back to the MEG activity estimate $\hat{x}_t$. $\mathrm{Conv1d}(C_{\text{in}}, C_{\text{out}}, K, S)$ represents a convolution over time with C_{in} input channels, C_{out} output channels, a kernel size K, and a stride S. Similarly, $\mathrm{ConvTransposed1d}(C_{\text{in}}, C_{\text{out}}, K, S)$ represents a transposed convolution over time.

In summary, our DRE model generalizes TRF and RTRF models by using nonlinearities both in the dynamics of the brain response h_t and in its measurement x_t Eq. 2.1. It is done respectively with LSTM cells and a convolutional auto-encoder. Importantly, the DRE is equipped with a subject embedding allowing us to learn a joint model for the group of subjects.

Optimization losses The dynamics for the above three models (TRF, RTRF, DRE) are given by different expressions of f_θ as well as the mappings between x and h via C for TRF and RTRF, or c and e for DRE.

At test time, the models aim to accurately forecast MEG data from initial steps combined with subsequent stimuli. Consequently, one should train the models in the same setup. This boils down to minimizing a “multi-step-ahead" ℓ_2 prediction error :

$$\begin{cases} \text{minimize}_{\ \theta_1,\theta_2,\theta_3} & \sum_t \|x_t - \hat{x}_t\|_2^2 \\ \text{s.t.} & \boldsymbol{x}_{t:t-\tau_x+1} = d_{\theta_2}(h_t) \\ & h_t = f_{\theta_1}(\boldsymbol{h}_{t-1:t-\tau_h}, e_{\theta_3}(\tilde{\boldsymbol{u}}_{t:t-\tau_u})) \end{cases}$$

Because the prediction task uses initial brain activity (in the augmented stimulus $\tilde{u}_t$) to predict future brain activity, this it is specified as "filtering" in probabilistic literature [147]. This “multi-step-ahead" minimization requires unrolling the recurrent expression of h_t over the preceding time steps, which the LSTM-based DRE model is able to do [148]. The DRE model takes as input the observed MEG at the beginning of the sequence, and must predict the future MEG measurements using the (augmented) stimulus $\tilde{u}_t$. Note that the mapping to and from the latent space, e_{θ_3} and d_{θ_2}, are learned *jointly* with

the dynamical operator f_{θ_1}. Furthermore, the DRE has reduced temporal receptive fields, thus the computational load is lightened and allows for a low or high-dimensional latent space.

Given that the RTRF model is linear and can suffer from numerical instabilities (see above), it is trained with the "one-step-ahead" version of the predictive loss with squared ℓ_2 regularization :

$$\begin{cases} \text{minimize}_{\ \theta} & \sum_t \|x_t - \hat{x}_t\|_2^2 + \lambda \|\theta\|_2^2 \\ \text{s.t.} & \hat{x}_t = C h_t \\ & h_t = f_\theta(\boldsymbol{h}_{t-1:t-\tau_h}, \boldsymbol{u}_{t:t-\tau_u}) \\ & \boldsymbol{h}_{t-1:t-\tau_h} = C^\dagger \boldsymbol{x}_{t-1:t-\tau_h} \end{cases}$$

TRF models are also trained with this "one-step-ahead" loss. As mentioned above, the linear models (TRF) require a larger receptive field than the nonlinear DRE. Large receptive fields induce a computational burden, because each time lag comes with a spatial dimension of size d_h or d_u. To tackle this issue, C is chosen to reduce this spatial dimension. In practice, we choose to learn C *separately* from the dynamics to simplify the training procedure of the linear models. Given a participant, we fit a Principal Component Analysis (PCA) with 40 components on their averaged (evoked) MEG data : the resulting PCA map is taken to be the matrix $C \in \mathbb{R}^{d_x \times 40}$. The resulting latent space will thus explain most of the variance of the original recording. Indeed, when training the TRF model on all 270 MEG sensors with no hidden state ($6.4 \pm 0.22\%$, MEAN and SEM across subjects) or using a 40-component PCA (6.43 ± 0.17%), we obtained similar performances. The pseudo-inverse $C^\dagger$ required to compute the previous latent state h_{t-1} is also obtained from the PCA model. Note that dimensionality reduction via linear demixing is a standard preprocessing step in MEG analysis [123, 149, 150].

Model Evaluation Following seminal works (*e.g.* by Kay et al. [151], Güçlü and Gerven [152]), models are evaluated using the Pearson Correlation R (between -1 and 1) between the model prediction $\hat{x}$ and the true MEG signals x for each channel and each time sample (after the initial state) independently [4]. When comparing the overall performance of the models, we average over all time steps after the stimulus onset, and over all MEG sensors for each subject independently.

Feature Importance To investigate what a model actually learns, we use Permutation Feature Importance [153] which measures the drop in prediction performance when the jth input feature u^j is shuffled :

$$\Delta R_j = R - R_j \ , \tag{2.8}$$

By tracking ΔR over time and across MEG channels, we can locate in time and space the contribution of a particular feature (e.g. word length) to the brain response.

Experiment The model weights are optimized with the training and validation sets, while the penalization λ for the linear models (TRF and RTRF) is optimized with a grid search over five values

4. Figure 2.5 in the Appendix 2.7 reports the same evaluations, using a different metric : the explained variance quantified by the coefficient of determination R^2 of the brain response by the model predictions.

distributed logarithmically between 10^{-3} and 10^{3}. Training of the DRE is performed with ADAM [154] using a learning rate of 10^{-4} and PyTorch's default parameters [155] for the running averages of the gradient and its square. The training is stopped when the error on the validation set increases. In practice, the DRE and the DRE-PCA were trained over approximately 20 and 80 epochs, respectively.

Statistics Each subject score is obtained using the model prediction on held-out trials, using the learned subject embeddings as the "alignment function", similar to Chen et al. [156], Zhang et al. [157], Haxby et al. [158], Bazeille et al. [131], Richard et al. [132]. To test the reliability of our effects (e.g. prediction performance, feature importance, model comparison), we assess confidence intervals and p-values across subjects using a non-parametric Wilcoxon rank test across subjects. When applicable, we correct these estimates for multiple comparisons using a false discovery rate (FDR) across time samples and channels. Note that subjects can be treated as independent observations to derive meaningful p-values since the statistics are based on held-out data independent from the training set.

Noise Ceiling Noise ceilings are typically estimated using batches of repeated conditions [159, 160, 152], to evaluate the maximal amount of explainable variance. This involves multiple presentations of the same stimulus characterized by a given feature set. In our case, however, sentences are presented only once to each subject. Further, we cannot control one of the variables input to our encoding models : namely, the baseline brain activity.

2.4 . Experiment on MEG data

Experimental design We analyze 99 subjects from the Mother Of Unification Studies (MOUS) dataset [161] who performed a one-hour reading task while being recorded with a 273-channel CTF MEG scanner. The task consisted of reading approximately 2,700 words flashed on a screen in rapid series. Words were presented sequentially in groups of 10 to 15, with a mean inter-stimulus interval of 0.78 seconds (min : 300ms, max : 1,400ms). Sequences were either random word lists or actual sentences (50% each). For this study, both conditions were used to obtain a larger data sample. However, this study does not investigate the differences obtained across these two conditions and instead focuses on word attributes (*e.g.* visual length and frequency of use in language) that are independent of the sentence context. Out of the original 102 subjects, 3 were discarded from the study because we could not reliably parse their stimulus channels.

Stimulus preprocessing We focus on four well-known features associated with reading, namely word length (i.e., the number of letters in a word), word frequency in natural language (as derived by the wordfreq Python package [162], and measured on a logarithmic scale), and a binary indicator for the first and last words of the sequence. At a given time t, each stimulus $u_t \in \mathbb{R}^4$ is therefore encoded with four values, fed to the models as a square function that is non-zero for the duration of the stimulus. Each feature is standardized to have zero mean and unit variance. Word length is expected to elicit an early (from t=100 ms) visual response in posterior MEG sensors, whereas word

frequency is expected to elicit a late (from t=400 ms) left-lateralized response in anterior sensors. In the present task, word length and word frequency are correlated R=-0.48.

MEG Preprocessing As we are primarily interested in evoked responses [163], we band-pass filtered between 1 and 30 Hz and downsampled the data to 120 Hz using the MNE software [164] with default settings : i.e. a FIR filter with a Hamming window, a lower transition bandwidth of 1 Hz with -6 dB attenuation at 0.50 Hz and a 7.50 Hz upper transition bandwidth with an attenuation of -6 dB at 33.75 Hz.

To limit the interference of large artefacts on model training, we use Scikit-learn's RobustScaler with default settings [165] to normalize each sensor using the 25th and 75th quantiles. Following this step, most of the MEG signals will have a scale around 1. Since we observed a few large scale outliers, we chose to reject any segment of 3 seconds that contains amplitudes higher than 16 in absolute value (fewer than 0.8% of the time samples).

These continuous data are then segmented between 0.5 s before and 2 s after word onset, yielding a three-dimensional MEG tensor per subject : words, sensors, and time samples relative to word onset. For each subject, we form a training, validation, and test set using respectively 70%, 10%, and 20% of these segments, ensuring that two segments from different sets do not originate from the same word sequence to avoid information leakage. This corresponds to 191K, 27K, and 53K segments used for the train, validation, and test sets, respectively. Each segment has a spatial dimension of 273 sensors and a temporal dimension of 300 time points (2.5 s sampled at 120 Hz).

For clarity, some figures use global field power (GFP) to summarize effects over time. GFP refers to the standard deviation across MEG channels of an average evoked response.

Model hyper parameters We compare the three models introduced in Section 2.3 over $n_s = 99$ subjects. For the TRF, we use a lag on the control of $\tau_u = 250$ time steps (about 2 s). This corresponds to the duration of the signal after the stimulus onset. For the RTRF, we use $\tau_u = \tau_h = 40$ time steps. This lag is close to the minimum inter-word duration of 300 ms, and corresponds to the part of the initial MEG (i.e. 333 ms out of 500 ms, at 120 Hz) that is passed to the model to predict the 2.5 s MEG sequence during the evaluation.

For the DRE model, we use a subject embedding of dimension $d_s = 16$, a latent state of dimension $d_h = 512$, a kernel size $K = 4$, and a stride $S = 2$. The subject embeddings are initialized as Gaussian vectors with zero mean and unit variance, while the weights of the convolutional layers and the LSTM are initialized using the default "Kaiming" initialization [166]. Like its linear counterpart, the DRE is given the first 333 ms of the MEG signal to predict the complete 2.5 s of a training sample.

Ablation study To investigate the importance of the different components of the DRE model, we implement an ablation study by fitting the model with all but one of its components. To this end, we compare the DRE to i) the DRE without using the 333 ms of pre-stimulus initial MEG data (DRE NO-INIT), ii) the DRE trained in the 40-dimensional PCA space used for the linear models (DRE PCA), iii) the DRE devoid of a subject embedding (DRE NO-SUBJECT), and to iv) the DRE devoid of the convolutional auto-encoder (DRE NO-CONV).

The code developed for the present study is available at :
https://github.com/facebookresearch/deepmeg-recurrent-encoder.

2.5 . Results

We first evaluate the DRE's ability to predict brain responses to written words presented in rapid serial visual presentation and measured with MEG, and compare these brain predictions to those of linear encoding models (TRF, RTRF). Then, we show with ablation experiments which elements of the DRE help address the challenges of rich dynamics, inter-subject, and inter-trial variability. Finally, we show how feature importance helps address the third challenge introduced above, namely : identifying the relationship between brain responses and stimulus features. Our feature importance analysis shows that representations learnt by our DRE model are useful beyond the *supervised* task of forecasting brain activity. They capture meaningful interactions between brain activity and external stimuli, which makes them interpretable as an *unsupervised* goal. Using a supervised objective to achieve an unsupervised goal falls within our definition of self-supervised learning from section 1.1.

Modeling rich MEG dynamics : model comparison. DRE outperforms the baseline TRF and RTRF models with up to a three-fold improvement (Figure 2.2). To provide a fair comparison between the models, we also compare TRF to a NO-INIT DRE, i.e. to a DRE architecture that ignores the pre-stimulus MEG activity. The results show that DRE NO-INIT consistently outperforms TRF (mean correlation score $R = 0.077$ on average across all subjects, time samples, and all channels ; standard error of the mean across subjects : $\pm$ 0.002 for DRE NO-INIT vs. $R = 0.064 \pm 0.002$ for TRF). This difference is strongly significant ($p < 10^{-17}$) under a Wilcoxon test across subjects. Similarly, DRE (R=0.20 ± 0.003) significantly ($p < 10^{-17}$) outperforms RTRF (R=0.10 ± 0.003), when both of these models are given as input the pre-stimulus MEG activity. To verify that this gain is not trivially accounted for by the limited dimensionality of RTRF (trained with 40-dimensional Principal Components because of computational limitations), we trained DRE with the *same* PCA-reduced data as RTRF. The results confirm that DRE obtains a higher performance (R=0.16 ± 0.003, $p < 10^{-16}$) than RTRF. Overall, these results suggest that DRE better models the rich M/EEG dynamics than linear models.

Subject embeddings efficiently capture inter-individual variability To evaluate the importance of the subject-embedding layer in capturing inter-individual variability, we trained the DRE without a subject embedding layer (DRE NO-SUB). The comparison between DRE and DRE NO-SUB reveals a clear difference ($\Delta R = 0.038$, $p < 10^{-17}$). This result shows that the subject embedding layer efficiently re-aligns subjects' brain responses to model the dynamics specific to each – or shared across – subject(s).

Recurrence efficiently captures inter-trial variability. Brain responses to sensory input are known to vary with ongoing brain activity[71, 167]. Recurrent models (RTRF, DRE) are thus well suited to capture such phenomenon : initialized with 333 ms of pre-stimulus MEG, they can use basal brain activity to modulate the post-stimulus MEG predictions. Our results confirm this prediction : TRF is outperformed by RTRF (0.10 ± 0.003, $p < 10^{-16}$) with an average performance increment of

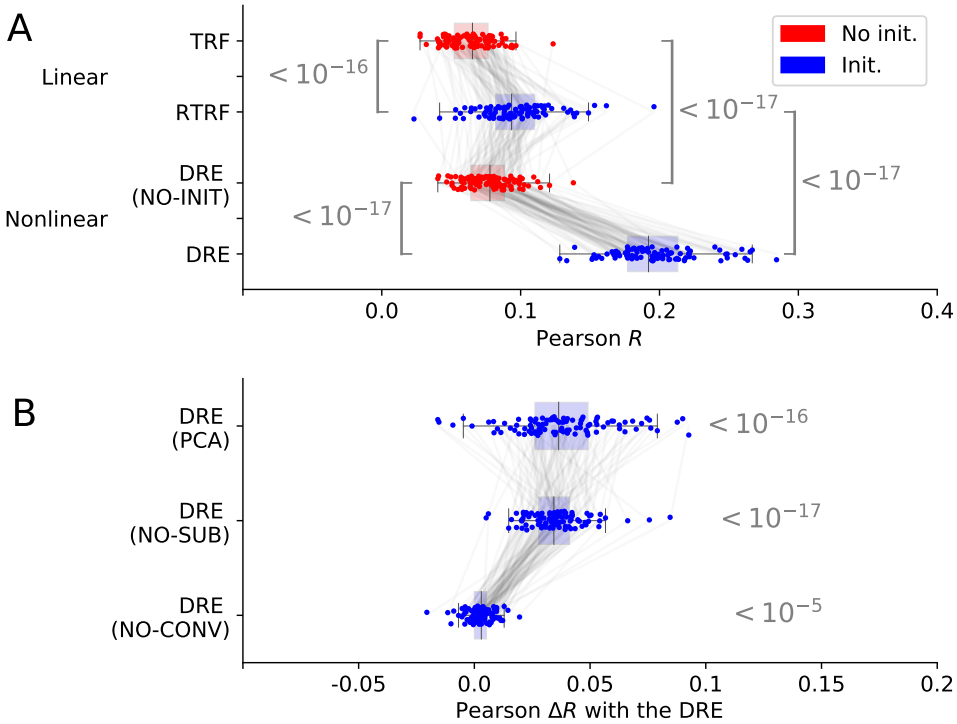


Figure 2.2 – *A : Model Comparison*. Predictive performance of different models over 99 subjects (dots) (chance=0). Boxplots indicate median and interquartile range, while gray lines follow each subject. The gray brackets indicate the p-value obtained with a pair-wise Wilcoxon comparison across subjects. Initialization and nonlinearity increase predictive performance. *B : Ablation Study of our model (DRE)*. Pearson R mean obtained by removing a component from (Convolutional Auto-encoder or Subject-embedding) or adding an element (PCA embedding composed with the Convolutional Auto-encoder) to the DRE architecture and retraining the model, and the resulting p-values obtained by comparing the resulting scores to those of the normal DRE. Convolutions marginally impact performance of the DRE (they are essentially used for speed), but training over all MEG sensors (vs. PCA space) and using subject embeddings designed for realignment are responsible for an increase of approximately 4%.

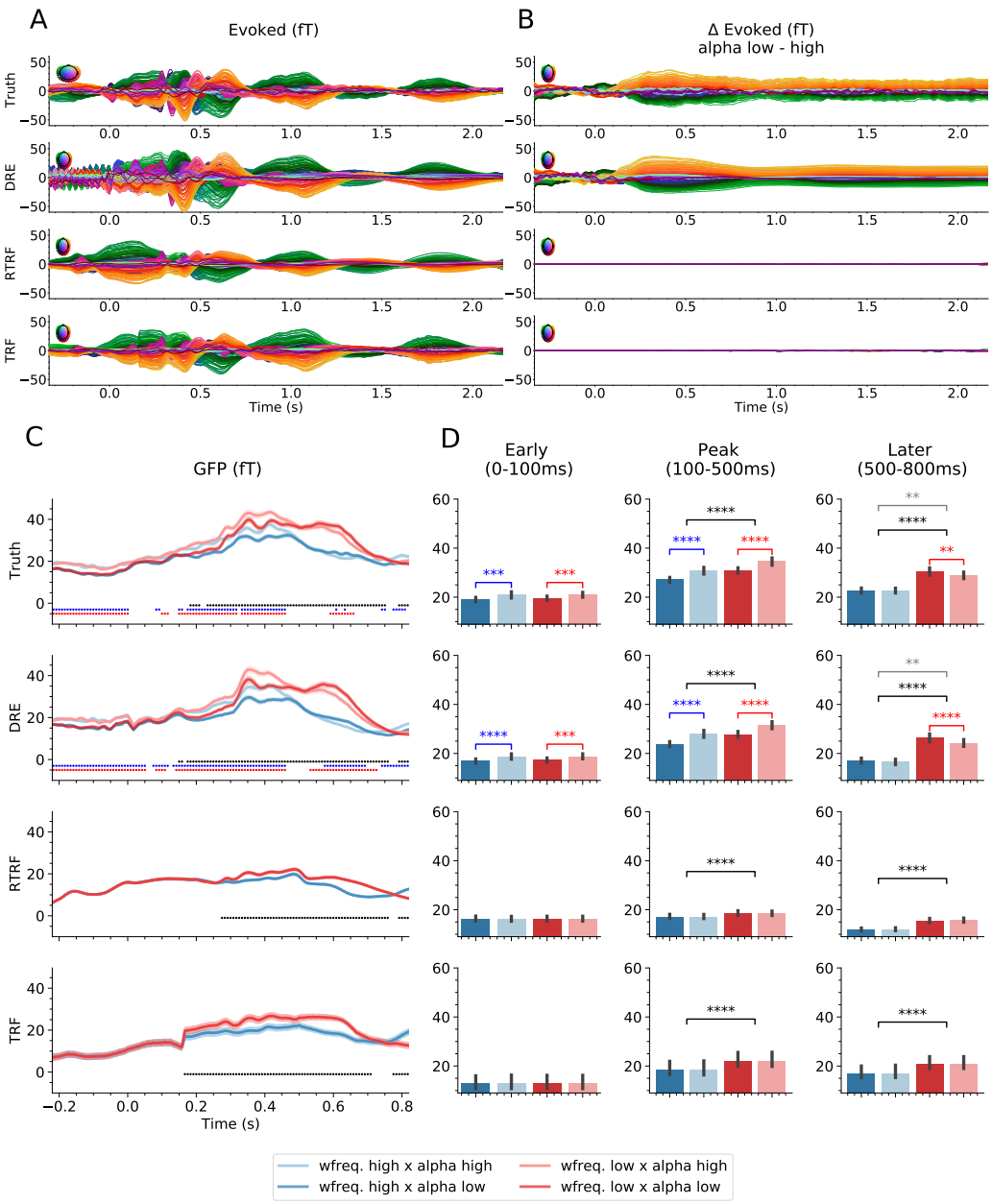


Figure 2.3 – **A :** Brain response obtained for all three models (TRF, RTRF, DRE) and the actual data (Truth), averaged over the test set for all 99 subjects. Qualitatively, all models learn the mean MEG response, although with a variable precision. **B :** Difference of average brain response between trials with high *vs* low pre-stimulus alpha power (8-13 Hz). This analysis shows that DRE modulates the predicted brain response as a function of pre-stimulus alpha power. **C :** Global Field Power (GFP) of the brain response, as a function of pre-stimulus alpha power and word frequency. Qualitatively, all models capture the stimulus-modulation of the brain response, but only the DRE captures the alpha-modulation, which varies with the latency. **D :** Global Field Power (GFP) of the brain response averaged before the peak (0-100ms), during the ramp up of the peak (100-500ms) and after the peak of the evoked response (500-800ms). The number of stars illustrates the level of significance (** : 10^{-2}, *** : 10^{-3}, or **** : 10^{-4}). The marginal pre-stimulus alpha effect (light vs. dark) is followed by the marginal word frequency effect (black) and then an interaction (grey) between pre-stimulus alpha and word frequency (red vs. blue). Only DRE captures the same effects as in the actual data.

$\Delta R = 0.03$, and DRE (0.20 ± 0.003) outperforms DRE NO-INIT ($0.077 \pm 0.002\%$, $p < 10^{-17}$) with an average performance increment of $\Delta R = 0.12$.

DRE's recurrence specifically captures alpha-dependent evoked responses. To further explore how DRE learns inter-trial variability, we investigate a well-known interaction between evoked responses and pre-stimulus activity. Specifically, brain responses to sensory input are known to be modulated by pre-stimulus oscillatory activity in the "alpha" frequency range (8 - 13 Hz) [71]. To test whether this phenomenon can be detected in the present dataset, we compared the average evoked responses to words for "high pre-stimulus alpha" versus "low alpha" trials, using a median split for each subject separately. The results (Figure 2.3) show an effect of up to 50×10^{-12}T in fronto-temporal channels, peaking around 400 ms after word onset. Critically, while the single-trial predictions of DRE capture this phenomenon, neither TRF nor RTRF learn to modulate their evoked responses depending on the alpha power (Figure 2.3B. Bottom).

This interaction between pre-stimulus alpha activity and evoked responses varies with the content of words, and more specifically, with their frequency in natural language : a factorial split between "high alpha" *versus* "low alpha" trials and "high word frequency" *versus* "low word frequency" trials resulted in both main and interaction effects (Figure 2.3C-D). Specifically, the comparison between these 2x2 conditions reveals three main phases. First, a main effect of alpha can be observed before the evoked response (light vs. dark lines in Figure 2.3C, $p < 10^{-3}$). Second, the main effect of word frequency starts to become significant from ≈ 200 ms (blue vs. red lines, $p < 10^{-4}$). Finally, the main effect of alpha starts to fade away after ≈ 500 ms ($p > 10^{-2}$ for high-frequency words), but its interaction with the stimulus continues to be significant ($p < 10^{-2}$). Critically, DRE learns these interactions between pre-stimulus alpha power and stimulus responses, while the linear models do not.

Feature importance helps interpreting the links between brain responses and stimulus features Interpreting nonlinear and/or high-dimensional models is notoriously challenging [168]. This issue poses strong limitations on the application of deep learning to neural recordings, where interpretability remains a major goal [70, 169]. While DRE faces the same types of issues as any deep neural network, we show below that a simple feature importance analysis of the predictions of this model (as opposed to its parameters) yields results that are consistent with those obtained by linear models, and with those described in neuroscientific literature (cf. Section 2.3).

Feature importance quantifies the loss of prediction performance ΔR when a unique feature is shuffled across words as compared to a non-shuffled prediction. Here, we focus our analysis on word length and word frequency, as these two features have been repeatedly associated with early sensory responses in the visual cortex and late lexical responses in the temporal cortex, respectively [170, 171]. As expected, the feature importance for word length in Figure 2.4 peaked around 150 ms in posterior MEG channels, whereas the feature importance of word frequency peaked around 400 ms in fronto-temporal MEG channels, for both the TRF and the DRE models. Furthermore, we recover a second phenomenon known in the literature : the *lateralization* of lexical processing in the brain. Indeed, Figure 2.4 shows, for the word frequency, an effect similar in shape across hemispheres, but significantly higher in amplitude for the left hemisphere (e.g. $p < 10^{-10}$ in the frontal region, $p < 10^{-12}$ in the temporal region, for the DRE).

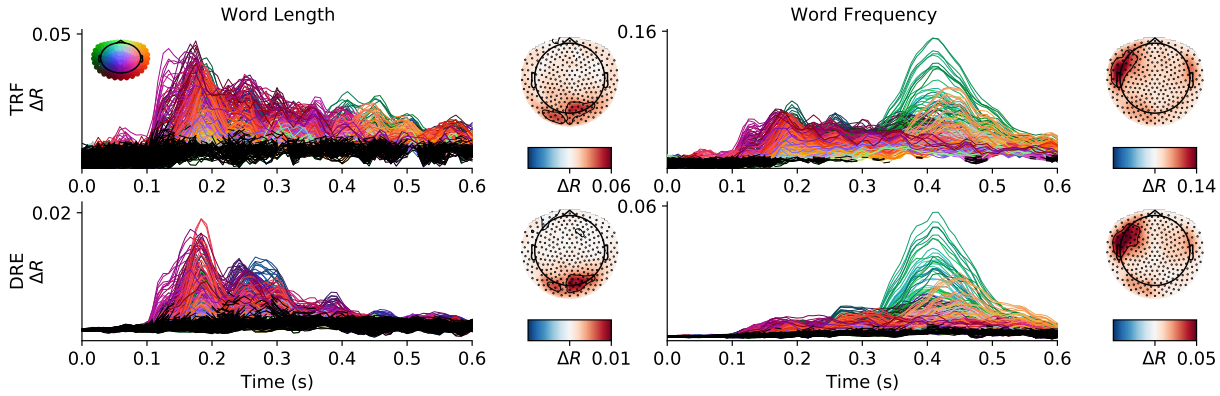


Figure 2.4 – Permutation importance (ΔR) of word length (left column) and word frequency (right column), as a function of spatial location (color-coded by channel position, see top-left rainbow topography) and time relative to word onset for the two main encoding models (rows). The amplitudes of each line represent the mean across words and subjects for each channel. An FDR-corrected Wilcoxon test across subjects assesses the significance for each channel and time samples independently. Non-significant effects (p-value higher than 5%) are displayed in black. Overall, this feature importance analysis confirms that early visual and late frontal responses are modulated by word length and word frequency, respectively, as expected.

These results suggest that, in spite of being high dimensional and nonlinear, DRE can be interpreted similarly to linear models in the present context.

2.6 . Discussion

The present study demonstrates that DRE outperforms several of its linear counterparts to predict MEG time series. In particular, it addresses the three challenges introduced above. First, the complex, nonlinear and non-stationary dynamics can be efficiently modeled by deep convolutional LSTM layers. Second, inter-trial and inter-individual variability can be addressed with recurrence (i.e. MEG-INIT) and subject embeddings, respectively. Finally, the relationship between stimulus features and brain responses can be interpreted in light of a permutation-based feature importance analyses.

Overall, the present study shows that the gain in prediction performance obtained by deep learning algorithms may not necessarily come at the price of interpretability. Indeed, we show here that DRE can be probed *a posteriori* to reveal how evoked responses relate to each stimulus feature and/or to pre-stimulus brain activity. This feature importance supplements ongoing efforts to open black-box models of brain activity. For example, Güçlü and Gerven [152] used a recurrent neural network to predict fMRI recordings, and quantified the impact of stimulus features by correlating them with the model's hidden state. Similarly, Keshishian et al. [96], analyzed the activations of a deep convolutional network with TRF to show how they captured "dynamical receptive fields". In both of these cases, however, these post-hoc analyses are based on (i) linear assumptions and (ii) the inner activations of the model. By contrast, the permutation feature importance used here focuses on probabilistic depen-

dencies between input features and the models' predictions, which generalize linear dependencies measured by correlation [153]. This approach can thus be applied to any black-box predictive model.

Deep neural networks have not yet emerged as the go-to method for neuroimaging [172, 173]. Nevertheless, several studies have successfully used deep learning architectures to model brain signals [174–177, 134]. In particular, deep nets trained on natural images [178–180], sounds [96], or text [181] are being increasingly used as an *encoding* model for predicting neural responses to sensory stimulation [97, 182, 98, 99]. Conversely, deep nets have also been trained to *decode* sensory-motor signals from neural activity, successfully reconstructing text [183] from Electrocorticography (ECoG) recordings, or images [184] from functional magnetic resonance imaging (fMRI). Despite these successes, we would like to argue that what possibly limits a wider impact of deep learning in human cognitive neuroscience is a combination of factors including : (i) low signal-to-noise ratio, (ii) small datasets, and (iii) a lack of high temporal resolution, where nonlinear dynamics may be the most prominent. The present experimental results make a step in this direction, and could thus open an avenue towards leveraging the many existing shorter naturalistic stimulus datasets collected on many subjects. This could be an alternative to making new long recordings of many hours of data from a handful of subjects [185].

While the DRE's architecture may be efficient at handling the dynamical structure of brain data, the dynamics assessed in this study are driven by specific linguistic features (i.e. word-length and word frequency). By contrast, recent ECoG and MEG studies have used more complex word features, represented as activations of a deep network pretrained on visual or language tasks [186–188], and then predict the brain response in a way that is unaware of the dynamics (using a linear classifier for each time sample independently). Given the successes independently observed with these two approaches, a natural extension of this work would be to combine the two and learn to map complex stimuli to brain responses using (1) rich representations for the stimuli (such as the activations of a pretrained deep network), followed by (2) a rich dynamical model such as the DRE. It is however worth pointing out that this approach would naturally lead to more high-dimensional parameter spaces, which would require larger datasets to limit potential overfitting.

The present work is based on a deterministic and predictive framework using deep learning. Other complementary approaches such as Hidden Markov Models (HMMs) and Gaussian Processes (GPs) have also been proposed to model brain data in a probabilistic framework. Such approaches have been exploited to explore the spatio-temporal statistics of fMRI or MEG data [189, 190], but also in an encoding context [191, 192]. In particular, [189] combine GPs and dynamical system modeling to account for MEG responses to tactile input and shows that it captures meaningful modulations of oscillatory activity. This approach may offer a promising avenue to further clarify the interaction between baseline alpha oscillations and visual responses captured by DRE (see Figure 2.3). Similarly, [192] show that Gaussian modeling efficiently learns to predict fMRI responses to visual stimuli, and, importantly, can be inverted to achieve zero-shot decoding of individual characters. By contrast, our encoding approach would necessitate additional fine-tuning to transfer DRE to a novel decoding task. Overall, a key advantage of probabilistic models like GP is the ability to quantify uncertainty in the predictions, which in the present forecasting scenario would likely increase when looking at late latencies. While the proposed approach does not offer this possibility, the present study benefits from the highly-optimized ecosystem of deep learning, which allows us to efficiently deal with the large size of raw MEG data (273 MEG channels sampled at 1,200 Hz and recorded for 60 min in 99 subjects).

It is worth noting that because the losses used to train the models in this paper are MSEs evaluated in the time domain, the TRF, RTRF, and DRE are solely trained and evaluated on their ability to predict the *amplitude* of the neural signal *at each time sample*. Consequently, the objective solely focuses on “evoked" activity, i.e. neural signals that are strictly phase-locked to stimulus onset or to past brain signals [163]. A quick time-frequency decomposition of the models suggest that none of them capture “induced" activity, e.g. changes in the amplitude of neural oscillations with a non-deterministic phase. A fundamentally distinct loss would be necessary to capture such non phase-locked neural dynamics.

As for many other scientific disciplines, deep neural networks will undoubtedly complement – if not shift – the myriad of analytical pipelines developed over the years toward standardized end-to-end modeling. While such methodological development may improve our ability to predict how the brain responds to various exogenous stimuli, the present attempt already highlights the many challenges that this approach entails. Nevertheless, the present results hopefully clearly demonstrate that deep networks are a very relevant technology to capture complex neural dynamics collected non-invasively by MEG and certainly EEG.

Acknowledgements Experiments on MEG data were made possible thanks to MNE [193, 164], as well as the scientific Python ecosystem : Matplotlib [194], Scikit-learn [165], Numpy [195], Scipy [196] and PyTorch [155].
This work was supported by the French ANR-20-CHIA-0016 and the European Research Council Starting Grant SLAB ERC-StG-676943 to AG, and by the French ANR-17-EURE-0017 and the Fyssen Foundation to JRK for his work at PSL.
The authors would like to thank Nicholas Tolley and Hubert Banville for their feedback and suggestions on the early versions of this manuscript.

2.7 . Supplemental Material

Figure 2.5 reports an alternative scoring of the main models using the R^2 score instead of the Pearson correlation R. It is interpretable as explained variance is a quantity upper bounded by 1, as opposed to MSE which depends on the input scaling of the data. Due to the very small SNR of single trial and unaveraged MEG recordings, we obtain low values as expected. Indeed, some variance in the signal is explained by noise only, whose amplitude is on a scale 10 times larger that of the evoked response. However, the ordering of model performance and the conclusions are left unchanged.

Note on convolution and computational efficiency The introduction of convolutional layers is here mainly motivated by computational efficiency : convolutional layers reduced training time on an NVIDIA V100 GPU from 2.6 h for a DRE devoid of convolutional layers down to 1.4 h for our DRE. The performance between these two models is relatively similar, although with a slight benefit in favour of DRE :(NO-CONV : $0.194 \pm 0.003\%$; DRE : $0.197 \pm 0.003\%$, $p < 10^{-5}$).

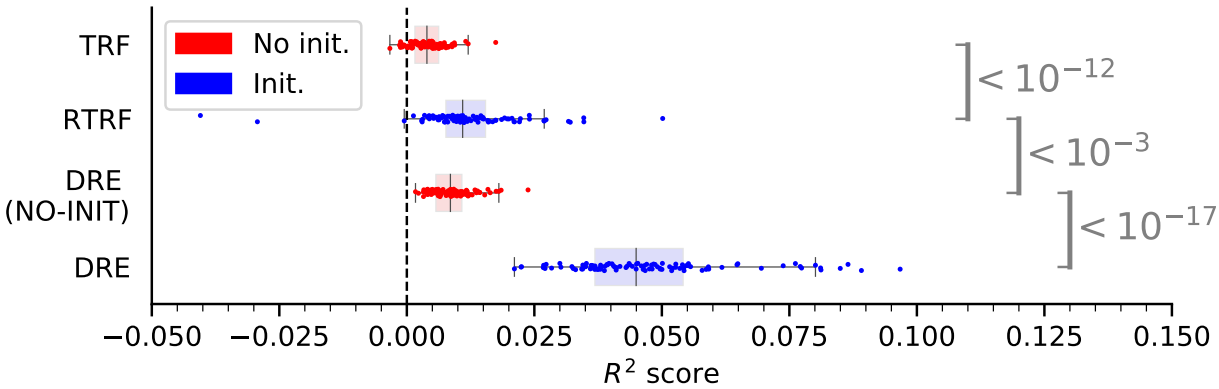


Figure 2.5 – Explained Variance captured by coefficient of determination (R^2) for each model. Boxplots indicate median and interquartile range, while each point corresponds to a single subject. The gray brackets indicate the p-value obtained with a pairwise Wilcoxon comparison across subjects. Initialization and non-linearity increase predictive performance.

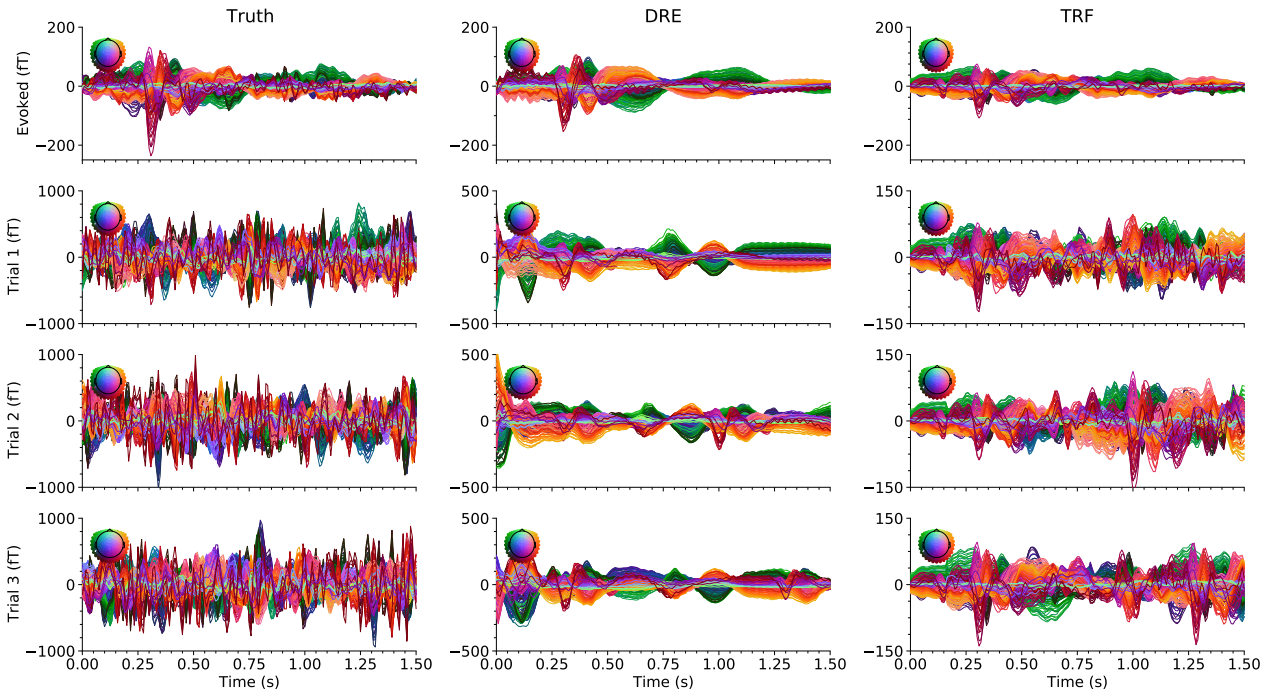


Figure 2.6 – True and predicted brain response (averaged “Evoked" (top row) or single-trial) to visual stimuli. These qualitative results illustrate that individual MEG responses vary along many different aspects, including the amplitude, latency and overall shape of the evoked-related fields. These variations are particularly challenging for TRF to learn. The predictions of TRF illustrated here also highlight that this model appears to modulate the MEG response but typically outputs predictions with reduced amplitudes – an effect likely due to a reduction-to-the-mean phenomenon : *i.e.* when the model fails to learn the MEG dynamics, it converges its coefficients towards 0.

3 - Classification Tasks on EEG data

This chapter presents the work published in :
H. Banville, O. Chehab, A. Hyvarinen, D. Engemann, A. Gramfort. *Uncovering the structure of clinical EEG signals with self-supervised learning*. Journal of Neural Engineering, 2021.

This version is shortened form of the original article. We also include some unpublished theoretical investigations in section 3.5.

3.1 . Summary

Supervised learning paradigms are often limited by the amount of labeled data that is available. This phenomenon is particularly problematic in clinically-relevant data, such as electroencephalography (EEG), where labeling can be costly in terms of specialized expertise and human processing time. Consequently, deep learning architectures designed to learn on EEG data have yielded relatively shallow models and performances at best similar to those of traditional feature-based approaches. However, in most situations, unlabeled data is available in abundance. By extracting information from this unlabeled data, it might be possible to reach competitive performance with deep neural networks despite limited access to labels. We investigated self-supervised learning (SSL), a promising technique for discovering structure in unlabeled data, to learn representations of EEG signals. Specifically, we explored two tasks based on temporal context prediction as well as contrastive predictive coding on two clinically-relevant problems : EEG-based sleep staging and pathology detection. We conducted experiments on two large public datasets with thousands of recordings and performed baseline comparisons with purely supervised and hand-engineered approaches. Linear classifiers trained on SSL-learned features consistently outperformed purely supervised deep neural networks in low-labeled data regimes while reaching competitive performance when all labels were available. Additionally, the embeddings learned with each method revealed clear latent structures related to physiological and clinical phenomena, such as age effects. We demonstrate the benefit of SSL approaches on EEG data. Our results suggest that self-supervision may pave the way to a wider use of deep learning models on EEG data.

3.2 . Self-Supervised Classification Tasks

In the following, we describe the three Contrastive tasks used in Banville et al. [197]. A visual explanation of the tasks can be found in Fig. 3.1. An implementation of one of the proposed tasks (Relative Positioning) is available in the braindecode [1] Python library [198].

1. https://github.com/braindecode/braindecode

Electroencephalography (EEG) is a non-invasive technique for recording brain activity. Typically, the electric activity of the brain is recorded from a few subjects in a controlled setting. Such a controlled experiment enables annotations (e.g. stimulus presented to a subject) [70] and metadata (e.g. age) [199] to be collected and statistically analysed in the framework of supervised statistical learning. However, it does not scale well to a large cohort of subjects : it can be both time-consuming and expensive for experts to manually annotate the recorded signals. This motivates looking beyond classical supervised approaches where few labels are available. Increasingly, EEG datasets are being shared in the public domain and larger quantities of data are made available but with few annotations. This is where unsupervised methods such as Contrastive Learning have their mark to make.

Notation We denote by $[\![q]\!]$ the set $\{1, \ldots, q\}$ and by $[\![p, q]\!]$ the set $\{p, \ldots, q\}$ for any integer $p, q \in \mathbb{N}$. The index t refers to time indices in the multivariate time series $S \in \mathbb{R}^{C \times M}$, where M is the number of time samples and C is the dimension of samples (*e.g.*, channels). We assume for simplicity that each S has the same size. We denote by $y \in \{-1, 1\}$ a binary label used in the learning task.

Relative Positioning To produce labeled samples from the multivariate time series S, we propose to sample pairs of time windows $(\boldsymbol{x}_t, \boldsymbol{x}_{t'})$ where each window $\boldsymbol{x}_t$, $\boldsymbol{x}_{t'}$ is in $\mathbb{R}^{C \times T}$ and T is the duration of each window, and where the index t indicates the time sample at which the window starts in S. The first window x_t is referred to as the "anchor window". Our assumption is that an appropriate representation of the data should evolve slowly over time (akin to the driving hypothesis behind Slow Feature Analysis (SFA) [200, 201]) suggesting that time windows close in time should share the same label. In the context of sleep staging, for instance, sleep stages usually last between 1 to 40 minutes [202]; therefore, nearby windows likely come from the same sleep stage, whereas faraway windows likely come from different sleep stages. Given $\tau_{pos} \in \mathbb{N}$, which controls the duration of the positive context, and $\tau_{neg} \in \mathbb{N}$, which corresponds to the negative context around each window x_i, we sample N labeled pairs :

$$\mathcal{Z}_N = \{((\boldsymbol{x}_{t_i}, \boldsymbol{x}_{t'_i}), y_i) \mid i \in [\![N]\!], (t_i, t'_i) \in \mathcal{T}, y_i \in \mathcal{Y}\},$$

where $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{T} = \{(t, t') \in [\![M - T + 1]\!]^2 \mid |t - t'| \leq \tau_{pos} \text{ or } |t - t'| > \tau_{neg}\}$. Intuitively, $\mathcal{T}$ is the set of all pairs of time indices (t, t') which can be constructed from windows of size T in a time series of size M, given the duration constraints imposed by the particular choices of τ_{pos} and τ_{neg} [2]. Here $y_i \in \mathcal{Y}$ is specified by the positive or negative contexts parameters :

$$y_i = \begin{cases} 1, & \text{if } |t_i - t'_i| \leq \tau_{pos} \\ -1, & \text{if } |t_i - t'_i| > \tau_{neg} \end{cases} \tag{3.1}$$

.

We ignore window pairs where $x_{t'}$ falls outside of the positive and negative contexts of the anchor window x_t. In other words, the label indicates whether two time windows are closer together than τ_{pos} or farther apart than τ_{neg} in time. Noting the connection with the task in [203], we call this pretext task "relative positioning" (RP).

2. The values of τ_{pos} and τ_{neg} can be selected based on prior knowledge of the signals and/or with a hyperparameter search.

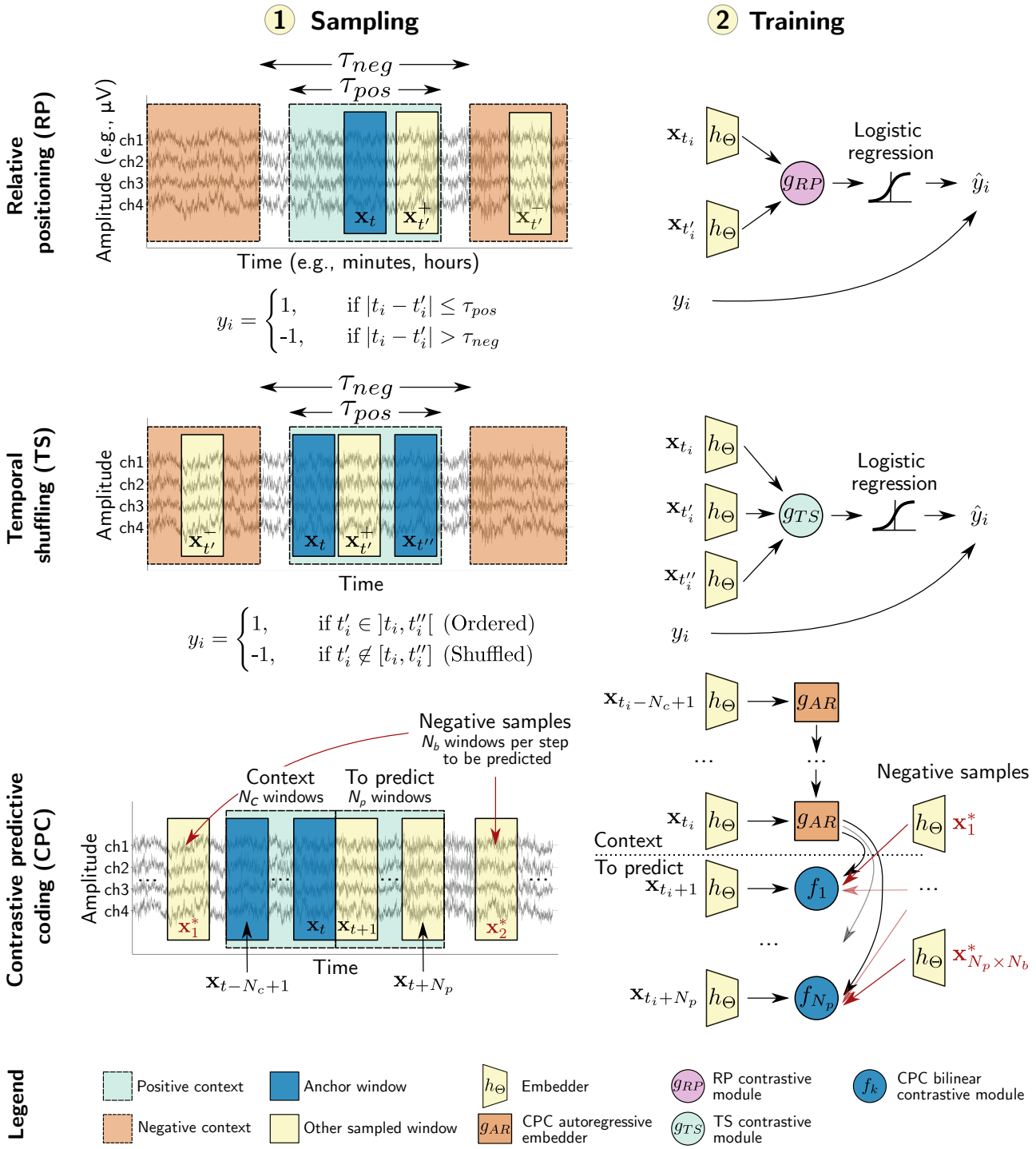


Figure 3.1 – Visual explanation of the three proposed Contrastive pretext tasks (RP, TS and CPC). The first column illustrates the sampling process by which examples are extracted from a time series S (EEG recording) in each pretext task. The second column describes the training process, where sampled examples are used to train a feature extractor h_Θ end-to-end. **RP** : Pairs of windows are sampled from S such that the two windows of a pair are either close in time ("positive pairs") or farther away ("negative pairs"). h_Θ is then trained to predict whether a pair is positive or negative. **TS** : Triplets of windows (rather than pairs) are sampled from S. A triplet is given a positive label if its windows are ordered or a negative label if they are shuffled. h_Θ is then trained to predict whether the windows of a triplet are ordered or shuffled. **CPC** : Sequences of $N_c + N_p$ consecutive windows are sampled from S along with random distractor windows ("negative samples"). Given the first N_c windows of a sequence (the "context"), a neural network is trained to identify which window out of a set of distractor windows actually follows the context.

In order to learn end-to-end how to discriminate pairs of time windows based on their relative position, we introduce two functions $\boldsymbol{h}_\Theta$ and $\boldsymbol{g}_{RP}$. $\boldsymbol{h}_\Theta : \mathbb{R}^{C\times T} \rightarrow \mathbb{R}^D$ is a feature extractor with parameters Θ which maps a window x to its representation in the feature space. Ultimately, we expect h_Θ to learn an informative representation of the raw EEG input which can be reused in different downstream tasks. A contrastive module $\boldsymbol{g}_{RP}$ is then used to aggregate the feature representations of each window. For the RP task, $g_{RP} : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^D$ combines representations from pairs of windows by computing an elementwise absolute difference, denoted by the $|\cdot|$ operator : $\boldsymbol{g}_{RP}(\boldsymbol{h}_\Theta(\boldsymbol{x}), \boldsymbol{h}_\Theta(\boldsymbol{x}')) = |\boldsymbol{h}_\Theta(x) - \boldsymbol{h}_\Theta(x')| \in \mathbb{R}^D$. The role of $\boldsymbol{g}_{RP}$ is to aggregate the feature vectors extracted by h_Θ on the two input windows and highlight their differences to simplify the contrastive task. Finally, a linear context discriminative model with coefficients $\boldsymbol{w} \in \mathbb{R}^D$ and bias term $\boldsymbol{w}_0 \in \mathbb{R}$ is responsible for predicting the associated target y. Using the binary logistic loss on the predictions of $\boldsymbol{g}_{RP}$ we can write a joint loss function $\mathcal{L}(\Theta, \boldsymbol{w}, \boldsymbol{w}_0)$ as

$$\begin{aligned}&\mathcal{L}(\Theta, \boldsymbol{w}, \boldsymbol{w}_0)\\&= \sum_{(\boldsymbol{x}_t, \boldsymbol{x}_{t'}, y)\in\mathcal{Z}_N} \log(1 + \exp(-y[\boldsymbol{w}^\top \boldsymbol{g}_{RP}(\boldsymbol{h}_\Theta(\boldsymbol{x}_t), \boldsymbol{h}_\Theta(\boldsymbol{x}_{t'})) + \boldsymbol{w}_0]))\end{aligned} \tag{3.2}$$

which we assume to be fully differentiable with respect to the parameters $(\Theta, \boldsymbol{w}, \boldsymbol{w}_0)$. Given the convention used for y, the predicted target is the sign of $\boldsymbol{w}^\top \boldsymbol{g}(\boldsymbol{h}_\Theta(\boldsymbol{x}_t), \boldsymbol{h}_\Theta(\boldsymbol{x}_{t'})) + \boldsymbol{w}_0$.

Temporal shuffling We also introduce a variation of the RP task that we call "temporal shuffling" (TS), in which we instead sample two anchor windows $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t''}$ from the positive context, and a third window $\boldsymbol{x}_{t'}$ that is either between the first two windows or in the negative context. We then construct window triplets that are either temporally ordered ($t < t' < t''$) or shuffled ($t < t'' < t'$ or $t' < t < t''$). We augment the number of possible triplets by also considering the mirror image of the previous triplets, e.g., $(\boldsymbol{x}_t, \boldsymbol{x}_{t'}, \boldsymbol{x}_{t''})$ becomes $(\boldsymbol{x}_{t''}, \boldsymbol{x}_{t'}, \boldsymbol{x}_t)$. The label y_i then indicates whether the three windows are ordered or have been shuffled, similar to [204].

The contrastive module for TS is defined as $\boldsymbol{g}_{TS} : \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^{2D}$ and is implemented by concatenating the absolute differences :

$$\boldsymbol{g}_{TS}(\boldsymbol{h}_\Theta(\boldsymbol{x}), \boldsymbol{h}_\Theta(\boldsymbol{x}'), \boldsymbol{h}_\Theta(\boldsymbol{x}'')) = (|\boldsymbol{h}_\Theta(\boldsymbol{x}) - \boldsymbol{h}_\Theta(\boldsymbol{x}')|, |\boldsymbol{h}_\Theta(\boldsymbol{x}') - \boldsymbol{h}_\Theta(\boldsymbol{x}'')|) \in \mathbb{R}^{2D} \ .$$

Moreover, Eq. 3.2 is extended to TS by replacing $\boldsymbol{g}_{RP}$ by $\boldsymbol{g}_{TS}$ and introducing $\boldsymbol{x}_{t''}$ to obtain :

$$\begin{aligned}&\mathcal{L}(\Theta, \boldsymbol{w}, \boldsymbol{w}_0)\\&= \sum_{(\boldsymbol{x}_t, \boldsymbol{x}_{t'}, \boldsymbol{x}_{t''}, y)\in\mathcal{Z}_N} \log(1 + \exp(-y[\boldsymbol{w}^\top \boldsymbol{g}_{TS}(\boldsymbol{h}_\Theta(\boldsymbol{x}_t), \boldsymbol{h}_\Theta(\boldsymbol{x}_{t'}), \boldsymbol{h}_\Theta(\boldsymbol{x}_{t''})) + \boldsymbol{w}_0])) \ .\end{aligned} \tag{3.3}$$

TS shares similarities with the unsupervised metric learning approach of [205], however the sampling procedure and loss function both differ.

Contrastive Predictive Coding The contrastive predictive coding (CPC) pretext task, introduced by Oord et al. [206], is defined here in comparison to RP and TS, as all three tasks share key similarities. Indeed, CPC can be seen as an extension of RP, where the single anchor window x_t is replaced by a

sequence of N_c non-overlapping windows that are summarized by an autoregressive encoder $\boldsymbol{g}_{AR} : \mathbb{R}^{D \times N_c} \to \mathbb{R}^{D_{AR}}$ with parameters Θ_{AR}[3]. This way, the information in the context can be represented by a single vector $c_t \in \mathbb{R}^{D_{AR}}$. $\boldsymbol{g}_{AR}$ can be implemented for example as a recurrent neural network with gated-recurrent units (GRU).

The context vector $\boldsymbol{c}_t$ is paired with not one, but N_p future windows (or "steps") which immediately follow the context. Negative windows are then sampled in a similar way as with RP and TS when $\tau_{neg} = 0$, i.e., the negative context is relaxed to include the entire time series. For each future window, N_b negative windows $\boldsymbol{x}^*$ are sampled inside each multivariate time series S ("same-recording negative sampling") or across all available S ("across-recording negative sampling"). For the sake of simplicity and to follow the notation of the original CPC article, we modify our notation slightly : we now denote a time window by $\boldsymbol{x}_t$ where t is the index of the window in the list of all non-overlapping windows of size T that can be extracted from a time series S. Therefore, the procedure for building a dataset with N examples boils down to sampling sequences X^c, X^p and X^n in the following manner :

$$\begin{aligned}
X_i^c &= (\boldsymbol{x}_{t_i - N_c + 1}, \ldots, \boldsymbol{x}_{t_i}) && (N_c \text{ context windows}) \\
X_i^p &= (\boldsymbol{x}_{t_i + 1}, \ldots, \boldsymbol{x}_{t_i + N_p}) && (N_p \text{ future windows}) \\
X_i^n &= (\boldsymbol{x}_{t^*_{i_{1,1}}}, \ldots, \boldsymbol{x}_{t^*_{i_{1,N_b}}}, \ldots, \boldsymbol{x}_{t^*_{i_{N_p,1}}}, \ldots, \boldsymbol{x}_{t^*_{i_{N_p,N_b}}}) && (N_p N_b \text{ random negative windows})
\end{aligned}$$

where $t_i \in [\![N_c, M - N_p]\!]$. We denote with t^* time indices of windows sampled uniformly at random. The dataset then reads :

$$\mathcal{Z}_N = \{(X_i^c, X_i^p, X_i^n) \mid i \in [\![N]\!]\} \ . \tag{3.4}$$

As with RP and TS, the feature extractor $\boldsymbol{h}_\Theta$ is used to extract a representation of size D from a window $\boldsymbol{x}_t$. Finally, whereas the contrastive modules $\boldsymbol{g}_{RP}$ and $\boldsymbol{g}_{TS}$ explicitly relied on the absolute value of the difference between embeddings $\boldsymbol{h}$, here for each future window $\boldsymbol{x}_{t+k}$ where $k \in [\![N_p]\!]$ a bilinear model $\boldsymbol{f}_k$ parametrized by $\boldsymbol{W}_k \in \mathbb{R}^{D \times D_{AR}}$ is used to predict whether the window chronologically follows the context $\boldsymbol{c}_t$ or not :

$$\boldsymbol{f}_k(\boldsymbol{c}_t, \boldsymbol{h}_\Theta(\boldsymbol{x}_{t+k})) = \boldsymbol{h}_\Theta(\boldsymbol{x}_{t+k})^\top \boldsymbol{W}_k \boldsymbol{c}_t \tag{3.5}$$

The whole CPC model is trained end-to-end using the InfoNCE loss [26] (a categorical cross-entropy loss) defined as

$$\begin{aligned}
&\mathcal{L}(\Theta, \Theta_{AR}, \boldsymbol{W}_k, \ldots, \boldsymbol{W}_{k+N_p-1}) \\
&= - \sum_{\substack{(\boldsymbol{X}_i^c, \boldsymbol{X}_i^p, \boldsymbol{X}_i^n) \in \mathcal{Z}_N \\ \boldsymbol{c}_{t_i} = \boldsymbol{g}_{AR}(\boldsymbol{X}_i^c)}} \sum_{k=1}^{N_p} \log \left[\frac{\exp(\boldsymbol{f}_k(\boldsymbol{c}_{t_i}, \boldsymbol{h}_\Theta(\boldsymbol{x}_{t_i+k})))}{\exp(\boldsymbol{f}_k(\boldsymbol{c}_{t_i}, \boldsymbol{h}_\Theta(\boldsymbol{x}_{t_i+k}))) + \sum_{j \in [\![N_b]\!]} \exp(\boldsymbol{f}_k(\boldsymbol{c}_{t_i}, \boldsymbol{h}_\Theta(\boldsymbol{x}_{t^*_{i_{k,j}}})))} \right]
\end{aligned} \tag{3.6}$$

While in RP and TS the model must predict *whether* a pair is positive or negative, in CPC the model must pick *which* of $N_b + 1$ windows actually follows the context. In practice, we sample batches of $N_b + 1$ sequences and for each sequence use the N_b other sequences in the batch to supply negative examples.

3. CPC's encoder $\boldsymbol{g}_{AR}$ has parameters Θ_{AR}, however we omit them from the notation for brevity.

3.3 . Evaluation setup

Downstream Tasks We performed empirical benchmarks of EEG-based contrastive tasks on two clinical problems that are representative of the current challenges in machine learning-based analysis of EEG : sleep monitoring and pathology screening. These two clinical problems commonly give rise to classification tasks, albeit with different numbers of classes and distinct data-generating mechanisms : sleep monitoring is concerned with biological events (event level) while pathology screening is concerned with single patients as compared to the population (subject level). These two clinical problems have generated considerable attention in the research community, which has led to the curation of large public databases. To enable fair comparisons with supervised approaches, we benchmarked the contrastive tasks on the Physionet Challenge 2018 [207, 208] and the TUH Abnormal EEG [209] datasets.

First, we considered sleep staging, which is a critical component of a typical sleep monitoring assessment and is key to diagnosing and studying sleep disorders such as apnea and narcolepsy [86]. Sleep staging has been extensively studied in the machine (and deep) learning literature [210–212] (approximately 10% of papers reviewed in [212]), though not through the lens of SSL. Achieving fully automated sleep staging could have a substantial impact on clinical practice as (1) agreement between human raters is often limited [213] and (2) the annotation process is time-consuming and still largely manual [214]. Sleep staging typically gives rise to a 5-class classification problem where the possible predictions are W (wake), N1, N2, N3 (different levels of sleep) and R (rapid eye movement periods). Here, the task consists of predicting the sleep stages that correspond to 30-s windows of EEG.

Second, we applied contrastive learning to pathology detection : EEG is routinely used in a clinical context to screen individuals for neurological conditions such as epilepsy and dementia [87, 88]. However, successful pathology detection requires highly specialized medical expertise and its quality depends on the expert's training and experience. Automated pathology detection could, therefore, have a major impact on clinical practice by facilitating neurological screening. This gives rise to classification tasks at the subject level where the challenge is to infer the patient's diagnosis or health status from the EEG recording. In the TUH dataset, medical specialists have labeled recordings as either pathological or non-pathological, giving rise to a binary classification problem. Importantly, these two labels reflect highly heterogeneous situations : a pathological recording could reflect anomalies due to various medical conditions, suggesting a rather complex data-generating mechanism. Again, various supervised approaches, some of them leveraging deep architectures, have addressed this task in the literature [215–217], although none has relied on contrastive learning formulated as a self-supervised task.

The two aforementioned EEG datasets are described in Tables 3.1 and 3.2.

3.4 . Results

Results : all three Contrastive Tasks recover physiologically meaningful features The three tasks we consider – Relative Positioning (RP), Temporal Shuffling (TS), and Contrastive Predictive Coding (CPC) – learn representations of the data that encode the *same* sleep-stage structure that is

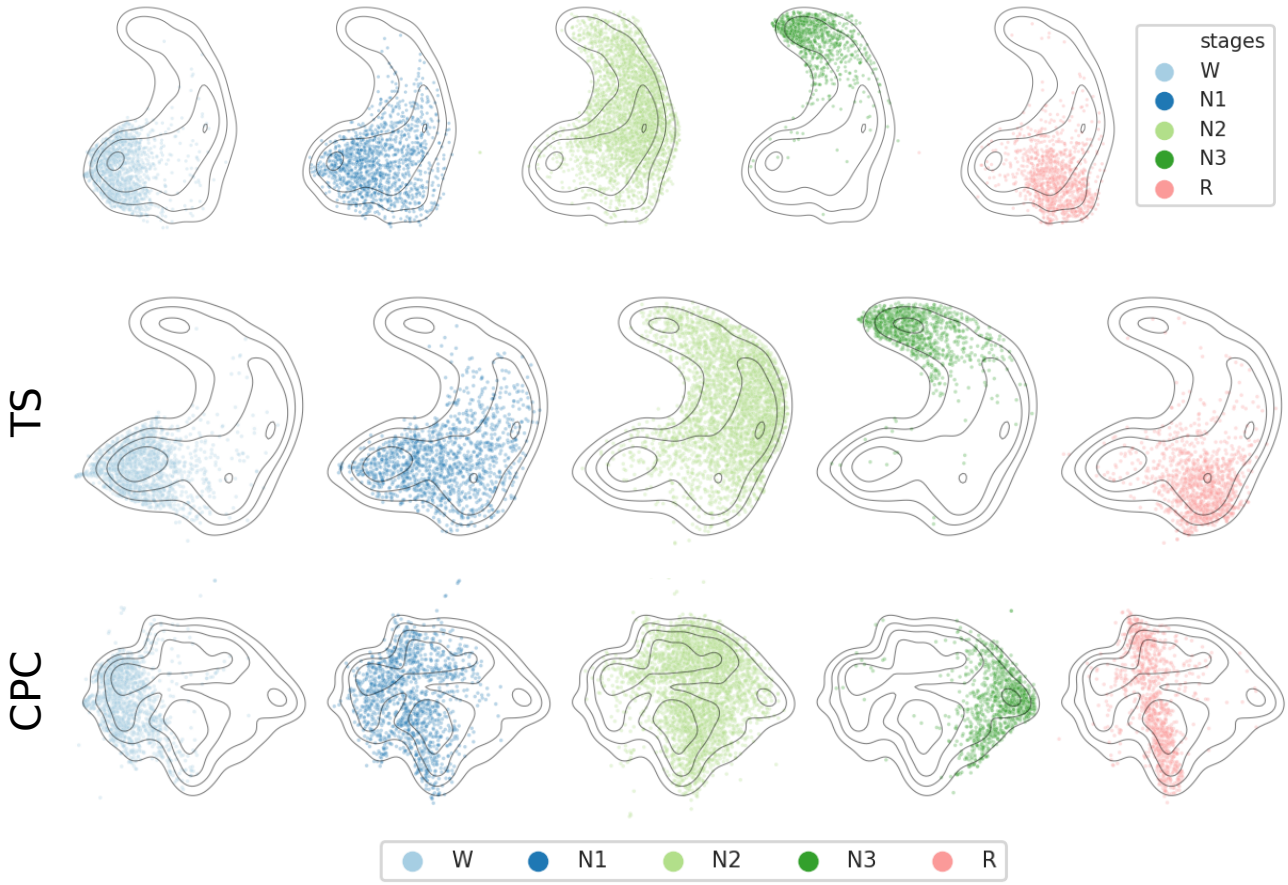


Figure 3.2 – UMAP visualization of Contrastive Learning features on the PC18 dataset. The subplots show the distribution of the 5 sleep stages as scatterplots for RP (first row), TS (second row) and CPC (third row) features. Contour lines correspond to the density levels of the distribution across all stages and are used as visual reference. Finally, each point corresponds to the features extracted from a 30-s window of EEG by the RP, TS and CPC embedders with the highest downstream performance. All available windows from the train, validation and test sets of PC18 were used. In each case, there is clear structure related to sleep stages although no labels were available during training.

PC18 (train)			
	# windows		
W	158,020	# unique subjects	994
N1	136,858	# recordings	994
N2	377,426	Sampling frequency	200 Hz
N3	102,492	# EEG channels	6
R	116,872	Reference	M1 or M2
Total	891,668		

Table 3.1 – Description of the Physionet Challenge 2018 (PC18) dataset used in this study for sleep staging experiments.

TUHab				
	train	eval	# unique subjects	2329
	# recordings	# recordings	# recordings	2993
Normal	1371	150	Sampling frequency	250, 256, 512 Hz
Abnormal	1346	126	# EEG channels	27 to 36
Total	2717	276	Reference	Common average

Table 3.2 – Description of the TUH Abnormal (TUHab) dataset used in this study for EEG pathology detection experiments.

known to practitioners. This is seen in Figure 3.2. The difference is, instead of a discrete partition into five stages (W, N1, N2, N3, R), the Contrastive Tasks learn a *continuous* geometry in the latent space, opening up the possibility of a more fine-grained organization of sleep. What is more, Figure 3.3 shows that RP, as well as TS and CPC (not shown), organize the latent space in such a way that physiologically meaningful features, such as age, gender, pathology and apnea, can be discriminated. These would typically be labels in a supervised setting : it is remarkable that a Contrastive Task designed with only a *general* notion of structure, basically recognizing temporal order, is able to learn *finer* structures such as age and sleep stages as summary statistics of sorts. While it is not clear the minimization of the RP, TS and CPC losses drive a similar latent space geometry, Figure 3.2 shows that as far as sleep-stage clustering goes, the features are semantically similar. In part II , we will choose to focus then on a simplified version of RP, the simplest of the three Contrastive Tasks considered here.

How to choose the Contrastive Hyperparameters Having empirically established the success of the Contrastive Tasks at recovering features that are useful for various downstream tasks (e.g. sleep-staging), the next step is to ask : how can we optimize these methods? Given an anchor $\boldsymbol{x}_t$, the most obvious hyperparameters are perhaps the time lags τ_{pos} and τ_{neg}, from which to sample positive and negative samples. Figure 3.5 shows that τ_{pos} is best chosen so that the positive pair is closest in time : this is most striking in the top-left graph. Interestingly, this supports the choice made by Permutation-Contrastive Learning (PCL) [16], where the positive windows are $(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$. What of

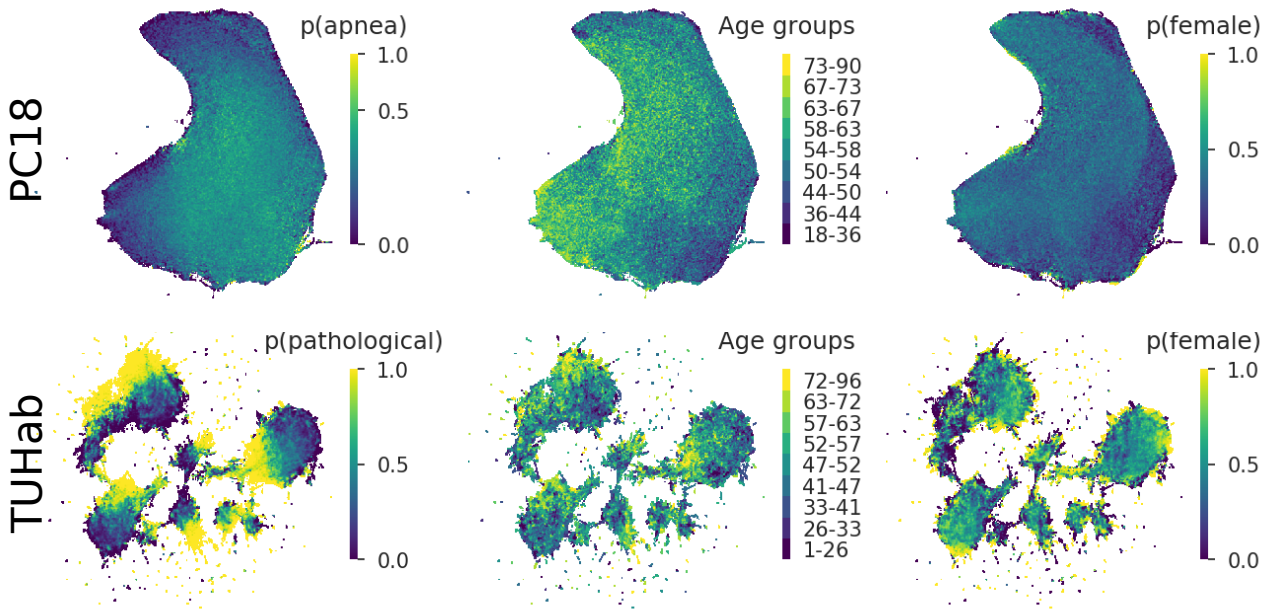


Figure 3.3 – Structure learned by the embedders trained on the RP task. The models with the highest downstream performance were used to embed the combined train, validation and test sets of the PC18 and TUHab datasets. The embeddings were then projected to two dimensions using UMAP and discretized into 500 x 500 "pixels". For binary labels ("apnea", "pathological" and "gender"), we visualize the probability as heatmaps, i.e., the color indicates the probability that the label is true (*e.g.*, that a window in that region of the embedding overlaps with an apnea annotation). For age, the subjects of each dataset were divided into 9 quantiles, and the color indicates which group was the most frequent in each bin. The features learned with Contrastive Learning capture physiologically-relevant structure, such as pathology, age, apnea and gender. Note that the different clusters in the second row correspond to different experimental setups, as seen in Figure 3.4

.

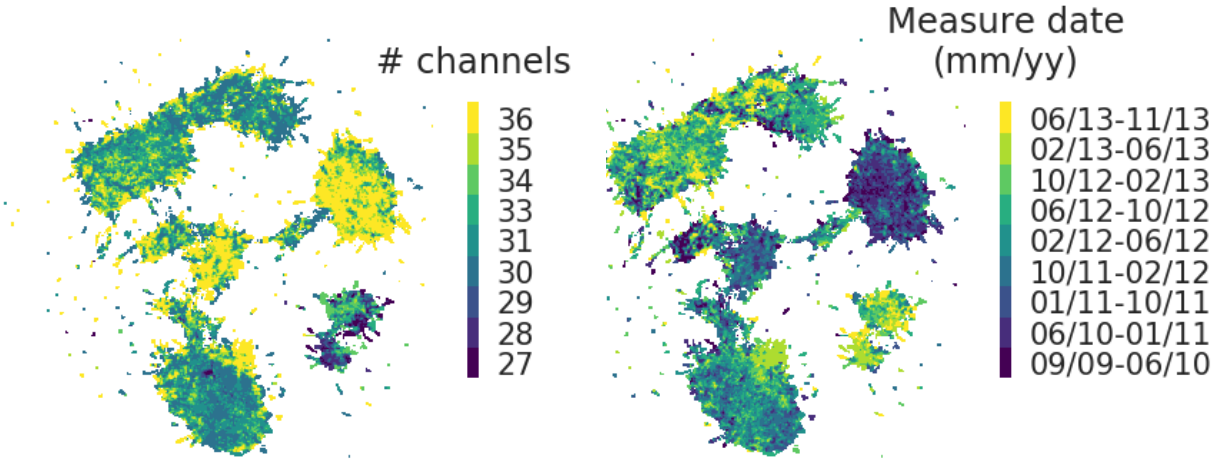


Figure 3.4 – Structure related to the original recording's number of EEG channels and measurement date in learned features on the entire TUHab dataset. The overall different number of EEG channels and measurement date in each cluster reveals that the cluster-like structure reflects differences in experimental setups.

the negative pair? Figure 3.5 provides little insight into the optimal negative pair of samples, given that different choices of τ_{neg} yield roughly equivalent performances on the downstream task. Interestingly, the Figure (row : CPC; columns : 'sampling') does show that downstream performance can remain constant as the Pretext Contrastive Tasks yield greatly varying performances. This 'disconnect' between Pretext and Downstream performance, suggesting that they may not vary proportionally to each other, is a topic we have just started to explore.

Let us conclude this chapter with a question that remains unanswered : what is the optimal noise distribution for the Relative Positioning Task? Chapter 4 tackles this question in a simplified but principled framework.

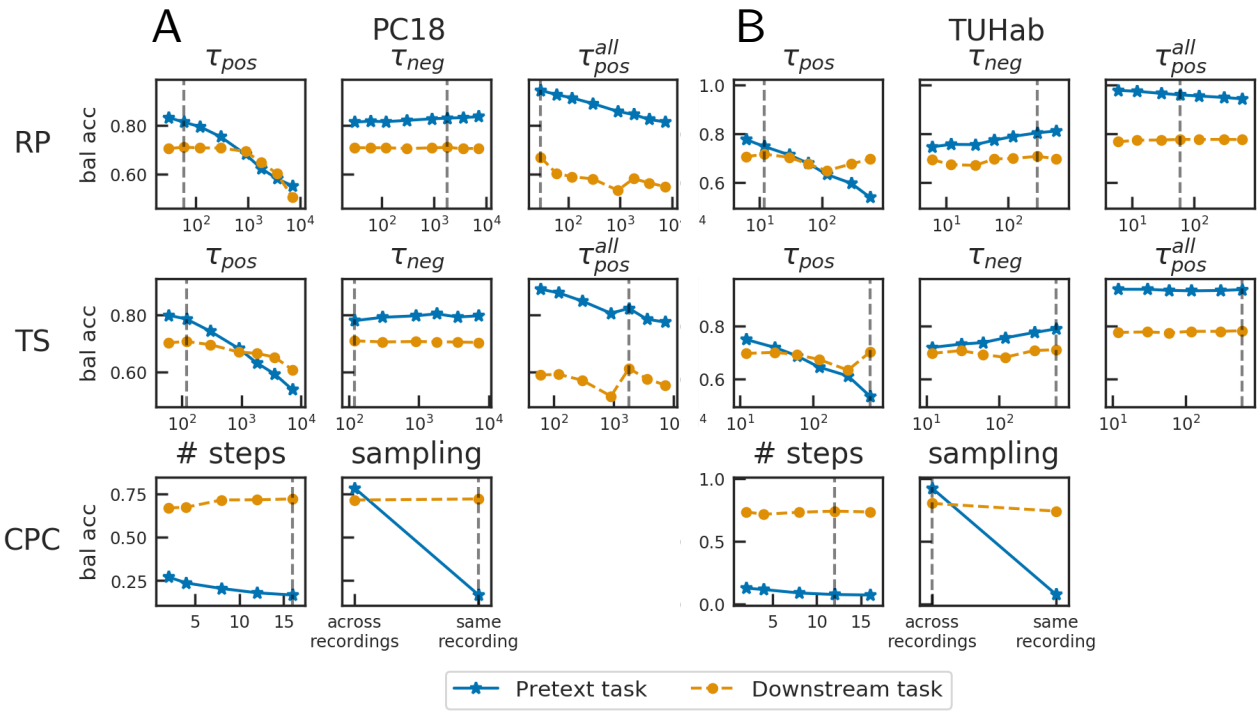


Figure 3.5 – Impact of principal hyperparameters on pretext (blue) and downstream (yellow) task performance, measured with balanced accuracy on the validation set of (A) PC18 and (B) TUHab. Each row corresponds to a different Contrastive Learning pretext task. For both RP and TS, we varied the hyperparameters that control the length of the positive and negative contexts (τ_{pos}, τ_{neg}, in seconds); the exponent "all" indicates that negative windows were sampled across all recordings instead of within the same recording. For CPC, we varied the number of predicted windows and the type of negative sampling. Finally, the best hyperparameter values in terms of downstream task performance are emphasized using vertical dashed lines.

3.5 . Toward a Probabilistic Interpretation of Classification Tasks

We here sketch out how the self-supervised tasks presented in Banville et al. [197] could be motivated as probabilistic objectives. This work is not included in the publication but may serve as a starting point for future research.

Relative Positioning This task is a classification task between pairs of windows from a timeseries $\boldsymbol{x}(t)$. Nearby windows form a class $Y = 1$ and random windws form the other class $Y = 0$. The decision model $F(\boldsymbol{x})$ that was used in the experiments [197, Eq. 2] was

$$F(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{w}, \boldsymbol{w}_0, \boldsymbol{\Theta}) = \sum_{d=1}^{D} w^d |h_{\boldsymbol{\Theta}}(\boldsymbol{x}_1)^d - h_{\boldsymbol{\Theta}}(\boldsymbol{x}_2)^d| + w_0^d \tag{3.7}$$

using the notations from the paper. Recall that d indexes the components of a vector. We know from classification theory that the optimal decision function is

$$F^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \log(p(\boldsymbol{x}_1, \boldsymbol{x}_2|Y = 1)/p(\boldsymbol{x}_1, \boldsymbol{x}_2|Y = 0)) \tag{3.8}$$

We know from classification theory that the logistic loss used in the paper corresponds to matching the model and true ratios, $\exp(F)$ and $\exp(F^*)$, with a Bregman divergence. We want to equalize these expressions, to find the statistical model p on the sources. We now make assumptions on p that will conveniently make these expressions equal, and later discuss the validity of these assumptions.

Assumption 1 : bijective forward model The timeseries is generated by a forward model $\boldsymbol{x}(t) = f(\boldsymbol{z}(t))$ where the mapping f is bijective and $\boldsymbol{z}(t)$ is a "source" timeseries. This means that $p(\boldsymbol{x}_1, \boldsymbol{x}_2|Y = 1) = p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 1) \times |J_g(\boldsymbol{x}_1)| \times |J_g(\boldsymbol{x}_2)|$ where $g = f^{-1}$. And implies that

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 1)}{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 0)} \tag{3.9}$$

Assumption 2 : independence The components of the representation $\left(\boldsymbol{z}^i(t)\right)_{i\in[\![1,D]\!]}$ are mutually independent processes. This implies

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \prod_{d=1}^{D} \frac{p_z(z_1^d, z_2^d|Y = 1)}{p_z(z_1^d, z_2^d|Y = 0)} \tag{3.10}$$

Assumption 3 : stationarity The components of the representation each follow a stationary process : the marginal distribution is the same $p_z(z_1^d|Y = 1) = p_z(z_2^d|Y = 0)$. This implies

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \prod_{d=1}^{D} \frac{p_z(z_2^d|z_1^d|Y = 1)}{p_z(z_2^d|z_1^d|Y = 0)} \tag{3.11}$$

Assumption 4 : Laplace innovations The components of the representation are conditionally driven by Laplace innovations $z_2^d|z_1^d, Y = y \sim \mathrm{Laplace}(\mathrm{location} = z_1^d, \mathrm{scale} = s_y^d)$. Note that we suppose that the location and scale are respectively data-dependent and label-dependent, only. We then obtain

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \prod_{d=1}^{D} \frac{(1/2s_1)\exp(-|z_2^d - z_1^d|/s_1^d)}{(1/2s_0)\exp(-|z_2^d - z_1^d|/s_0^d)} \tag{3.12}$$

Assumption 5 : identifiability We assume that ratio model being used satisfies some identifiability conditions, so that equalizing r^* and r leads to term-to-term identification. Then,

$$w_0^d = \log s_0^d / s_1^d \tag{3.13}$$

$$w^d = \frac{1}{s_0^d} - \frac{1}{s_1^d} \tag{3.14}$$

the coefficient w_0^d recovers the relative scale between classes, and the sign of the coefficient w^d says which scale is bigger.

These five assumptions define a statistical model of sources for which the choice of classifier in Eq. 3.7 is not mis-specified. There remains to check whether these assumptions are mutually compatible, or if some contradict others. We do not check this here and instead leave this as an intial sketch-of-proof that could be used for a future project.

Temporal Shuffling This task is a classification task between triplets of windows from a timeseries $\boldsymbol{x}(t)$. Each class $Y = 1$ and $Y = 0$ is obtained by sampling triplets of windows in a different way; we refer the reader to section 3.2 for details on the sampling strategy that are not necessary to understand our sketch of proof. The decision model $F(\boldsymbol{x})$ that was used in the experiments [197, Eq. 3] was

$$F(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{w}, \boldsymbol{w}_0, \boldsymbol{\Theta}) \tag{3.15}$$

$$= \sum_{d=1}^{D} w'^d |h_{\boldsymbol{\Theta}}(\boldsymbol{x}_2)^d - h_{\boldsymbol{\Theta}}(\boldsymbol{x}_1)^d| + w''^d |h_{\boldsymbol{\Theta}}(\boldsymbol{x}_3)^d - h_{\boldsymbol{\Theta}}(\boldsymbol{x}_2)^d| + w_0'^d + w_0''^d \tag{3.16}$$

using the notations from the paper. We know from classification theory that the optimal decision function is

$$F^*(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \log(p(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 | Y = 1) / p(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 | Y = 0)) \tag{3.17}$$

We know from classification theory that the logistic loss used in the paper corresponds to matching the model and true ratios, $\exp(F)$ and $\exp(F^*)$, with a Bregman divergence. We want to equalize these expressions, to find the statistical model p on the sources. We now make assumptions on p that will conveniently make these expressions equal, and later discuss the validity of these assumptions.

Assumption 1 : bijective forward model The timeseries is generated by a forward model $\boldsymbol{x}(t) = f(\boldsymbol{z}(t))$ where the mapping f is bijective and $\boldsymbol{z}(t)$ is a "source" timeseries. This means that $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 | Y = 1) = p_z(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3 | Y = 1) \times |J_g(\boldsymbol{x}_1)| \times |J_g(\boldsymbol{x}_2)| \times |J_g(\boldsymbol{x}_3)|$ where $g = f^{-1}$. And implies that

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \frac{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3 | Y = 1)}{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3 | Y = 0)} \tag{3.18}$$

Assumption 2 : independence The components of the representation $\left(\boldsymbol{z}^i(t)\right)_{i \in [\![1,D]\!]}$ are mutually independent processes. This implies

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \prod_{d=1}^{D} \frac{p_z(z_1^d, z_2^d, z_3^d | Y = 1)}{p_z(z_1^d, z_2^d, z_3^d | Y = 0)} \tag{3.19}$$

Assumption 3 : stationarity and graphical model The components of the representation each follow a stationary process : the marginal distribution is the same $p_z(z_1^d|Y=1) = p_z(z_1^d|Y=0)$. Additionally, we suppose a graphical model $\boldsymbol{z}_1 \to \boldsymbol{z}_2 \to \boldsymbol{z}_3$ which implies that $p(\boldsymbol{z}_3|\boldsymbol{z}_1, \boldsymbol{z}_2, Y=j)$ does not depend on $\boldsymbol{z}_1$. It follows,

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \prod_{d=1}^{D} \frac{p_z(z_3^d|z_2^d|Y=1) \times p_z(z_2^d|z_1^d|Y=1)}{p_z(z_3^d|z_2^d|Y=0) \times p_z(z_2^d|z_1^d|Y=0)} \tag{3.20}$$

Assumption 4 : Laplace innovations The components of the representation are conditionally driven by Laplace innovations $z_2^d|z_1^d, Y=y \sim \mathrm{Laplace}(\mathrm{location} = z_1^d, \mathrm{scale} = s_y^{'d})$ and $z_3^d|z_2^d, Y=y \sim \mathrm{Laplace}(\mathrm{location} = z_2^d, \mathrm{scale} = s_y^{''d})$. Note that we suppose that the location and scale are respectively data-dependent and label-dependent, only.

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \prod_{d=1}^{D} \frac{(1/2s_1^{'d})\exp(-|z_2^d - z_1^d|/s_1^{'d}) \times (1/2s_1^{''})\exp(-|z_3^d - z_2^d|/s_1^{''d})}{(1/2s_0^{'d})\exp(-|z_2^d - z_1^d|/s_0^{'d}) \times (1/2s_0^{''d})\exp(-|z_3^d - z_2^d|/s_0^{''d})} \tag{3.21}$$

Assumption 5 : identifiability We assume that ratio model being used satisfies some identifiability conditions, so that equalizing r^* and r leads to term-to-term identification. Then,

$$w_0^{'d} = \log s_0^{'d}/s_1^{'d} \tag{3.22}$$

$$w^{'d} = \frac{1}{s_0^{'d}} - \frac{1}{s_1^{'d}} \tag{3.23}$$

$$w_0^{''d} = \log s_0^{''d}/s_1^{''d} \tag{3.24}$$

$$w^{''d} = \frac{1}{s_0^{''d}} - \frac{1}{s_1^{''d}} \tag{3.25}$$

similar to Relative Positioning.

Again, these five assumptions define a statistical model of sources for which the choice of classifier in Eq. 3.16 is not mis-specified. There remains to check whether these assumptions are mutually compatible, or if some contradict others. We do not check this here and instead leave this as an intial sketch-of-proof that could be used for a future project.

Contrastive Predictive Coding This task uses a classification objective that is different to binary classification, but can nevertheless be interpreted as matching log density ratios [26]. In the following, we assume the integer $k > 0$ is fixed, for simplicity. The decision model used in the experiments was

$$F(\boldsymbol{z}_t, \boldsymbol{z}_{t+k}; \boldsymbol{W}_{k+1}) = \boldsymbol{z}_t^\top \boldsymbol{W}_k \boldsymbol{z}_{t+k} \tag{3.26}$$

and the true decision model is

$$F^*(\boldsymbol{x}_t, \boldsymbol{x}_{t+k}) = \log \frac{p(\boldsymbol{x}_t, \boldsymbol{x}_{t+k}|Y=1)}{p(\boldsymbol{x}_t, \boldsymbol{x}_{t+k}|Y=0)} \tag{3.27}$$

We can re-write the model ratio as

$$r(\boldsymbol{z}_t, \boldsymbol{z}_{t+k}; \boldsymbol{W}_{k+1}) = \exp\left(\begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{z}_{t+k} \end{bmatrix}^\top \begin{bmatrix} \epsilon\mathbf{Id} & \boldsymbol{W}_k \\ \boldsymbol{W}_k & \epsilon\mathbf{Id} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{z}_{t+k} \end{bmatrix} \right) . \tag{3.28}$$

We know from the classification theory in section 1.2 that the loss used in the paper corresponds to matching the model and true ratios, $\exp(F)$ and $\exp(F^*)$, with a Bregman divergence. We want to equalize these expressions, to find the statistical model p on the sources. We now make assumptions on p that will conveniently make these expressions equal, and later discuss the validity of these assumptions.

Assumption 1 : bijective forward model The timeseries is generated by a forward model $\boldsymbol{x}(t) = f(\boldsymbol{z}(t))$ where the mapping f is bijective and $\boldsymbol{z}(t)$ is a "source" timeseries. This means that $p(\boldsymbol{x}_1, \boldsymbol{x}_2|Y = 1) = p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 1) \times |J_g(\boldsymbol{x}_1)| \times |J_g(\boldsymbol{x}_2)|$ where $g = f^{-1}$. And implies that

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 1)}{p_z(\boldsymbol{z}_1, \boldsymbol{z}_2|Y = 0)} \tag{3.29}$$

Assumption 2 : stationarity and graphical model The components of the representation follow a stationary, gaussian process, such that $(\boldsymbol{z}_t, \boldsymbol{z}_{t+k})|y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_{\mathbf{y}})$. Note that the covariance depends on the label (which defines the sampling of time indices) only, not the time, hence stationarity. The ratio is then

$$r^*(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto \exp\Big(-\frac{1}{2} \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{z}_{t+k} \end{bmatrix}^\top (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_0) \begin{bmatrix} \boldsymbol{z}_t \\ \boldsymbol{z}_{t+k} \end{bmatrix} \Big) \tag{3.30}$$

Assumption 3 : identifiability We assume that ratio model being used satisfies some identifiability conditions, so that equalizing r^* and r leads to term-to-term identification. Then,

$$\boldsymbol{\Lambda}_0 - \boldsymbol{\Lambda}_1 = \begin{bmatrix} \epsilon \mathbf{Id} & \boldsymbol{W}_k \\ \boldsymbol{W}_k & \epsilon \mathbf{Id} \end{bmatrix} \quad . \tag{3.31}$$

In the limit of $\epsilon \to 0$, which means that $(\boldsymbol{z}_t, \boldsymbol{z}_{t+k})$ have a marginal variances but high co-variance, the model and true ratios match.

Again, these five assumptions define a statistical model of sources for which the choice of classifier in Eq. 3.26 is not mis-specified. There remains to check whether these assumptions are mutually compatible, or if some contradict others. Importantly, there is no assumption on independence between sources. We do not check these assumptions here and instead leave this as an intial sketch-of-proof that could be used for a future project.

Discussion It was observed in Figure 3.2 that the representations learnt using RP and TS have a similar UMAP reduction, while CPC does not. This could be due to the UMAP reduction itself, which is inconsistent across seeds [218]. Another explanation however, could be that the representations learnt using RP and TS are similarly distributed, whereas those learnt with CPC are not. This would be supported by the proof-sketches which suggest that the representations learnt with RP and TS are both autoregressive processes driven by Laplace innovations, whereas those learnt with CPC follow a Gaussian process. What is clear is that the choice of analytical formula for the classifier(Eq. 3.7, Eq. 3.16, Eq. 3.26) makes an assumption on the density ratio between classes. This is particularly relevant as these classifier models were chosen by trial-and-error, as we had noticed during experiments that using the "absolute value of the difference" (Eq. 3.7, Eq. 3.16) was crucial for learning compared with other classifier choices.

Deuxième partie

Statistical Analysis

4 - Noise-Contrastive Estimation

This section presents the works published in :
O. Chehab, A. Gramfort, A. Hyvarinen. *The Optimal Noise in Noise-Contrastive Learning Is Not What You Think*. Uncertainty in Artificial Intelligence (UAI), 2022.

4.1 . Summary

Learning a parametric model of a data distribution is a well-known statistical problem that has seen renewed interest as it is brought to scale in deep learning. Framing the problem as a self-supervised task, where data samples are discriminated from noise samples, is at the core of state-of-the-art methods, beginning with Noise-Contrastive Estimation (NCE). Yet, such contrastive learning requires a good noise distribution, which is hard to specify; domain-specific heuristics are therefore widely used. While a comprehensive theory is missing, it is widely assumed that the optimal noise should in practice be made equal to the data, both in distribution and proportion; this setting underlies Generative Adversarial Networks (GANs) in particular. Here, we empirically and theoretically challenge this assumption on the optimal noise. We show that deviating from this assumption can actually lead to better statistical estimators, in terms of asymptotic variance. In particular, the optimal noise distribution is different from the data's and even from a different family.

4.2 . Introduction

Learning a parametric model of a data distribution is at the core of statistics and machine learning. Once a model is learnt, it can be used to generate new data, to evaluate the likelihood of existing data, or be introspected for meaningful structure such as conditional dependencies between its features. Among an arsenal of statistical methods developed for this problem, Maximum-Likelihood Estimation (MLE) has stood out as the go-to method : given data samples, it evaluates a model's likelihood to have generated them and retains the best fit. However, MLE is limited by the fact that the parametric model has to be properly normalized, which may not be computationally feasible.

In recent years, an alternative has emerged in the form of Noise-Contrastive Estimation (NCE) [11] : given data samples, it generates noise samples and trains a discriminator to learn the data distribution by constrast. Its supervised formulation, as a binary prediction task, is simple to understand and easy to implement. In fact, NCE can be seen as one of the first and most fundamental methods of *self-supervised* learning, which has seen a huge increase of interest recently [26, 219].

Crucially, NCE can handle unnormalized, i.e. energy-based, models. It has shown remarkable success in Natural Language Processing [220, 221] and has spearheaded an entire family of contrastive methods [63, 38, 222, 42, 223, 26].

While MLE is known to be optimal in the asymptotic limit of infinite samples, NCE is a popular choice in practice due to its computational advantage. In fact, NCE outperforms Monte Carlo Maxi-

mum Likelihood (MLE-MC) [224] - an MLE estimation procedure where normalization is performed by importance sampling.

Nevertheless, NCE's performance is however dependent on two hyperparameters : the choice of noise distribution and the noise-data ratio (or, proportion of noise samples) [11]. A natural question follows : what is the optimal choice of noise distribution, and proportion of noise (or, noise-data ratio) for learning the data distribution ? There are many heuristics for choosing the noise distribution and ratio in the NCE setting. Conventional wisdom in the related setting of GANs and variants [223, 225] is to set both the proportion and the distribution of noise to be equal to those of the data. The underlying assumption is a game-theoretic notion of optimality : the task of discriminating data from noise is hardest, and therefore most "rewarding", when noise and data are virtually indistinguishable. The noise would then be optimal when the discriminator is no longer able to distinguish noise samples from data samples.

However, such an adversarial form of training where a noise generator aims to fool the discriminator suffers from instability and mode-collapse [226, 227]. Furthermore, while the above assumptions (optimal noise equals data) have been supported by numerous empirical successes, it is not clear whether such a choice of noise (distribution and ratio) achieves optimality from a statistical estimation viewpoint. In fact, the original NCE paper [11] already challenges this assumption by empirically showing that an unbalanced noise-data proportion can reduce the estimation error. Since NCE is fundamentally motivated by parameter estimation, the optimization of hyperparameters should logically be based on that same framework.

In this work, we propose a principled approach for choosing the optimal noise distribution and ratio while challenging, both theoretically and empirically, the current practice. In particular, we make the following claims that challenge conventional wisdom :

1. The optimal noise distribution is not the data distribution ; in fact, it is of a very different family than the model family.
2. The optimal noise proportion is generally not 50% ; the optimal noise-data ratio is not one.

The paper is organized as follows. First, we present NCE and related works in Section 4.3, as well as the theoretical framework of asymptotic MSE that we use to optimize the NCE estimator. We start Section 4.4 by empirically showing that the optimal noise distribution is not the data distribution. Our main theoretical results describing the optimal noise distribution are in Section 4.4. Specifically, we analytically provide the optimal noise for NCE in two interesting limits, and numerically verify how optimal that optimal noise remains outside these limits. We further show empirically that the optimal noise proportion is not 50% either. Finally we discuss the limitations of this work in Section 4.6 and conclude in Section 4.6.

Notation We denote with p_d a data distribution, p_n a noise distribution, and $(p_{\boldsymbol{\theta}})_{\boldsymbol{\theta}\in\Theta}$ a parametric family of distributions assumed to contain the data distribution $p_d = p_{\boldsymbol{\theta}^*}$. All distributions are normalized, meaning that the NCE estimator does not consider the normalizing constant as a parameter to be estimated to simplify the analysis : in this setup, NCE can be fairly compared to MLE and the Cramer-Rao bound is well-defined and applicable. The logistic function is denoted by $\sigma(x)$. We will denote by ν the ratio between the number of noise samples and data samples : $\nu = T_n/T_d$. The notation $\langle x, y\rangle_{\boldsymbol{A}} := \langle x, \boldsymbol{A}y\rangle$ refers to the inner product with metric $\boldsymbol{A}$. The induced norm is $\|\boldsymbol{x}\|_{\boldsymbol{A}} := \|\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{x}\|$.

4.3 . Background

Definition of NCE Noise-Contrastive Estimation consists in approximating a data distribution p_d by training a discriminator $D(\boldsymbol{x})$ to distinguish data samples $(\boldsymbol{x}_i)_{i\in[1,T_d]} \sim p_d$ from noise samples $(\boldsymbol{x}_i)_{i\in[1,T_n]} \sim p_n$ [11]. This defines a binary task where $Y = 1$ is the data label and $Y = 0$ is the noise label. The discriminator is optimal when it equals the (Bayes) posterior

$$D(\boldsymbol{x}) = P(Y = 1|X) = \sigma\left(\frac{p_d(\boldsymbol{x})}{\nu p_n(\boldsymbol{x})}\right) \tag{4.1}$$

i.e. when it learns the density-ratio $\frac{p_d}{p_n}$ [11, 228]. The basic idea in NCE is that replacing in the ratio the data distribution by $p_{\boldsymbol{\theta}}$ and optimizing a discriminator with respect to θ, yields a useful estimator $\hat{\boldsymbol{\theta}}_{\mathrm{NCE}}$ because at the optimum, the model density has to then equal the data density.

Importantly, there is no need for the model to be normalized; the normalization constant (partition function) can be input as an extra parameter, in stark contrast to MLE.

Asymptotic analysis We consider here a very well-known framework to analyze the statistical performance of an estimator. Fundamentally, we are interested in the Mean-Squared Error (MSE), generally defined as

$$\mathbb{E}_{\boldsymbol{\theta}}[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2] = \mathrm{Var}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) + \mathrm{Bias}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})^2$$

It can mainly be analyzed in the asymptotic regime, with the number of data points T_d being very large. For (asymptotically) unbiased estimators, the estimator's statistical performance is in fact completely characterized by its asymptotic variance (or rather, covariance matrix) because the bias squared is of a lower order for such estimators. The asymptotic variance is classically defined as

$$\boldsymbol{\Sigma} = \lim_{T_d\to\infty} T_d\, \mathbb{E}_{\boldsymbol{\theta}}[(\hat{\boldsymbol{\theta}} - \mathbb{E}_{\boldsymbol{\theta}}[\hat{\boldsymbol{\theta}}])(\hat{\boldsymbol{\theta}} - \mathbb{E}_{\boldsymbol{\theta}}[\hat{\boldsymbol{\theta}}])^\top] \tag{4.2}$$

where the estimator is evaluated for each sample size T_d separately. Thus, we use the asymptotic variance to compute an asymptotic approximation of the total Mean-Squared Error which we define as

$$\mathrm{MSE} = \frac{1}{T_d}\mathrm{tr}(\boldsymbol{\Sigma}) \ . \tag{4.3}$$

In the following, we talk about MSE to avoid any confusion regarding the role of bias : we emphasize that the MSE is given by the asymptotic variance since the bias squared is of a lower order (for consistent estimators, and under some technical constraints). Furthermore, the MSE is always defined in the asymptotic sense as in Eqs. (4.2) and (4.3).

When considering normalized distributions, classical statistical theory tells us that the best attainable MSE (among unbiased estimators) is the Cramer-Rao bound, achieved by Maximum-Likelihood Estimation (MLE). This provides a useful baseline, and implies that $\mathrm{MSE}_{\mathrm{NCE}} \geq \mathrm{MSE}_{\mathrm{MLE}}$ necessarily.

In contrast to a classical statistical framework, however, we consider here the case where the bottleneck of the estimator is the computation, while data samples are abundant. This is the case in many modern machine learning applications. The computation can be taken proportional to the total number of data points used, real data and noise samples together, which we denote by $T = T_d + T_n$. Still, the same asymptotic analysis framework can be used.

An asymptotic analysis of NCE has been carried out by Gutmann and Hyvärinen [11]. The MSE of NCE depends on three design choices (hyperparameters) of the experiment :

- the noise distribution p_n
- the noise-data ratio $\nu = T_n/T_d$, from which the noise proportion can be equivalently calculated
- the total number of samples $T = T_d + T_n$, corresponding here to the computational budget

Building on theorem 3 of Gutmann and Hyvärinen [11], we can write $\mathrm{MSE}_{\mathrm{NCE}}$ as a function of T (not T_d) to enforce a finite computational budget, giving

$$\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) = \frac{\nu+1}{T}\mathrm{tr}(\boldsymbol{I}^{-1} - \frac{\nu+1}{\nu}(\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1})) \tag{4.4}$$

where $\boldsymbol{m}$ and $\boldsymbol{I}$ are a generalized score mean and covariance, where the integrand is weighted by the term $(1 - D(\boldsymbol{x}))$ involving the optimal discriminator $D(\boldsymbol{x})$:

$$\begin{aligned} \boldsymbol{m} &= \int \boldsymbol{g}(\boldsymbol{x})(1 - D(\boldsymbol{x}))p(\boldsymbol{x})d\boldsymbol{x} \\ \boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top(1 - D(\boldsymbol{x}))p(\boldsymbol{x})d\boldsymbol{x} \end{aligned} \tag{4.5}$$

The (Fisher) score vector is the gradient (or derivative in one dimension) of the log of the data distribution with respect to its parameter $\boldsymbol{g}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$. Its actual (without the discriminator weight term) mean is null and its covariance is the Fisher Information matrix, written as $I_F = \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p(\boldsymbol{x})d\boldsymbol{x}$ for the rest of the paper.

The question of statistical efficiency of NCE to bridge the gap with MLE therefore becomes to optimize Eq. 4.4 with respect to the three hyperparameters.

Previous work Despite some early results, choosing the best noise distribution to reduce the variance of the NCE estimator remains largely unexplored. Gutmann and Hyvärinen [11] and Pihlaja et al. [63] remark that setting $p_n = p_d$ offers a MSE $(1 + \frac{1}{\nu})$ times higher than the Cramer-Rao bound. Therefore, with an infinite budget $T \to \infty$, taking all samples from noise $\nu \to \infty$ brings the $\mathrm{MSE}_{\mathrm{NCE}}$ down to the Cramer-Rao bound.

Motivated by the same goal of improving the statistical efficiency of NCE, Pihlaja et al. [63], Gutmann and Hirayama [38] and Uehara et al. [91] have looked at reducing the variance of NCE. They relax the original NCE objective by writing it as an M-divergence between the distributions p_d and p_θ [91] or as a Bregman-divergence between the density ratios $\frac{p_d}{\nu p_n}$ and $\frac{p_\theta}{\nu p_n}$. Choosing a divergence boils down to the use of specific non-linearities, which when chosen for the Jensen-Shannon f-divergence leads to the NCE estimator. Pihlaja et al. [63] numerically explore which non-linearities lead to the lowest MSE, but they explore estimators different from NCE.

More recently, Uehara et al. [91] show that the asymptotic variance of NCE can be further reduced by using the MLE estimate of the noise parameters obtained from the noise samples, as opposed to the true noise distribution. A similar idea underlies Flow-Contrastive Estimation [225]. While this is useful in practice, it does not address the question of finding the optimal noise distribution.

When the noise distribution is fixed, it remains to optimize the noise-data ratio ν and samples budget T. The effect of the samples budget on the NCE estimator is clear : it scales as $\mathrm{MSE}_{\mathrm{NCE}} \propto \frac{1}{T}$. Consequently and remarkably, the optimal noise distribution and noise-data ratio actually do not

depend on the budget T. As for the noise-data ratio ν, while Gutmann and Hyvärinen [11] and Pihlaja et al. [63] report that NCE reaches Cramer-Rao when both ν and T tend to infinity, it is of limited practical use due to finite computational resources T. In the limit of finite samples, Pihlaja et al. [63] offers numerical results touching on this matter, although it considers the noise prior is 50% which greatly simplifies the problem as the MSE here becomes linearly dependent on ν.

4.4 . Optimizing noise in NCE

In this work we aim to directly optimize the MSE of the original NCE estimator with respect to the noise distribution and noise-data ratio. Analytical optimization of the $\mathrm{MSE}_{\mathrm{NCE}}$ with respect to the noise distribution p_n or ratio ν is a difficult task : both terms appear nonlinearly within the integrands. Even in the simple case where the data follows a one-dimensional Gaussian distribution parameterized by variance, as specified in Section 2 of the Supplementary Material, the resulting expression is intractable. This motivates the need for numerical methods.

In the following, we pursue two different strategies for finding the optimal p_n. Either p_n can be chosen within the same parametric family as the data distribution (we use the same parametric model for simplicity) as in Section 4.4; this leads to a simple one-dimensional optimization problem (e.g. optimizing a Gaussian mean or variance θ). Or one can relax this assumption and use more flexible “non-parametric" methods as in Sections 4.4 and 4.5, such as a histogram-based expression for p_n. In the latter case, assuming the bins of histograms are fixed, one has in practice a higher-dimensional optimization problem with one weight per histogram bin to estimate.

Optimization within the same parametric family We use here simple data distributions to illustrate the difficulty of finding the optimal distribution. We work with families of a single scalar parameter to make sure that the numerical calculations can be performed exactly.

The data distributions considered from now on are picked among three generative models with a scalar parameter :

(a) a univariate Gaussian parameterized by its mean and whose variance is fixed to 1,

(b) a univariate zero-mean Gaussian parameterized by its variance,

(c) a two-dimensional zero-mean Gaussian parameterized by correlation, i.e. the off-diagonal entries of the covariance matrix. The variables are taken standardized.

While the Gaussian distribution is simple, it is ubiquitous in generative models literature and remains a popular choice in state-of-the-art deep learning algorithms, such as Variational Auto-Encoders (VAEs). Yet, to our knowledge, it remains completely unknown to date how to design the optimal noise to infer the parameters of a Gaussian using NCE.

Assuming the same parametric distribution for the noise as for the data, Figure 4.1 presents the optimal noise parameter as a function of the data parameter. Details on numerical methods are explained below. For the three models above and setting $\nu = 1$, one can observe that the noise parameter systematically differs from the data parameter. They are equal only in the very special case of estimating correlation (case c) for uncorrelated variables. This means that the optimal noise distribu-

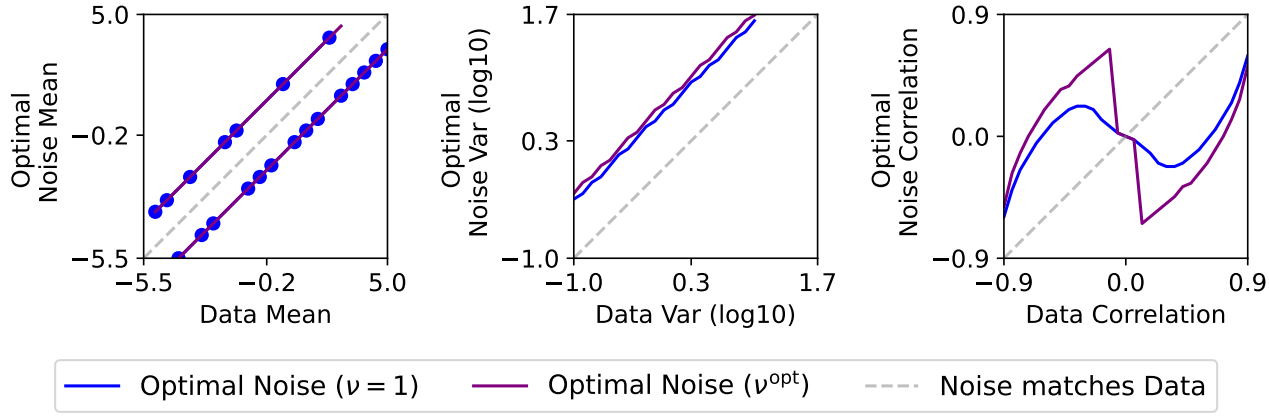


Figure 4.1 – Relationship between the (optimal) noise parameter and the data parameter. (left) Optimal variance in model (a) as function of the data mean. Note that the noise parameter has two symmetric local minima, given by the individual points, which are joined by a manually drawn line. (center) Optimal variance in model (b) as function of the data variance. (right) Optimal noise correlation in model (c) as a function of the data correlation.

tion is not equal to the data distribution, even when the noise and the data are restricted to be in the same parametric family of distributions.

Looking more closely, one can notice that the relationship between the optimal noise parameter and the data parameter highly depends on the estimation problem. For model (a), the optimal noise mean is (randomly) above or below the data mean, while at constant distance (cf. the two local minima of the MSE landscape shown in Section 1 of the Supplementary Material). For model (b), the optimal noise variance is obtained from the data variance by a scaling of 3.84. This linear relationship is coherent with the symmetry of the problem with respect to the variance parameter. Interestingly for model (c), the optimal noise parameter exhibits a nonlinear relationship to the data parameter : for a very low positive correlation between variables the noise should be negatively correlated, whereas when data variables are strongly correlated, the noise should also be positively correlated.

Having established how different the optimal parametric noise can be, a question naturally follows : what does the optimal, unconstrained noise distribution look like ?

Theory While the analytical optimization of the noise model is intractable, it is possible to study some limit cases, and by means of Taylor expansions, obtain analytical results which hopefully shed some light to the general behaviour of the estimator even far away from those limits.

In what follows, we study an analytical expression for the optimal noise distribution in three limit cases : (i) when the noise distribution is a (infinitesimal) perturbation of the data distribution $\frac{p_d}{p_n} \approx 1$; as well as when the noise proportion (ratio) is chosen so that training uses either (ii) all noise samples $\nu \to \infty$ or (iii) all data samples $\nu \to 0$. The following Theorem is proven in Section 4 of the Supplementary Material.

Theorem 1 *In either of the following two limits :*

(i) the noise distribution is a (infinitesimal) perturbation of the data distribution $\frac{p_d}{p_n}(\boldsymbol{x}) = 1 + \epsilon(\boldsymbol{x})$*;*

(ii) in the limit of all noise samples $\nu \to \infty$;

the noise distribution minimizing asymptotic MSE is

$$p_n^{\mathrm{opt}}(\boldsymbol{x}) \propto p_d(\boldsymbol{x}) \|\boldsymbol{g}(\boldsymbol{x})\|_{\boldsymbol{I}_F^{-2}} \ . \tag{4.6}$$

Interestingly, this is the same as the optimal noise derived by Pihlaja et al. [63] for another, related estimator (Monte Carlo MLE with Importance Sampling). For example, in the case of estimating Gaussian variance : $p_n^{\mathrm{opt}}(x) \propto \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{x^2}{2\theta}} |x^2 - \theta|$ which is highly *non-Gaussian unlike the data distribution*. Similar derivations can be easily done for the cases of Gaussian mean or correlation.

In Section 4 of the Supplementary Material, we further derive a general formula for the gap between the MSE for the typical case $p_n = p_d$ and the optimal case $p_n = p_n^{\mathrm{opt}}$. It is given by

$$\Delta\mathrm{MSE} = \frac{1}{T} \mathrm{Var}_{x \sim p_d}(\|\boldsymbol{g}(\boldsymbol{x})\|_{\boldsymbol{I}_F^{-2}}) \ . \tag{4.7}$$

This quantity seems to be positive for any reasonable distribution, which implies (in the all-noise limit) that the optimal noise cannot be the data distribution p_d. Furthermore, we can compute the gap to efficiency in the all noise limit, i.e. between $p_n = p_n^{\mathrm{opt}}$ and the Cramer-Rao lower bound $\Delta_{\mathrm{opt}}\mathrm{MSE} = \frac{1}{T}\mathbb{E}_{x \sim p_d}(\|\boldsymbol{g}(\boldsymbol{x})\|_{\boldsymbol{I}_F^{-2}})^2$.

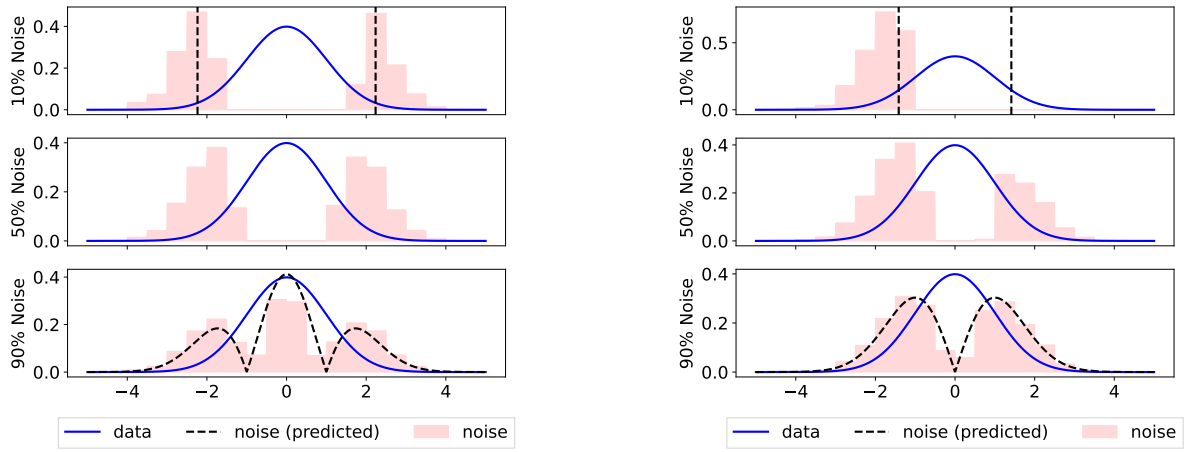
In the third case, the limit of all data, we have the following conjecture :

Conjecture 1 *In case (iii), the limit of all data samples $\nu \to 0$, the optimal noise distribution is such that it is all concentrated at the set of those $\boldsymbol{\xi}$ which are given by*

$$\begin{aligned} &\arg\max_{\boldsymbol{\xi}} p_d(\boldsymbol{\xi}) \mathrm{tr}\Big((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-1} \Big)^{-1} \\ &\text{s.t.} \quad \boldsymbol{g}(\xi) = \text{constant} \ . \end{aligned} \tag{4.8}$$

This is typically a degenerate distribution since it is concentrated on a low-dimensional manifold, in the sense of a Dirac delta. For a scalar parameter, the function whose maxima are sought is simply $p_d(\boldsymbol{\xi})\|\boldsymbol{g}(\xi)\|^2$. An informal proof of this conjecture is given in Section 4 of the Supplementary Material. The "proof" is not quite rigorous due to the singularity of the optimal "density", which is why we label this as conjecture only. Indeed, this closed-form formula (Eq. 4.8) was obtained using a Taylor expansion up to the first order. This formula is well-defined in one dimension but is challenging in higher dimensions as it involves the inversion of a rank-one matrix, which we accomplish by regularization (provided at the end of Section 4 of the Supplementary Material). While this is in apparent contradiction to having the noise distribution's support include the data distribution's, this result can be understood as a first-order approximation of what one should do with few noise data points available.

Specifically, in the case of estimating a Gaussian mean (for unit variance), the maximization in the first line of Eq. 4.8 yields two candidates for $p_n^{\mathrm{opt}}(x)$ to concentrate its mass on : $\delta_{-\sqrt{2}}$ and $\delta_{\sqrt{2}}$.



(a) Optimal noise for model (b) (Gaussian variance).

(b) Optimal noise for model (a) (Gaussian mean).

Figure 4.2 – Histogram-based optimal noise distributions. Each row gives a different ν or noise proportion. The pink bars give the numerical approximations. The theoretical approximation of optimal noise is given by the dashed lines : the all-noise limit in the bottom panel, and the all-data limit in the top panel. In the top panel, the optimal noise is given by single points (Dirac masses) which are chosen symmetric for the purposes of illustration, but as explained in the text, they are two global minima in the case of Gaussian mean estimation, whereas when estimating the variance, any distribution of probability on those two points is equally optimal.

Moreover, the second line of Eq. 4.8 predicts how the probability mass should be distributed to the two candidates : because they have different scores $g(-\sqrt{2}) \neq g(\sqrt{2})$, they are two distinct global minima. This is coherent with the two minima observed for the Gaussian mean in Figure 4.1 (top-left). Similarly, when estimating a Gaussian variance, the maximization in the first line of Eq. 4.8 yields candidates $\delta_{-\sqrt{5}}$ and $\delta_{\sqrt{5}}$ for $p_n^{\mathrm{opt}}(x)$. In this case however, both candidates have the same score $g(-\sqrt{5}) = g(\sqrt{5})$. The theory above does not say anything about how the probability mass should be distributed to these two points : it can be 50-50 or all on just one point. A possible solution is $p_n^{opt}(x) = \frac{1}{2}(\delta_{-\sqrt{5}} + \delta_{\sqrt{5}})$ as observed in Figure 4.2a. Throughout, the optimal noise distributions are highly *non-Gaussian unlike the data distribution*.

So far, we have obtained the optimal noise which minimizes the (asymptotic) estimation error $\mathbb{E}\big[\|\hat{\theta}_T - \theta^*\|^2\big] = \frac{1}{T_d}\mathrm{tr}(\Sigma)$ of NCE for the data *parameter*. However, sometimes estimating the parameter is only a means for estimating the data *distribution* — not an end in itself. We therefore consider the (asymptotic) estimation error induced by the NCE estimator $\hat{\theta}_T$ in the distribution space using the Kullback-Leibler divergence which is well-known to equal

$$\mathbb{E}\big[\mathcal{D}_{\mathrm{KL}}(p_d, p_{\hat{\theta}_T})\big] = \frac{1}{2T_d}\mathrm{tr}(\Sigma I_F) \tag{4.9}$$

(shown in Section 5 of the Supplementary Material). We are thus able to obtain the optimal noise for estimating the data *distribution* in cases (i), (ii) and (iii).

Theorem 2 *In the two limit cases of Theorem 1, the noise distribution minimizing the expected Kullback-Leibler divergence is given by*

$$p_n^{\mathrm{opt}}(\boldsymbol{x}) \propto p_d(\boldsymbol{x}) \|\boldsymbol{g}(\boldsymbol{x})\|_{\boldsymbol{I}_F^{-1}} \ . \tag{4.10}$$

In the third case, the limit of all data, we have the following conjecture :

Conjecture 2 *In the limit of Conjecture 1 the noise distribution minimizing the expected Kullback-Leibler divergence is such that it is all concentrated at the set of those $\boldsymbol{\xi}$ which are given by*

$$\begin{aligned} &\arg\max_{\boldsymbol{\xi}} p_d(\boldsymbol{\xi}) \mathrm{tr}\left((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-\frac{1}{2}} \right)^{-1} \\ &\text{s.t.} \quad \boldsymbol{g}(\xi) = \text{constant} \ . \end{aligned} \tag{4.11}$$

These optimal noise distributions resemble those from Theorem 1 and Conjecture 1 : only the exponent on the Fisher Information matrix changes. This is predictable, as the new cost function $\frac{1}{2T_d}\mathrm{tr}(\Sigma I_F)$ is obtained by scaling with the Fisher Information matrix. More specifically, when the data parameter is scalar, the optimal noises from Theorems 1 and 2 coincide, as the Fisher Information becomes a multiplicative constant; those from Conjectures 1 and 2 do not coincide but are rather similar. The scope of this paper is to investigate the already rich case of a one-dimensional parameter, hence the following focuses on the optimal noise distributions from Theorem 1 and Conjecture 1.

4.5 . Experiments

We now turn to experiments to validate the theory above. Specifically, we verify our formulae for the optimal noise distribution in the all-data (Eq.4.8) and all-noise (Eq.4.6) limits, by numerically minimizing the MSE (Eq.4.4). Outside these limits, we show that our formulae are competitive against a parametric approach, and that the general-case optimal noise is an interpolation between both limits. We first describe numerical strategies.

Numerical Methods The integrals from Eq. 4.5 involved in evaluating the asymptotic MSE can be approximated using numerical integration (quadrature) or Monte-Carlo simulations. While both approaches lead to comparable results, quadrature is significantly faster and more precise, especially in low dimension. However, using Monte-Carlo leads to an estimate that is fully differentiable with respect to the parameters of p_n.

To tackle the one-dimensional parametric problem, we simply employed quadrature for evaluating the function to optimize over a dense grid and then selected the minimum. This appeared as the most computationally efficient strategy and allows for visualizing the MSE landscape reported in Section 1 of the Supplementary Material. In the multi-dimensional non-parametric case, the histogram's weights can be optimized by first-order methods using automatic differentiation.

In the following experiments, the optimization strategy consists in obtaining the gradients of the Monte-Carlo estimate using PyTorch [155] and plugging them into a non-linear conjugate gradient scheme implemented in Scipy [229]. We chose the conjugate-gradient algorithm as it is deterministic (no residual asymptotic error as with SGD), and as it offered fast convergence. None of the experiments below required more than 100 iterations of conjugate-gradient. Note that for numerical precision, we had to set PyTorch's default to 64-bit floating-point precision. Our code is available at https://github.com/l-omar-chehab/nce-noise-variance.

Results Figure 4.2a shows the optimal histogram-based noise distribution for estimating the variance of a zero-mean Gaussian, together with our theoretical predictions (Theorem 1 and Conjecture 1). We can see that our theoretical predictions in the all-data and all-noise limits match numerical results. It is apparent in Figure 4.2a that the optimal noise places its mass where the data distribution is high, and where it varies most when θ^* changes. Furthermore, the noise distribution in the all-data limit has higher mass concentration, which also matches our predictions. Interestingly, in a case not covered by our hypotheses, when there are as many noise samples as data samples i.e. noise proportion of 50% or $\nu = 1$, the optimal noise in Figure 4.2a (middle) is qualitatively not very different from the limit cases of all data or all noise samples.

Figure 4.2b gives the same results for the estimation of a Gaussian's mean. The conclusions are similar; in this case, the optimal distributions in the two limits resemble each other even more. It is here important to take into account the indeterminacy of distributing probability mass on the two Diracs, which is coherent with initial experiments in Figure 4.1 as well as the MSE landscape included in Section 1 of the Supplementary Material. Figure 4.2b is a perfect illustration of a complex phenomenon occurring in a setup as simple as Gaussian mean estimation. Our conjecture in Eq. 4.8 predicts the equivalent optimal noises seen in our experiments, in Figure 1 (top-left) and Figure 2.b., where the noise concentrates its mass on either point of the set $\{-\sqrt{2}, \sqrt{2}\}$. Indeed, Eq. 4.8 shows that any noise which concentrates its mass on a set of points where the score is constant is (equally) optimal. So despite its approximative quality, Eq. 4.8 is able to explain what we observed empirically : in the all-data limit, there can be many equivalent optimal noises.

Figure 4.3 shows the numerically estimated optimal noise distribution for model (c) using a Gaussian correlation parameter. Here, the distributions are perhaps even more surprising than in previous figures. This can be partly understood by the extremely nonlinear dependence of the optimal noise parameter from the data parameter shown in Fig. 4.1.

We next ask : how robust to ν is the analytical noise we derived in these limiting cases? Figure 4.4 shows the Asymptotic MSE achieved by two noise models, across a range of noise proportions. The first noise model is the optimal noise in the parametric family containing the data distribution $p_n = p_\theta$, optimized for $\nu = 1$, while the second noise model is the optimal analytical noise p_n^{opt} derived in the all-noise limit (Eq.4.6). They are both compared to the Cramer-Rao lower bound. For all models (a) (b) and (c), the optimal analytical noise p_n^{opt} (red curve) is empirically useful even far away from the all-noise limit, and across the entire range of noise proportions. In fact, $p_n = p_n^{\text{opt}}$ empirically seems a better choice than using the data distribution $p_n = p_d$, and is (quasi) uniformly equal to or better than a parametric noise $p_n = p_\theta$ optimized for $\nu = 1$.

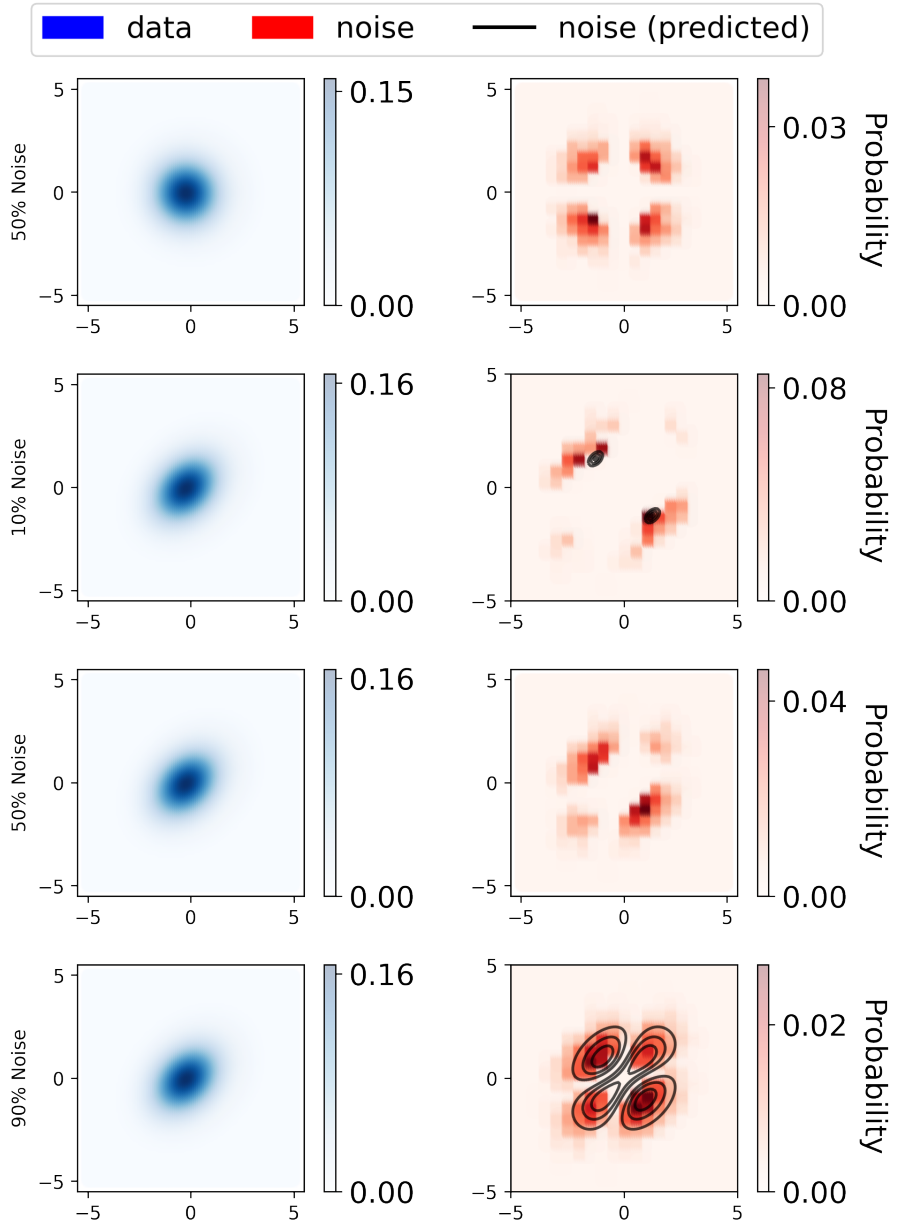


Figure 4.3 – Optimal noise for a 2D Gaussian parameterized by correlation. 2D Gaussian with correlation 0 (top) and 0.3 (bottom three) are considered. Left panel is data density, right panel is the optimal histogram-based noise density. The theoretical approximation of optimal noise is given by the black level lines : the case of Theorem 1 the bottom panel, and the Conjecture 1 in the second panel. Here, the optimal noise in the latter limit is given by a softmax relaxation with temperature 0.01. It makes the choice of placing its mass symmetrically on the single points (Dirac masses), but as explained in the text, any distribution of probability on those two points could be equally optimal.

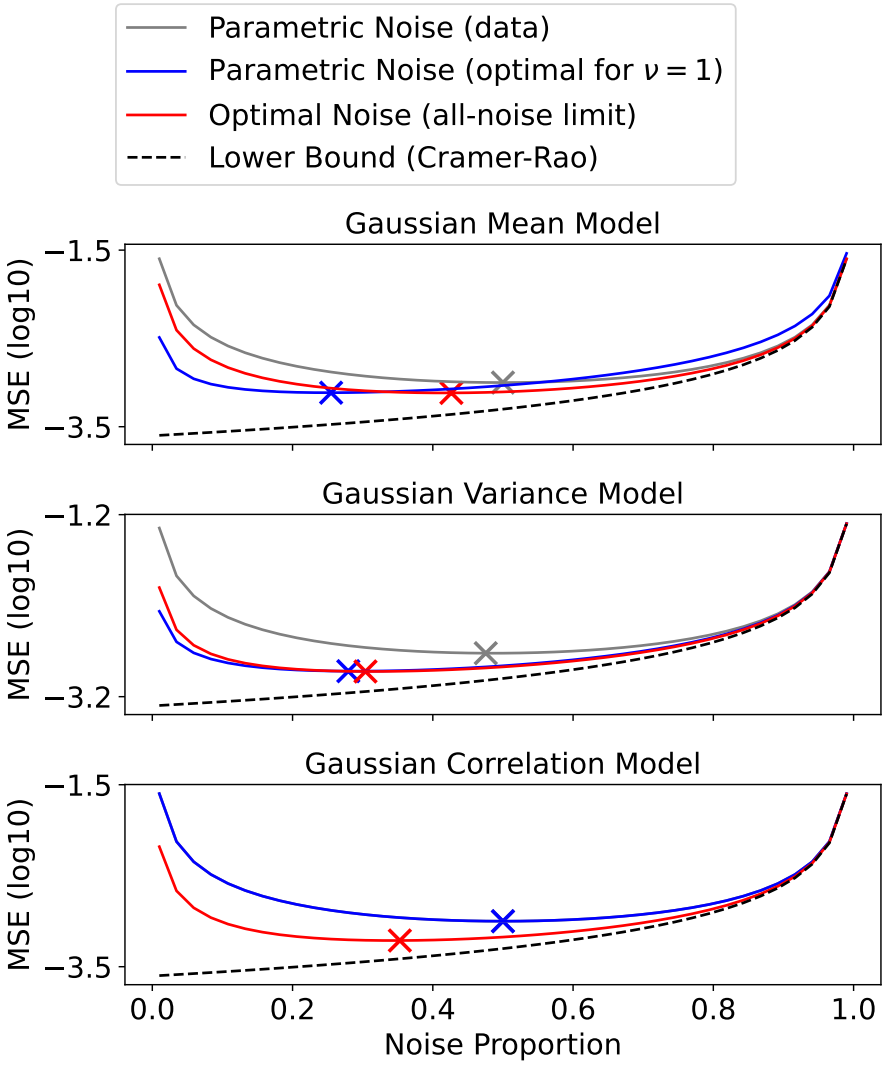


Figure 4.4 – Asymptotic MSE vs. noise proportion. Top panel : Asymptotic MSE vs. noise proportion for model (a) with parameter mean; Middle panel : Asymptotic MSE vs. noise proportion for model (b) with parameter variance; Bottom panel : Asymptotic MSE vs. noise proportion for model (c) with parameter correlation. The parameter in “parametric noise" is the optimal parameter for $\nu = 1$, i.e. for when half the samples are noise and half are data. The "optimal noise" is the approximation given by Theorem 1.

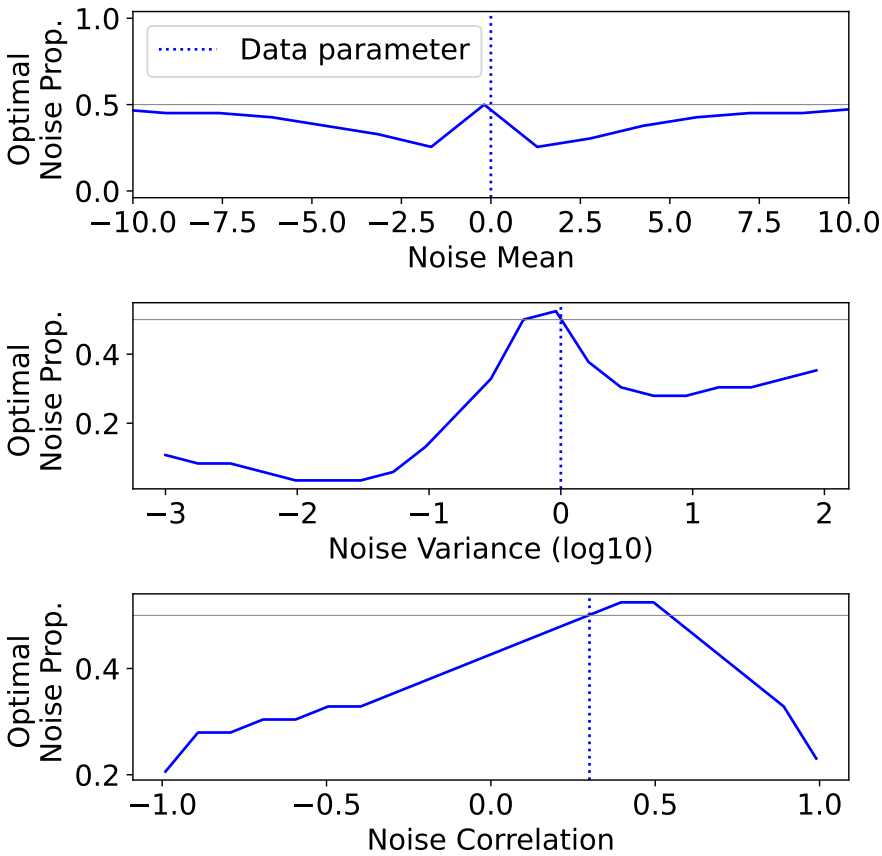


Figure 4.5 – Optimal noise proportion against the noise parameter. Top panel for model (a), Gaussian mean; Middle panel for model (b), Gaussian variance; Bottom panel for model (c), Gaussian correlation.

Optimizing Noise Proportion Next, we consider optimization of the noise proportion. It is often heuristically assumed that having 50% noise, i.e. $\nu = 1$ is optimal. On the other hand, Pihlaja et al. [63] provided a general analysis, although it didn't quite answer this question from a practical viewpoint and uses a slightly different NCE estimator. To see this, compare the NCE estimator defined in Pihlaja et al. [63, Eq. 15] with the estimator we use from Gutmann and Hyvärinen [11, Eq. 10]. Note that $\nu = T_n/T_d$ represents only the ratio of samples used to approximate the expectations in the objective of Pihlaja et al. [63, Eq. 15]. In particular, it appears linearly in the objective minimized by the NCE estimator. In contrast, in Gutmann and Hyvärinen [11, Eq. 10] this quantity represents also the ratio of prior distributions on the noise and data $\nu = P(Y = 0)/P(Y = 1)$. In particular, it appears non-linearly in the objective minimized by the NCE objective.

In the special case where $p_d = p_n$, we can actually show (see Section 3 of the Supplementary Material) that the optimal noise proportion is 50%. This is obtained for a fixed computational budget T, as the noise proportion varies between 0 and 1. When this constraint on the budget is relaxed, the optimal noise proportion is $\nu \to \infty$ as in Corollary 7 and Figure 4.d. of [11]. The reciprocal for the theoretical result above does not hold : a noise proportion of 50% does *not* ensure that the noise distribution equals the data's, as shown by counter-examples in Figures 4.1 and 4.5.

However, in the general case $p_n \neq p_d$, the optimal proportion is not 50%. We can again look at Figure 4.4 which analyses the MSE as a function of noise proportion for simple one-parameter families. It is not optimized, in general, at 50%, for the noise distributions considered here. In fact, the parameter of the noise distribution is here optimized for a proportion of 50%, so the results are skewed towards finding that proportion optimal, but still that is not the optimum for most cases.

A closer look at this phenomenon is given by Figure 4.5 which shows the optimal noise proportion as a function of a Gaussian's parameter (mean, variance, or correlation). We see that while it is 50% for when the data parameter is used for noise, it is in general less.

4.6 . Discussion

We have shown that choosing an optimal noise means choosing a noise distribution that is *different* to the data's. An interesting question is what implications does this have for GANs, which iteratively guide the noise distribution to *match* the data's? Both NCE and GANs in fact solve the binary task of discriminating data from noise. While the optimal discriminator for the binary task recovers the density ratio between data and noise, GANs parameterize the entire ratio (as well as the noise distribution), while NCE only parameterizes the ratio numerator. Hence they do not learn the same object, though GANs do claim inspiration from NCE [223]. Moreover, of course, the goals of the two methods are completely different : GANs do not perform estimation of parameters of a statistical model but focus on the generation of data.

Nevertheless, GAN updates *have* inspired the choice of NCE noise as in Flow-Contrastive Estimation (FCE) by Gao et al. [225], which parameterizes both the discriminator numerator and discriminator, providing a bridge between NCE and GANs. Results on FCE by Gao et al. [225] empirically demonstrate that the choice of noise matters : NCE is made quicker by iterative noise updates *à la* GAN, presumably because setting the noise distribution equal to the data's reduces asymptotic variance compared to choosing a generic noise distribution such as the best-matching Gaussian. Noise-updates

based on the optimal noise in this paper, could perhaps accelerate convergence even further, avoiding the numerical difficulties of an adversarial game while still increasing the statistical efficiency.

However, using the optimal noise distributions we present in Section 4.4 can be numerically challenging, especially when the parametric model $p_{\boldsymbol{\theta}}$ is higher-dimensional and unnormalized (e.g. θ is a dense covariance matrix along with the normalization term as a parameter). Evaluating an optimal noise involves the Fisher score (and therefore access to the very data distribution we seek to estimate) and a Monte-Carlo method may be needed for sampling. We hope that these questions can be resolved in practice by having a relatively simple noise model which is still more statistically efficient than alternatives typically used with NCE, and whose choice is guided by our optimality results.

Conclusion We studied the choice of optimal design parameters in Noise-Contrastive Estimation. These are essentially the noise distribution and the proportion of noise. We assume that the total number of data points (real data + noise) is fixed due to computational resources, and try to optimize those two hyperparameters. It is easy to show empirically that, in stark contrast to what is often assumed, the optimal noise distribution is not the same as the data distribution, thus extending the analysis by Pihlaja et al. [63]. Our main theoretical results derive the optimal noise distribution in limit cases where either almost all samples to be classified are noise, or almost all samples are real data, or the noise distribution is an (infinitesimal) perturbation of the data distribution. The optimal noise distributions in two of these cases are different but have in common the point of emphasizing parts of the data space where the Fisher score function changes rapidly. We hope these results will improve the performance of NCE in demanding applications.

Acknowledgements Numerical experiments were made possible thanks to the scientific Python ecosystem : Matplotlib [230], Scikit-learn [165], Numpy [231], Scipy [229] and PyTorch [155].

We would like to thank our reviewers whose detailed comments have helped improve this paper.

This work was supported by the French ANR-20-CHIA-0016 to Alexandre Gramfort. Aapo Hyvärinen was supported by funding from the Academy of Finland and a Fellowship from CIFAR.

4.7 . Supplemental Material

4.7.1 . Visualizations of the MSE landscape

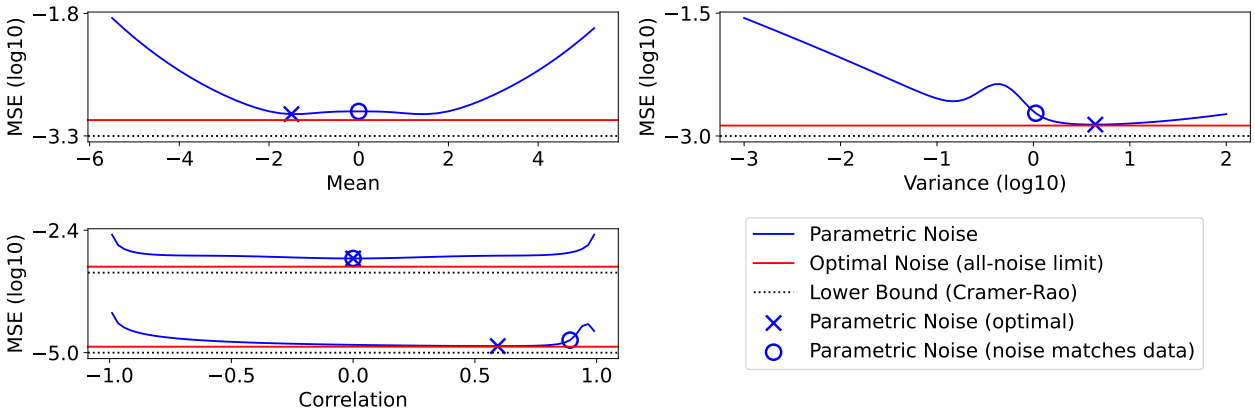


Figure 4.6 – MSE vs. the noise parameter. Top left panel for model (i), Gaussian mean ; Top right panel for model (ii), Gaussian variance ; Bottom left for model (iii), Gaussian correlation.

We provide visualizations of the MSE landscape of the NCE estimator, when the noise is constrained within a parametric family containing the data.

We draw attention to the two local minima symmetrically placed to the left and to the right of the Gaussian mean. This corroborates the indeterminacies observed in this paper (Conjecture on limit of zero noise), as to where the optimal noise should place its mass for this estimation problem.

4.7.2 . Intractability of the 1D Gaussian case

Suppose the data distribution p_d is a one-dimensional standardized zero-mean Gaussian. The model and noise distributions are of the same family, parameterized by mean and/or variance (we write these together in one model) :

$$p_\theta(x) = \frac{1}{\sqrt{2\pi\alpha}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\alpha}}, \quad p_n(x) = \frac{1}{\sqrt{2\pi\beta}} e^{-\frac{1}{2}(\frac{(x-\pi)^2}{\beta}} \qquad x \in \mathbb{R}$$

We can write out the relevant functions, evaluated at $\alpha = 1, \mu = 0$ as the 2D score :

$$\boldsymbol{g}(x) = \begin{pmatrix} \partial_\mu \log p_\theta \\ \partial_\alpha \log p_\theta \end{pmatrix} \Bigg|_{\mu=0,\alpha=1} = \begin{pmatrix} x \\ -1 + x^2 \end{pmatrix}$$

and its “pointwise covariance" : $\boldsymbol{g}(x)\boldsymbol{g}(x)^\top = \begin{pmatrix} x^2 & -x + x^3 \\ -x + x^3 & x^4 - x^2 + 1 \end{pmatrix}$

In the following, we consider estimation of variance only. i.e. only the second term in $\boldsymbol{m}$ and the second diagonal term in the Fisher information matrix I . Now we can compute the generalized score mean m and mean of square I as they intervene in the MSE formula for Noise-Contrastive Estimation :

$$
\begin{aligned}
m &= \int g(x)(1-D(x))p(x)dx \\
&= -\frac{1}{2\sqrt{2\pi}} \int \left(e^{\frac{-x^2}{2}} \frac{1}{1+\frac{1}{\nu}\sqrt{\beta}e^{\frac{-x^2}{2}(1-\frac{1}{\beta})}} \right) dx + \\
&\frac{1}{2\sqrt{2\pi}} \int x^2 \left(e^{\frac{-x^2}{2}} \frac{1}{1+\frac{1}{\nu}\sqrt{\beta}e^{\frac{-x^2}{2}(1-\frac{1}{\beta})}} \right) dx
\end{aligned}
$$

and

$$
\begin{aligned}
I &= \int g(x)^2(1-D(x))p(x)dx \\
&= \frac{1}{4\sqrt{2\pi}} \int x^4 \left(e^{\frac{-x^4}{2}} \frac{1}{1+\frac{1}{\nu}\sqrt{\beta}e^{\frac{-x^2}{2}(1-\frac{1}{\beta})}} \right) dx \\
&- \frac{1}{2\alpha^3\sqrt{2\pi}} \int x^2 \left(e^{\frac{-x^2}{2}} \frac{1}{1+\frac{1}{\nu}\sqrt{\beta}e^{\frac{-x^2}{2}(1-\frac{1}{\beta})}} \right) dx \\
&+ \frac{1}{4\sqrt{2\pi}} \int \left(e^{\frac{-x^2}{2}} \frac{1}{1+\frac{1}{\nu}\sqrt{\beta}e^{\frac{-x^2}{2}(1-\frac{1}{\beta})}} \right) dx
\end{aligned}
$$

We see that even in a simple 1D Gaussian setting, evaluating the asymptotic MSE of the Noise-Contrastive Estimator is untractable in closed-form, given the integrals in I, where the integrand includes the product of a Gaussian density with the logistic function compounded by the Gaussian density, further multiplied by monomials. While here we considered the case of variance, the intractability is seen even in the case of the mean. Optimizing the asymptotic MSE with respect to β and π (noise distribution) or ν (identifiable to the noise proportion) yields similarly intractable integrals.

4.7.3 . Optimal Noise Proportion when the Noise Distribution matches the Data Distribution : Proof

We wish to minimize the MSE given by

$$
\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) = \frac{\nu+1}{T}\mathrm{tr}(\boldsymbol{I}^{-1} - \frac{\nu+1}{\nu}(\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1}))
$$

when $p_n = p_d$. In that case,

$$
D(\boldsymbol{x}) = \frac{p_d}{p_d + \nu p_n}(\boldsymbol{x}) = \frac{p_d}{p_d + \nu p_d}(\boldsymbol{x}) = \frac{1}{1+\nu}
$$

and the integrals involved become

$$
\boldsymbol{m} = \int \boldsymbol{g}(\boldsymbol{x})(1-D(\boldsymbol{x}))p(\boldsymbol{x})d\boldsymbol{x}
$$

$$
\begin{aligned}
&= \frac{\nu}{1+\nu} \int \boldsymbol{g}(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \\
&= 0
\end{aligned}
$$

given the score has zero mean, and

$$
\begin{aligned}
\boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top (1 - D(\boldsymbol{x})) p(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{\nu}{1+\nu} \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{\nu}{1+\nu} \boldsymbol{I}_F \ .
\end{aligned}
$$

The objective function thus reduces to

$$
\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) = \frac{\nu+1}{T} \mathrm{tr}(\boldsymbol{I}^{-1}) = \frac{(\nu+1)^2}{\nu T} \mathrm{tr}(\boldsymbol{I}_F^{-1}) \propto \frac{(\nu+1)^2}{\nu} \ .
$$

The derivative with respect to ν is proportional to $\frac{1}{\nu^2} - 1$ and is null when $\nu = 1$ so when the noise proportion is 50%.

Note that in that case where $p_n = p_d$, we can compare the MSE achieved by NCE (using T_d data samples and T_n noise samples) with the MSE achieved my MLE (using T_d data samples) :

$$
\frac{\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n)}{\mathrm{MSE}_{\mathrm{MLE}}(T_d)} = \frac{\frac{(\nu+1)^2}{\nu T} \mathrm{tr}(\boldsymbol{I}_F^{-1})}{\frac{1}{T_d} \mathrm{tr}(\boldsymbol{I}_F^{-1})} = \frac{\frac{(\nu+1)^2}{\nu T} \mathrm{tr}(\boldsymbol{I}_F^{-1})}{\frac{\nu+1}{T} \mathrm{tr}(\boldsymbol{I}_F^{-1})} = 1 + \frac{1}{\nu}
$$

which is known from [11, 63].

4.7.4 . Optimal Noise for Estimating a Parameter : Proofs

We here prove the theorem and conjecture for the optimal noise distribution in three limit cases $\nu \to 0$ (all data samples), $\nu \to \infty$ (all noise samples), and $\frac{p_d}{p_n}(.) = 1 + \epsilon(.)$ as $\epsilon(.) \to 0$ (noise distribution is an infinitesimal perturbation of the data distribution).

The goal is to optimize the $\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n)$ with respect to the noise distribution p_n, where

$$
\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) = \frac{\nu+1}{T} \mathrm{tr}(\boldsymbol{I}^{-1} - \frac{\nu+1}{\nu} (\boldsymbol{I}^{-1} \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{I}^{-1})) \tag{4.12}
$$

where the integrals

$$
\begin{aligned}
\boldsymbol{m} &= \int \boldsymbol{g}(\boldsymbol{x}) (1 - D(\boldsymbol{x})) p(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top (1 - D(\boldsymbol{x})) p(\boldsymbol{x}) d\boldsymbol{x}
\end{aligned}
$$

depend non-linearly on p_n via the optimal discriminator :

$$
1 - D(\boldsymbol{x}) = \frac{\nu p_n(\boldsymbol{x})}{p_d(\boldsymbol{x}) + \nu p_n(\boldsymbol{x})}
$$

The general proof structure is :

- Perform a Taylor expansion of $1 - D(\boldsymbol{x})$ in the $\nu \to 0$ or $\nu \to \infty$ limit
- Plug into the integrals $\boldsymbol{m}$, $\boldsymbol{I}$ and evaluate them (up to a certain order)
- Perform a Taylor expansion of $\boldsymbol{I}^{-1}$ (up to a certain order)
- Evaluate the $\mathrm{MSE}_{\mathrm{NCE}}$ (up to a certain order)

- Optimize the $\mathrm{MSE}_{\mathrm{NCE}}$ w.r.t. p_n
- Compute the MSE gaps at optimality

Theorem 1 *In either of the following two limits :*

(i) the noise distribution is a (infinitesimal) perturbation of the data distribution $\frac{p_d}{p_n} = 1 + \epsilon(x)$;

(ii) in the limit of all noise samples $\nu \to \infty$;

the noise distribution minimizing asymptotic MSE is

$$p_n^{\mathrm{opt}}(\boldsymbol{x}) \propto p_d(\boldsymbol{x}) \|\boldsymbol{I}_F^{-1} \boldsymbol{g}(\boldsymbol{x})\| \ . \tag{4.13}$$

Proof : case where $\nu \to \infty$.
We start with a change of variables $\gamma = \frac{1}{\nu} \to 0$ to bring us to a zero-limit.
The MSE in terms of our new variable $\gamma = \frac{1}{\nu}$ can be written as :

$$\begin{aligned}
&\mathrm{MSE}_{\mathrm{NCE}}(T, \gamma, p_n) \\
&= \frac{\gamma + 1}{\gamma T} \mathrm{tr}(\boldsymbol{I}^{-1}) - \frac{(\gamma+1)^2}{T\gamma} \mathrm{tr}(\boldsymbol{I}^{-1} \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{I}^{-1}) \\
&= \Big(\gamma^{-1} T^{-1} + \gamma^0 T^{-1}\Big) \mathrm{tr}(\boldsymbol{I}^{-1}) - \Big(\gamma^{-1} T^{-1} + \gamma^0 2T^{-1} + \gamma^1 T^{-1}\Big) \mathrm{tr}(\boldsymbol{I}^{-1} \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{I}^{-1})
\end{aligned}$$

Given the term up until γ^{-1} in the MSE, we will use Taylor expansions up to order 2 throughout the proof, in anticipation that the MSE will be expanded until order 1.

- Taylor expansion of the discriminator

$$1 - D(\boldsymbol{x}) = \frac{\nu p_n(\boldsymbol{x})}{p_d(\boldsymbol{x}) + \nu p_n(\boldsymbol{x})} = \frac{1}{1 + \gamma \frac{p_d}{p_n}(\boldsymbol{x})} = 1 - \gamma \frac{p_d}{p_n}(\boldsymbol{x}) + \gamma^2 \frac{p_d^2}{p_n^2}(\boldsymbol{x}) + \circ(\gamma^2)$$

- Evaluating the integrals $\boldsymbol{m}$, $\boldsymbol{I}$

$$\begin{aligned}
\boldsymbol{m} &= \int \boldsymbol{g}(\boldsymbol{x}) p_d(\boldsymbol{x}) \Big(1 - D(\boldsymbol{x})\Big) d\boldsymbol{x} \\
&= \int \boldsymbol{g}(\boldsymbol{x}) p_d(\boldsymbol{x}) \Big(1 - \gamma \frac{p_d}{p_n}(\boldsymbol{x}) + \gamma^2 \frac{p_d^2}{p_n^2}(\boldsymbol{x}) + \circ(\gamma^2)\Big) d\boldsymbol{x} \\
&= \boldsymbol{m}_F - \gamma \boldsymbol{a} + \gamma^2 \boldsymbol{b} + \circ(\gamma^2)
\end{aligned} \tag{4.14}$$

where $\boldsymbol{m}_F$ is the Fisher-score mean of the (possibly unnormalized) model and we use shorthand notations a and b for the remaining integrals :

$$\begin{aligned}
\boldsymbol{m}_F &= \int \boldsymbol{g}(\boldsymbol{x}) p_d(\boldsymbol{x}) d\boldsymbol{x} = 0 \\
\boldsymbol{a} &= \int \boldsymbol{g}(\boldsymbol{x}) \frac{p_d^2}{p_n}(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{b} &= \int \boldsymbol{g}(\boldsymbol{x}) \frac{p_d^3}{p_n^2}(\boldsymbol{x}) d\boldsymbol{x} \ .
\end{aligned}$$

Similarly,

$$\begin{aligned}
\boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x}) \Big(1 - D(\boldsymbol{x})\Big) d\boldsymbol{x} \\
&= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x}) \Big(1 - \gamma \frac{p_d}{p_n}(\boldsymbol{x}) + \gamma^2 \frac{p_d^2}{p_n^2}(\boldsymbol{x}) + \circ(\gamma^2)\Big) d\boldsymbol{x} \\
&= \boldsymbol{I}_F - \gamma \boldsymbol{A} + \gamma^2 \boldsymbol{B} + \circ(\gamma^2)
\end{aligned}$$

where the Fisher-score covariance (Fisher information) is $\boldsymbol{I}_F$ and we use shorthand notations A and B for the remaining integrals :

$$\begin{aligned}
\boldsymbol{I}_F &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{A} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top \frac{p_d^2}{p_n}(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{B} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top \frac{p_d^3}{p_n^2}(\boldsymbol{x}) d\boldsymbol{x} \ .
\end{aligned}$$

- Taylor expansion of $\boldsymbol{I}^{-1}$

$$\begin{aligned}
\boldsymbol{I}^{-1} &= \Big(\boldsymbol{I}_F - \gamma \boldsymbol{A} + \gamma^2 \boldsymbol{B} + \circ(\gamma^2)\Big)^{-1} \\
&= \Big(\boldsymbol{I}_F(\mathbf{Id} - \gamma \boldsymbol{I}_F^{-1} \boldsymbol{A} + \gamma^2 \boldsymbol{I}_F^{-1} \boldsymbol{B}) + \circ(\gamma^2)\Big)^{-1} \\
&= \boldsymbol{I}_F^{-1} \Big(\mathbf{Id} - \gamma \boldsymbol{I}_F^{-1} \boldsymbol{A} + \gamma^2 \boldsymbol{I}_F^{-1} \boldsymbol{B}\Big)^{-1} + \circ(\gamma^2) \\
&= \boldsymbol{I}_F^{-1} \Big(\mathbf{Id} + \gamma \boldsymbol{I}_F^{-1} \boldsymbol{A} + \gamma^2((\boldsymbol{I}_F^{-1} \boldsymbol{A})^2 - \boldsymbol{I}_F^{-1} \boldsymbol{B}) + \circ(\gamma^2)\Big) + \circ(\gamma^2) \\
&= \boldsymbol{I}_F^{-1} + \gamma \boldsymbol{I}_F^{-2} \boldsymbol{A} + \gamma^2 (\boldsymbol{I}_F^{-1} (\boldsymbol{I}_F^{-1} \boldsymbol{A})^2 - \boldsymbol{I}_F^{-2} \boldsymbol{B}) + \circ(\gamma^2)
\end{aligned} \tag{4.15}$$

- Evaluating the $\mathrm{MSE}_{\mathrm{NCE}}$

$$\boldsymbol{I}^{-1} \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{I}^{-1}$$

$$= \boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1} + \gamma^2(\boldsymbol{I}_F^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{I}_F^{-1} + \boldsymbol{I}_F^{-2}\boldsymbol{A}\boldsymbol{m}_F\boldsymbol{m}_F\top\boldsymbol{I}_F^{-2}\boldsymbol{A}) + \circ(\gamma^2)$$

by plugging in the Taylor expansions of $\boldsymbol{I}^{-1}$ and $\boldsymbol{m}$ and retaining only terms up to the second order. Hence, the second term of the MSE without the trace is

$$\begin{aligned}
&\Big(\gamma^{-1}T^{-1} + \gamma^0 2T^{-1} + \gamma^1 T^{-1}\Big)\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1} \\
&= \gamma^{-1}\frac{1}{T}(\boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1}) + \gamma^0\frac{2}{T}(\boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1}) + \\
&\quad \gamma^1\frac{1}{T}(\boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1} + \boldsymbol{I}_F^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{I}_F^{-1} + \boldsymbol{I}_F^{-2}\boldsymbol{A}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-2}\boldsymbol{A}) + \circ(\gamma)
\end{aligned}$$

and the first term of the MSE without the trace is

$$\begin{aligned}
&\Big(\gamma^{-1}T^{-1} + \gamma^0 T^{-1}\Big)(\boldsymbol{I}^{-1}) \\
&= \Big(\gamma^{-1}T^{-1} + \gamma^0 T^{-1}\Big)\Big(\boldsymbol{I}_F^{-1} + \gamma\boldsymbol{I}_F^{-2}\boldsymbol{A} + \gamma^2(\boldsymbol{I}_F^{-1}(\boldsymbol{I}_F^{-1}\boldsymbol{A})^2 - \boldsymbol{I}_F^{-2}\boldsymbol{B}) + \circ(\gamma^2)\Big) \\
&= \gamma^{-1}\frac{1}{T}\boldsymbol{I}_F^{-1} + \gamma^0\frac{1}{T}(\boldsymbol{I}_F^{-2}\boldsymbol{A} + \boldsymbol{I}_F^{-1}) + \gamma^1\frac{1}{T}[\boldsymbol{I}_F^{-1}(\boldsymbol{I}_F^{-1}\boldsymbol{A})^2 - \boldsymbol{I}_F^{-2}\boldsymbol{B} + \boldsymbol{I}_F^{-2}\boldsymbol{A}] + \circ(\gamma) \ .
\end{aligned}$$

Subtracting the second term from the first term and applying the trace, we finally write the MSE :

$$\mathrm{MSE}_{\mathrm{NCE}} = \mathrm{tr}\Big(\gamma^{-1}\frac{1}{T}\big(\boldsymbol{I}_F^{-1} - \boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1}\big) + \tag{4.16}$$

$$\gamma^0\frac{1}{T}\big(\boldsymbol{I}_F^{-2}\boldsymbol{A} + \boldsymbol{I}_F^{-1} - 2\boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1}\big)\Big) + \circ(\gamma) \tag{4.17}$$

- Optimize the $\mathrm{MSE}_{\mathrm{NCE}}$ w.r.t. p_n
  To optimize w.r.t. p_n, we need only keep the two first orders of the $\mathrm{MSE}_{\mathrm{NCE}}$, which depends on p_n only via the term $\mathrm{tr}(\boldsymbol{I}_F^{-2}\boldsymbol{A}) = \int \|\boldsymbol{I}_F^{-1}\boldsymbol{g}(x)\|^2\frac{p_d^2}{p_n}(x)d\boldsymbol{x}$. Hence, we need to optimize

$$J(p_n) = \frac{1}{T}\int \|I_F^{-1}\boldsymbol{g}(\boldsymbol{x})\|^2\frac{p_d^2}{p_n}(\boldsymbol{x})d\boldsymbol{x} \tag{4.18}$$

  with respect to p_n. We compute the variational (Fréchet) derivative together with the Lagrangian of the constraint $\int p_n(\boldsymbol{x}) = 1$ (with λ denoting the Lagrangian multiplier) to obtain

$$\delta_{p_n}J = -\|\boldsymbol{I}_F^{-1}\boldsymbol{g}\|^2\frac{p_d^2}{p_n^2} + \lambda \ . \tag{4.19}$$

  Setting this to zero and taking into account the non-negativity of p_n gives

$$p_n(\boldsymbol{x}) = \|\boldsymbol{I}_F^{-1}\boldsymbol{g}(x)\|p_d(\boldsymbol{x})/Z \tag{4.20}$$

  where $Z = \int \|\boldsymbol{I}_F^{-1}\boldsymbol{g}(x)\|p_d(\boldsymbol{x})d\boldsymbol{x}$ is the normalization constant. This is thus the optimal noise distribution, as a first-order approximation.

- Compute the MSE gaps at optimality
  Plugging this optimal p_n into the formula of $\mathrm{MSE}_{\mathrm{NCE}}$ and subtracting the Cramer-Rao MSE (which is a lower bound for a normalized model), we get :

$$\begin{aligned}\Delta_{\mathrm{opt}}\mathrm{MSE}_{\mathrm{NCE}} &= \mathrm{MSE}_{\mathrm{NCE}}(p_n = p_n^{\mathrm{opt}}) - \mathrm{MSE}_{\mathrm{Cramer-Rao}} \\ &= \frac{1}{T}\Big(\int \|I_F^{-1}\boldsymbol{\psi}\| p_d\Big)^2 \quad .\end{aligned}$$

  This is interesting to compare with the case where the noise distribution is the data distribution, which gives

$$\begin{aligned}\Delta_{\mathrm{data}}\mathrm{MSE}_{\mathrm{NCE}} &= \mathrm{MSE}_{\mathrm{NCE}}(p_n = p_d) - \mathrm{MSE}_{\mathrm{Cramer-Rao}} \\ &= \frac{1}{T}\int \|I_F^{-1}\boldsymbol{\psi}\|^2 p_d\end{aligned}$$

  where the squaring is in a different place. In fact, we can compare these two quantities by the Cauchy-Schwartz inequality, or simply the fact that

$$\begin{aligned}\Delta\mathrm{MSE}_{\mathrm{NCE}} &= \Delta_{\mathrm{data}}\mathrm{MSE}_{\mathrm{NCE}} - \Delta_{\mathrm{opt}}\mathrm{MSE}_{\mathrm{NCE}} \\ &= \mathrm{MSE}_{\mathrm{NCE}}(p_n = p_d) - \mathrm{MSE}_{\mathrm{NCE}}(p_n = p_n^{\mathrm{opt}}) \\ &= \frac{1}{T}\mathsf{Var}_{X\sim p_d}\{\|I_F^{-1}\boldsymbol{g}(\mathbf{X})\|\}\end{aligned}$$

  This implies that the two MSEs, when when the noise distribution is either p_n^{opt} or p_d, can be equal only if $\|I_F^{-1}\boldsymbol{g}(.)\|$ is constant in the support of p_d. This does not seem to be possible for any reasonable distribution.

Proof : case where $p_n \approx p_d$

We consider the limit case where $\frac{p_d}{p_n}(\boldsymbol{x}) = 1 + \epsilon(\boldsymbol{x})$ with $|\epsilon(\boldsymbol{x}) - 0| < \epsilon_{\max} \quad \forall \boldsymbol{x}$.

Note that in order to use Taylor expansions for terms containing $\epsilon(\boldsymbol{x})$ in an integral, we assume for any integrand $h(\boldsymbol{x})$ that $\int h(\boldsymbol{x})\epsilon(\boldsymbol{x})d\boldsymbol{x} \approx \epsilon \int h(\boldsymbol{x})d\boldsymbol{x}$, where ϵ would be a constant.

- Taylor expansion of the discriminator

$$\begin{aligned}1 - D(\boldsymbol{x}) &= \frac{\nu p_n(\boldsymbol{x})}{p_d(\boldsymbol{x}) + \nu p_n(\boldsymbol{x})} = \frac{1}{1 + \frac{1}{\nu} + \frac{p_d}{p_n}(\boldsymbol{x})} = \frac{1}{1 + \frac{1}{\nu} + \frac{1}{\nu}\epsilon(\boldsymbol{x})} \\ &= \frac{\nu}{1+\nu}\epsilon^0(\boldsymbol{x}) - \frac{\nu}{(1+\nu)^2}\epsilon^1(\boldsymbol{x}) + \frac{\nu}{(1+\nu)^3}\epsilon^2(\boldsymbol{x}) + \circ(\epsilon^2)\end{aligned}$$

- Evaluating the integrals $\boldsymbol{m}$, $\boldsymbol{I}$

$$\begin{aligned}\boldsymbol{m} &= \int \boldsymbol{g}(\boldsymbol{x})p_d(\boldsymbol{x})\Big(1 - D(\boldsymbol{x})\Big)d\boldsymbol{x} \\ &= \int \boldsymbol{g}(\boldsymbol{x})p_d(\boldsymbol{x})\Big(\frac{\nu}{1+\nu}\epsilon^0(\boldsymbol{x}) - \frac{\nu}{(1+\nu)^2}\epsilon^1(\boldsymbol{x}) + \frac{\nu}{(1+\nu)^3}\epsilon^2(\boldsymbol{x}) + \circ(\epsilon^2)\Big)d\boldsymbol{x}\end{aligned}$$

$$= \frac{\nu}{1+\nu}\boldsymbol{m}_F - \frac{\nu}{(1+\nu)^2}\boldsymbol{a}(\epsilon) + \frac{\nu}{(1+\nu)^3}\boldsymbol{b}(\epsilon^2) + \circ(\epsilon^3)$$

where the Fisher-score mean $\boldsymbol{m}_F$ is null and we use shorthand notations a and b for the remaining integrals :

$$\begin{aligned}
\boldsymbol{m}_F &= \int \boldsymbol{g}(\boldsymbol{x})p_d(\boldsymbol{x})d\boldsymbol{x} \\
\boldsymbol{a}(\epsilon) &= \int \boldsymbol{g}(\boldsymbol{x})p_d\epsilon(\boldsymbol{x})d\boldsymbol{x} \\
\boldsymbol{b}(\epsilon^2) &= \int \boldsymbol{g}(\boldsymbol{x})p_d\epsilon^2(\boldsymbol{x})d\boldsymbol{x} \ .
\end{aligned}$$

Similarly,

$$\begin{aligned}
\boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x})\Big(1 - D(\boldsymbol{x})\Big)d\boldsymbol{x} \\
&= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x})\Big(\frac{\nu}{1+\nu}\epsilon^0(\boldsymbol{x}) - \frac{\nu}{(1+\nu)^2}\epsilon^1(\boldsymbol{x}) + \frac{\nu}{(1+\nu)^3}\epsilon^2(\boldsymbol{x}) \\
&+ \circ(\epsilon^2)\Big)d\boldsymbol{x} = \frac{\nu}{1+\nu}\boldsymbol{I}_F - \frac{\nu}{(1+\nu)^2}\boldsymbol{A}(\epsilon) + \frac{\nu}{(1+\nu)^3}\boldsymbol{B}(\epsilon^2) + \circ(\epsilon^3)
\end{aligned}$$

where the Fisher-score covariance (Fisher information) is $\boldsymbol{I}_F$ and we use shorthand notations A and B for the remaining integrals :

$$\begin{aligned}
\boldsymbol{I}_F &= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x})d\boldsymbol{x} \\
\boldsymbol{A}(\epsilon) &= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p_d\epsilon(\boldsymbol{x})d\boldsymbol{x} \\
\boldsymbol{B}(\epsilon^2) &= \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top p_d\epsilon^2(\boldsymbol{x})d\boldsymbol{x} \ .
\end{aligned}$$

- Taylor expansion of $\boldsymbol{I}^{-1}$

$$\begin{aligned}
\boldsymbol{I}^{-1} &= \Big(\frac{\nu}{1+\nu}\boldsymbol{I}_F - \frac{\nu}{(1+\nu)^2}\boldsymbol{A}(\epsilon) + \frac{\nu}{(1+\nu)^3}\boldsymbol{B}(\epsilon^2) + \circ(\epsilon^3)\Big)^{-1} \\
&= \frac{1+\nu}{\nu}\boldsymbol{I}_F^{-1} + \frac{1}{\nu}\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon) + \frac{\nu}{1+\nu}\boldsymbol{I}_F^{-2}\big(\boldsymbol{I}_F^{-1}\boldsymbol{A}^2(\epsilon) - \boldsymbol{B}(\epsilon^2)\big) + \circ(\epsilon^3)
\end{aligned}$$

- Evaluating the $\mathrm{MSE}_{\mathrm{NCE}}$

$$\begin{aligned}
\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1} = \boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1} + \frac{1}{(1+\nu)^2}\Big(&\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon)\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon) + \\
&\boldsymbol{I}_F^{-1}\boldsymbol{a}(\epsilon)\boldsymbol{a}(\epsilon)^\top\boldsymbol{I}_F^{-1}\Big) + \circ(\epsilon^3)
\end{aligned}$$

by plugging in the Taylor expansions of $\boldsymbol{I}^{-1}$ and $\boldsymbol{m}$ and retaining only terms up to the second order. Finally, the MSE becomes :

$$\begin{aligned}
&\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) \\
&= \frac{\nu+1}{T}\mathrm{tr}(\boldsymbol{I}^{-1} - \frac{\nu+1}{\nu}(\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1})) \\
&= \mathrm{tr}\Big(\frac{(1+\nu)^2}{T\nu}(\boldsymbol{I}_F^{-1} - \boldsymbol{I}_F^{-1}\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-1}) + \frac{1+\nu}{T\nu}\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon) + \\
&\frac{1}{T\nu}\big(\boldsymbol{I}_F^{-3}\boldsymbol{A}^2(\epsilon) - \boldsymbol{I}_F^{-2}\boldsymbol{B}(\epsilon^2) - \boldsymbol{I}_F^{-1}\boldsymbol{a}(\epsilon)\boldsymbol{a}(\epsilon)^\top\boldsymbol{I}_F^{-1} - \\
&\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon)\boldsymbol{m}_F\boldsymbol{m}_F^\top\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon)\big)\Big) + \circ(\epsilon^3)
\end{aligned}$$

- Optimize the $\mathrm{MSE}_{\mathrm{NCE}}$ w.r.t. p_n
  To optimize w.r.t. p_n, we need only keep the $\mathrm{MSE}_{\mathrm{NCE}}$ up to order 1, which depends on p_n only via the term

$$\mathrm{tr}(\boldsymbol{I}_F^{-2}\boldsymbol{A}(\epsilon)) = \mathrm{tr}\Big(\boldsymbol{I}_F^{-2}\big(\int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top\frac{p_d^2}{p_n}(x)d\boldsymbol{x} - \boldsymbol{I}_F\big)\Big)$$

  . where we unpacked p_n from $\epsilon = \frac{p_d}{p_n} - 1$. Hence, we need to optimize

$$J(p_n) = \frac{1}{T}\int \|I_F^{-1}\boldsymbol{g}(\boldsymbol{x})\|^2\frac{p_d^2}{p_n}(\boldsymbol{x})d\boldsymbol{x} \tag{4.21}$$

  with respect to p_n. This was already done in the all-noise limit $\nu \to \infty$ and yielded

$$p_n(\boldsymbol{x}) = \|\boldsymbol{I}_F^{-1}\boldsymbol{g}(x)\|p_d(\boldsymbol{x})/Z \tag{4.22}$$

  where $Z = \int \|\boldsymbol{I}_F^{-1}\boldsymbol{g}(x)\|p_d(\boldsymbol{x})d\boldsymbol{x}$ is the normalization constant. This is thus the optimal noise distribution, as a first-order approximation.

In the third case, the limit of all data, we have the following conjecture :

Conjecture 1 *In case (iii), the limit of all data samples $\nu \to 0$, the optimal noise distribution is such that it is all concentrated at the set of those $\boldsymbol{\xi}$ which are given by*

$$\begin{aligned}
&\arg\max_{\boldsymbol{\xi}} p_d(\boldsymbol{\xi})\mathrm{tr}\Big((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-1}\Big)^{-1} \\
&\text{s.t.} \quad \boldsymbol{g}(\xi) = \text{constant}
\end{aligned} \tag{4.23}$$

Informal and heuristic *"proof"* :

We have the $\mathrm{MSE}_{\mathrm{NCE}}(T, \nu, p_n) = \frac{\nu+1}{T}\mathrm{tr}(\boldsymbol{I}^{-1} - \frac{\nu+1}{\nu}(\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1}))$.

Given the term up until ν^{-1} in the MSE, we will use Taylor expansions up to order 2 throughout the proof, in anticipation that the MSE will be expanded until order 1.

Note that in this no noise limit, the assumption made by Gutmann and Hyvärinen (2012) that p_n is non-zero whenever p_d is nonzero is not true for this optimal p_n, which reduces the rigour of this analysis. (This we denote by heuristic approximation 1.)

- Taylor expansion of the discriminator

$$1 - D(\boldsymbol{x}) = \frac{\nu p_n(\boldsymbol{x})}{p_d(\boldsymbol{x}) + \nu p_n(\boldsymbol{x})} = \frac{1}{1 + \frac{1}{\nu}\frac{p_d}{p_n}(\boldsymbol{x})} = \nu \frac{p_n}{p_d}(\boldsymbol{x}) - \nu^2 \frac{p_n^2}{p_d^2}(\boldsymbol{x}) + \circ(\nu^2)$$

- Evaluating the integrals $\boldsymbol{m}$, $\boldsymbol{I}$

$$\begin{aligned}
\boldsymbol{m} &= \int \boldsymbol{g}(\boldsymbol{x}) p_d(\boldsymbol{x}) \Big(1 - D(\boldsymbol{x})\Big) d\boldsymbol{x} \\
&= \int \boldsymbol{g}(\boldsymbol{x}) p_d(\boldsymbol{x}) \Big(\nu \frac{p_n}{p_d}(\boldsymbol{x}) - \nu^2 \frac{p_n^2}{p_d^2}(\boldsymbol{x}) + \circ(\nu^2)\Big) d\boldsymbol{x} \\
&= \nu \boldsymbol{m}_n - \nu^2 \boldsymbol{b} + \circ(\nu^2)
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{m}_n &= \int \boldsymbol{g}(\boldsymbol{x}) p_n(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{b} &= \int \boldsymbol{g}(\boldsymbol{x}) \frac{p_n^2}{p_d}(\boldsymbol{x}) d\boldsymbol{x} \quad .
\end{aligned}$$

Similarly,

$$\begin{aligned}
\boldsymbol{I} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x}) \Big(1 - D(\boldsymbol{x})\Big) d\boldsymbol{x} \\
&= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_d(\boldsymbol{x}) \Big(\nu \frac{p_n}{p_d}(\boldsymbol{x}) - \nu^2 \frac{p_n^2}{p_d^2}(\boldsymbol{x}) + \circ(\nu^2)\Big) d\boldsymbol{x} \\
&= \nu \boldsymbol{I}_n - \nu^2 \boldsymbol{B} + \circ(\nu^2)
\end{aligned}$$

where the Fisher-score covariance (Fisher information) is $\boldsymbol{I}_F$ and we use shorthand notations A and B for the remaining integrals :

$$\begin{aligned}
\boldsymbol{I}_n &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top p_n(\boldsymbol{x}) d\boldsymbol{x} \\
\boldsymbol{B} &= \int \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x})^\top \frac{p_n^2}{p_d}(\boldsymbol{x}) d\boldsymbol{x} \quad .
\end{aligned}$$

- Taylor expansion of $\boldsymbol{I}^{-1}$

$$\begin{aligned}
\boldsymbol{I}^{-1} &= \Big(\nu \boldsymbol{I}_n - \nu^2 \boldsymbol{B} + \circ(\nu^2)\Big)^{-1} \\
&= \Big(\nu \boldsymbol{I}_n (\mathbf{Id} - \nu \boldsymbol{I}_n^{-1} \boldsymbol{B}) + \circ(\nu^2)\Big)^{-1} \\
&= \nu^{-1} \boldsymbol{I}_n^{-1} \Big(\mathbf{Id} + \nu \boldsymbol{I}_n^{-1} \boldsymbol{B} + \nu^2 (\boldsymbol{I}_n^{-1} \boldsymbol{B})^2 + \nu^3 (\boldsymbol{I}_n^{-1} \boldsymbol{B})^3 + \circ(\nu^3)\Big) + \circ(\nu^2) \\
&= \nu^{-1} \boldsymbol{I}_n^{-1} + \nu^0 \boldsymbol{I}_n^{-2} \boldsymbol{B} + \nu^1 \boldsymbol{I}_n^{-1} (\boldsymbol{I}_n^{-2} \boldsymbol{B})^2 + \nu^2 \boldsymbol{I}_n^{-1} (\boldsymbol{I}_n^{-2} \boldsymbol{B})^3 + \circ(\nu^2)
\end{aligned}$$

- Evaluating the $\mathrm{MSE}_{\mathrm{NCE}}$

$$\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1} = \\ \nu^0(\boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-1}) + \nu^2(\boldsymbol{I}_n^{-1}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{I}_n^{-1} + \boldsymbol{I}_n^{-2}\boldsymbol{B}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-2}\boldsymbol{B}) + \circ(\nu^2)$$

by plugging in the Taylor expansions of $\boldsymbol{I}^{-1}$ and $\boldsymbol{m}$ and retaining only terms up to the second order. Hence, the second term of the MSE without the trace is

$$\begin{aligned}
&\Big(\nu^1 T^{-1} + \nu^0 2T^{-1} + \nu^{-1}T^{-1}\Big)\boldsymbol{I}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{I}^{-1} \\
&= \Big(\nu^1 T^{-1} + \nu^0 2T^{-1} + \nu^{-1}T^{-1}\Big)\Big(\nu^0(\boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-1}) + \\
&\nu^2(\boldsymbol{I}_n^{-1}\boldsymbol{b}\boldsymbol{b}^\top\boldsymbol{I}_n^{-1} + \boldsymbol{I}_n^{-2}\boldsymbol{B}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-2}\boldsymbol{B}) + \circ(\nu^2)\Big) \\
&= \nu^{-1}\frac{1}{T}(\boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-1}) + \nu^0\frac{1}{T}(2\boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-1}) + \\
&\nu^1\frac{1}{T}(\boldsymbol{I}_n^{-1}\boldsymbol{b}_n\boldsymbol{b}_n^\top\boldsymbol{I}_n^{-1} + \boldsymbol{I}_n^{-2}\boldsymbol{B}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-2}\boldsymbol{B} + \boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^\top\boldsymbol{I}_n^{-1}) + \circ(\nu)
\end{aligned}$$

and the first term of the MSE without the trace is

$$\begin{aligned}
&\Big(\nu^0 T^{-1} + \nu^1 T^{-1}\Big)\mathrm{tr}(\boldsymbol{I}^{-1}) \\
&= \Big(\nu^0 T^{-1} + \nu^1 T^{-1}\Big)\Big(\nu^{-1}\boldsymbol{I}_n^{-1} + \nu^0\boldsymbol{I}_n^{-2}\boldsymbol{B} + \nu^1\boldsymbol{I}_n^{-1}(\boldsymbol{I}_n^{-2}\boldsymbol{B})^2 + \\
&\nu^2\boldsymbol{I}_n^{-1}(\boldsymbol{I}_n^{-2}\boldsymbol{B})^3 + \circ(\nu^2)\Big) \\
&= \nu^{-1}\frac{1}{T}\boldsymbol{I}_n^{-1} + \nu^0\frac{1}{T}(\boldsymbol{I}_n^{-2}\boldsymbol{B} + \boldsymbol{I}_n^{-1}) + \nu^1\frac{1}{T}[\boldsymbol{I}_n^{-1}(\boldsymbol{I}_n^{-1}\boldsymbol{B})^2 + \boldsymbol{I}_n^{-2}\boldsymbol{B}] + \circ(\nu) \ .
\end{aligned}$$

Subtracting the second term from the first term and applying the trace, we finally write the MSE :

$$\begin{aligned}
\mathrm{MSE}_{\mathrm{NCE}} = \mathrm{tr}(&\nu^{-1}\frac{1}{T}(\boldsymbol{I}_n^{-1} - \boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-1}) + \nu^0\frac{1}{T}(\boldsymbol{I}_n^{-2}\boldsymbol{B} + \boldsymbol{I}_n^{-1} \\
&- 2\boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-1}) + \nu^1\frac{1}{T}[\boldsymbol{I}_n^{-1}(\boldsymbol{I}_n^{-1}\boldsymbol{B})^2 + \boldsymbol{I}_n^{-2}\boldsymbol{B} - \boldsymbol{I}_n^{-1}\boldsymbol{b}_n\boldsymbol{b}_n^T\boldsymbol{I}_n^{-1} - \\
&\boldsymbol{I}_n^{-2}\boldsymbol{B}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-2}\boldsymbol{B} - \boldsymbol{I}_n^{-1}\boldsymbol{m}_n\boldsymbol{m}_n^T\boldsymbol{I}_n^{-1}] + \circ(\nu)) \ .
\end{aligned}$$

Rewriting $\boldsymbol{I}_n^{-1} = \boldsymbol{I}_n^{-1}\boldsymbol{I}_n\boldsymbol{I}_n^{-1}$, using the circular invariance of the trace operator and stopping at order ν^0, we get :

$$\begin{aligned}
\mathrm{MSE}_{\mathrm{NCE}} &= \nu^{-1}\frac{1}{T}\langle\boldsymbol{I}_n^{-2}, \boldsymbol{I}_n - \boldsymbol{m}_n\boldsymbol{m}_n^\top\rangle + \nu^0\frac{1}{T}\langle\boldsymbol{I}_n^{-2}, \boldsymbol{B} + \boldsymbol{I}_n - 2\boldsymbol{m}_n\boldsymbol{m}_n^\top\rangle + \circ(1) \\
&= \nu^{-1}\frac{1}{T}\langle\boldsymbol{I}_n^{-2}, \mathrm{Var}_{N\sim p_n}\boldsymbol{g}(\mathbf{N})\rangle + \nu^0\frac{1}{T}\langle\boldsymbol{I}_n^{-2}, \boldsymbol{B} + \boldsymbol{I}_n - 2\boldsymbol{m}_n\boldsymbol{m}_n^\top\rangle + \circ(1) \ . \qquad (4.24)
\end{aligned}$$

- Optimize the $\mathrm{MSE}_{\mathrm{NCE}}$ w.r.t. p_n
  Looking at the above MSE, the dominant term of order ν^{-1} is $\langle \boldsymbol{I}_n^{-2}, \mathrm{Var}_{N\sim p_n}\boldsymbol{g}(\mathbf{N})\rangle \geq 0$ is minimized when it is 0, that is, when $\boldsymbol{g}$ is constant in the support of p_n. Typically this means that p_n is concentrated on a set of zero measure. In the 1D case, such case is typically the Dirac delta $p_n = \delta_z$, or a distribution with two deltas in case of symmetrical $\boldsymbol{g}$.
  We can plug this in the terms of the next order ν^0, which remain to be minimized :

$$\begin{aligned}\langle \boldsymbol{I}_n^{-2}, \boldsymbol{B} + \boldsymbol{I}_n - 2\boldsymbol{m}_n\boldsymbol{m}_n^T\rangle &= \langle \boldsymbol{I}_n^{-2}, \boldsymbol{B} - \boldsymbol{I}_n + 2\boldsymbol{I}_n - 2\boldsymbol{m}_n\boldsymbol{m}_n^T\rangle \\ &= \langle \boldsymbol{I}_n^{-2}, \boldsymbol{B} - \boldsymbol{I}_n + 2\mathrm{Var}_{N\sim p_n}\boldsymbol{g}(\mathbf{N})\rangle \\ &= \langle \boldsymbol{I}_n^{-2}, \boldsymbol{B} - \boldsymbol{I}_n\rangle\end{aligned}$$

  given we chose p_n so that the variance is 0.
  The integrands of $\boldsymbol{B}$ and $\boldsymbol{I}$ respectively involve p_n^2 and p_n. Because p_n is concentrated on a set of zero measure (Dirac-like), the term in $\boldsymbol{B}$ dominates the term in $\boldsymbol{I}$. This is because if we consider the p_n as the limit of a sequence of some proper pdf's, the value of the pdf gets infinite in the support of that pdf in the limit, and thus p_n^2 is infinitely larger than p_n. Hence we are left with $\langle \boldsymbol{I}_n^{-2}, \boldsymbol{B}\rangle$.
  The integral with respect to p_n simplifies to simply evaluating the $\boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top/p_d(\boldsymbol{x})$ the support of p_n. Since we know that $\boldsymbol{g}(\boldsymbol{x})$ is constant in that set, the main question is whether p_d is constant in that set as well. Here, we heuristically assume that it is; this is intuitively appealing in many cases, if not necessarily true. (This we denote by heuristic approximation 2.)
  Thus, we have

$$\int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top \frac{\delta_z^2}{p_d}(\boldsymbol{x})d\boldsymbol{x} \approx c\, \boldsymbol{g}(\boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})^\top \frac{1}{p_d(\boldsymbol{z})}$$

  for some constant c taking into account the effect of squaring of p_n (it is ultimately infinite, but the reasoning is still valid in any sequence going to the limit.)
  Next we make heuristic approximation 3 : we neglect any problems of inversion of singular, rank 1 matrices (note this is not a problem in the 1D case), and further obtain

$$\begin{aligned}\langle \boldsymbol{I}_n^{-2}, \boldsymbol{B}\rangle &\approx \mathrm{tr}\Big((\boldsymbol{g}(\boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})^\top)^{-1}\boldsymbol{g}(\boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})^\top \frac{1}{p_d(\boldsymbol{z})}(\boldsymbol{g}(\boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})^\top)^{-1}\Big) \\ &\approx \frac{1}{p_d(\boldsymbol{z})}\mathrm{tr}\Big((\boldsymbol{g}(\boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})^\top)^{-1}\Big) \ . \end{aligned} \tag{4.25}$$

  Minimizing this term is equivalent to the following maximization setup (still applying heuristic approximation 3) :

$$\arg\max_{\boldsymbol{\xi}} p_d(\boldsymbol{\xi})\mathrm{tr}\Big((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-1}\Big)^{-1} \ .$$

  Those points z obtained by the above condition are the best candidates for p_n to concentrate its mass on.
  We arrived this result by making three heuristic approximations as explained above; we hope to be able to remove some of them in future work.

Numerically, evaluating the optimal noise in the all-data limit requires computing a weight $w(\boldsymbol{x}) = \mathrm{tr}\Big((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-1}\Big)^{-1}$ that is intractable in dimensions bigger than 1, due to the singularity of the rank 1 matrix. We can avoid this numerically by introducing an (infinitesimal) perturbation $\epsilon > 0$ which removes the singularity problem. Using the Sherman-Morrison formula,

$$\begin{aligned}
w_\epsilon(\xi) &= \mathrm{tr}\Big((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top + \epsilon \mathrm{Id})^{-1}\Big)^{-1} \\
&= \mathrm{tr}\Big(\epsilon^{-1}\mathrm{Id} - \frac{1}{\epsilon^2 + \epsilon \boldsymbol{g}(\xi)^\top \mathrm{Id} \boldsymbol{g}(\xi)} \boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top\Big)^{-1} \\
&= \Big(\epsilon^{-1} d - \frac{1}{\epsilon^2 + \epsilon \|\boldsymbol{g}(\xi)\|^2} \|\boldsymbol{g}(\xi)\|^2\Big)^{-1} \\
&= \Big(\epsilon^{-1}(d-1) + \epsilon^0 \frac{1}{\|\boldsymbol{g}(\xi)\|^2} + \epsilon^1 \frac{-1}{\|\boldsymbol{g}(\xi)\|^4} + O(\epsilon^2)\Big)^{-1} \\
&= \epsilon \frac{1}{d-1} + \epsilon^2 \frac{-1}{\|\boldsymbol{g}(\xi)\|^2 (d-1)^2} + \epsilon^3 \frac{(2-d)}{\|\boldsymbol{g}(\xi)\|^4 (d-1)^3} + O(\epsilon^4)
\end{aligned}$$

where we go up to order 3 in the Taylor expansion to ensure the weight $w_\epsilon(\xi)$ is positive. Finally, we can approximate the $\arg\max$ operator with its relaxation $\mathrm{soft}\arg\max^\epsilon(x) = \frac{e^{\frac{x}{\epsilon}}}{\int e^{\frac{x}{\epsilon}} dx}$, so that

$$p_n(\boldsymbol{x}) \approx \mathrm{soft}\arg\max^{\epsilon_1} \big(p_d(\boldsymbol{x}) w_{\epsilon_2}(\boldsymbol{x})\big)$$

where $(\epsilon_1, \epsilon_2) \in (\mathbb{R}_+^*)^2$ are two hyperparameters taken close to zero.

4.7.5 . Optimal Noise for Estimating a Distribution : Proofs

So far, we have optimized hyperparameters (such as the noise distribution) so that the reduce the uncertainty of the *parameter* estimation, measured by the Mean Squared Error $\mathbb{E}\big[\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^2\big] = \frac{1}{T_d}\mathrm{tr}(\boldsymbol{\Sigma})$.

Sometimes, we might wish to reduce the uncertainty of the *distribution* estimation, which we can measure using the Kullback-Leibler (KL) divergence $\mathbb{E}\big[\mathcal{D}_{\mathrm{KL}}(p_d, p_{\hat{\boldsymbol{\theta}}_T})\big]$.

We can specify this error, by using the Taylor expansion of the estimated $\hat{\theta}_T$ near optimality, given in [11] :

$$\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^* = \boldsymbol{z} + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^2) \tag{4.26}$$

where $z \sim \mathcal{N}(0, \frac{1}{T_d}\boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma}$ is the asymptotic variance matrix.

We can similarly take the Taylor expansion of the KL divergence with respect to its second argument, near optimality :

$$\begin{aligned}
J(\hat{\boldsymbol{\theta}}_T) &:= \mathcal{D}_{\mathrm{KL}}(p_d, p_{\hat{\boldsymbol{\theta}}_T}) \\
&= J(\boldsymbol{\theta}^*) + < \nabla_\theta J(\theta^*), \hat{\theta}_T - \theta^* > + \frac{1}{2} < (\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*), \nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^*)\,(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*) > \\
&+ O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^3)
\end{aligned}$$

$$= J(\boldsymbol{\theta}^*) + < \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^*), \hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*) > + \frac{1}{2} \left\| \hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^* \right\|^2_{\nabla^2_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^*)} + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^3)$$

Note that some simplifications occur :

- $J(\boldsymbol{\theta}^*) = \mathcal{D}_{\mathrm{KL}}(p_{\boldsymbol{\theta}^*}, p_{\boldsymbol{\theta}^*}) = 0$
- $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^*) = 0$ as the gradient the KL divergence at $\boldsymbol{\theta}^*$ is the mean of the (negative) Fisher score, which is null.
- $\nabla^2_{\boldsymbol{\theta}} J(\theta^*) = \boldsymbol{I}_F$

Plugging in the estimation error 4.26 into the distribution error yields :

$$\begin{aligned} J(\hat{\boldsymbol{\theta}}_T) &= \frac{1}{2} \left\| \boldsymbol{z} + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^2) \right\|^2_{\boldsymbol{I}_F} + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^3) \\ &= \frac{1}{2} \left(\|\boldsymbol{z}\|^2_{\boldsymbol{I}_F} + 2 < \boldsymbol{z}, O(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2) >_{\boldsymbol{I}_F} + \left\| O(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2) \right\|^2_{\boldsymbol{I}_F} \right) + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^3) \\ &= \frac{1}{2} \|\boldsymbol{z}\|^2_{\boldsymbol{I}_F} + O(\|\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*\|^2) \end{aligned}$$

by truncating the Taylor expansion to the first order. Hence up to the first order, the expectation yields :

$$\begin{aligned} \mathbb{E}\big[\mathcal{D}_{\mathrm{KL}}(p_d, p_{\hat{\boldsymbol{\theta}}_T})\big] &= \frac{1}{2} \mathbb{E}\big[\|\boldsymbol{z}\|^2_{\boldsymbol{I}_F}\big] = \frac{1}{2} \mathbb{E}\big[\boldsymbol{z}^T \boldsymbol{I}_F \boldsymbol{z}\big] = \frac{1}{2} \mathbb{E}\big[\mathrm{tr}(\boldsymbol{z}^T \boldsymbol{I}_F \boldsymbol{z})\big] = \frac{1}{2} \mathbb{E}\big[\mathrm{tr}(\boldsymbol{I}_F \boldsymbol{z} \boldsymbol{z}^T)\big] \\ &= \frac{1}{2} \mathrm{tr}(\boldsymbol{I}_F \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]) = \frac{1}{2} \mathrm{tr}(\boldsymbol{I}_F \mathrm{Var}[\boldsymbol{z}]) = \frac{1}{2T_d} \mathrm{tr}(\boldsymbol{I}_F \boldsymbol{\Sigma}) \end{aligned}$$

Note that this is a general and known result which is applicable beyond the KL divergence : for any divergence, the 0th order term is null as it measures the divergence between the data distribution and itself, the 1st order term is null in expectation if the estimator $\hat{\theta}_T$ is asymptotically unbiased, which leaves an expected error given by the 2nd-order term $\frac{1}{2T_d} \mathrm{tr}(\nabla^2 J \boldsymbol{\Sigma})$ where J is the chosen divergence. Essentially, one would replace the Fisher Information above, which is the Hessian for a forward-KL divergence, by the Hessian for a given divergence.

Finding the optimal noise that minimizes the distribution error means minimizing $\frac{1}{T_d} \mathrm{tr}(\Sigma I_F)$. Contrast that with the optimal noise that minimizes the parameter estimation error (asymptotic variance) $\frac{1}{T_d} \mathrm{tr}(\Sigma)$. We can reprise each of the three limit cases from the previous proofs, and derive novel optimal noise distributions :

Theorem 2 *In the two limit cases of Theorem 1, the noise distribution minimizing the expected Kullback-Leibler divergence is given by*

$$p_n^{\mathrm{opt}}(\boldsymbol{x}) \propto p_d(\boldsymbol{x}) \|\boldsymbol{I}_F^{-\frac{1}{2}} \boldsymbol{g}(\boldsymbol{x})\| \ . \tag{4.27}$$

Proof : case of $\nu \to \infty$

We recall the asymptotic variance $\frac{1}{T_d}\Sigma$ in the all-noise limit is given by equation 4.17 at the first order and without the trace. Multiplying by I_F introduces no additional dependency in p_n, hence we retain the only term dependent that was dependent on p_n, $\boldsymbol{I}_F^{-2} \int \boldsymbol{g}(x) \boldsymbol{g}(x)^\top \frac{p_d^2}{p_n}(\boldsymbol{x}) d\boldsymbol{x}$, multiply it with I_F and take the trace. This yields the following cost to minimize :

$$J(p_n) = \frac{1}{T} \int \|I_F^{-\frac{1}{2}} \boldsymbol{g}(\boldsymbol{x})\|^2 \frac{p_d^2}{p_n}(\boldsymbol{x}) d\boldsymbol{x} \tag{4.28}$$

with respect to p_n. As in previous proofs, we compute the variational (Fréchet) derivative together with the Lagrangian of the constraint $\int p_n(\boldsymbol{x}) = 1$ (with λ denoting the Lagrangian multiplier) to obtain

$$\delta_{p_n} J = -\|\boldsymbol{I}_F^{-\frac{1}{2}} \boldsymbol{g}\|^2 \frac{p_d^2}{p_n^2} + \lambda \ . \tag{4.29}$$

Setting this to zero and taking into account the non-negativity of p_n gives

$$p_n(\boldsymbol{x}) = \|\boldsymbol{I}_F^{-\frac{1}{2}} \boldsymbol{g}(x)\| p_d(\boldsymbol{x})/Z \tag{4.30}$$

where $Z = \int \|\boldsymbol{I}_F^{-\frac{1}{2}} \boldsymbol{g}(x)\| p_d(\boldsymbol{x}) d\boldsymbol{x}$ is the normalization constant. This is thus the optimal noise distribution, as a first-order approximation.

In the third case, the limit of all data, we have the following conjecture :

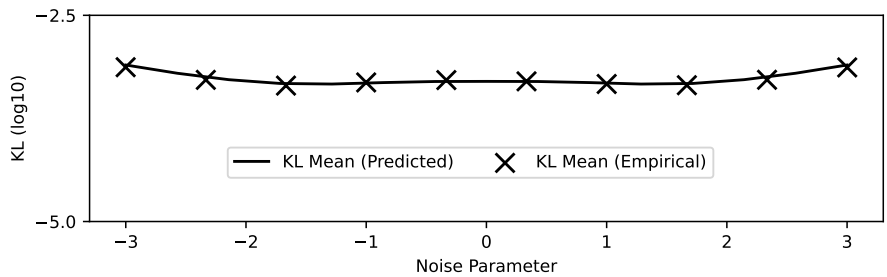
Conjecture 2 *In the limit of Conjecture 1 the noise distribution minimizing the expected Kullback-Leibler divergence is such that it is all concentrated at the set of those $\boldsymbol{\xi}$ which are given by*

$$\begin{aligned} &\arg\max_{\boldsymbol{\xi}} p_d(\boldsymbol{\xi}) \mathrm{tr}\left((\boldsymbol{g}(\xi)\boldsymbol{g}(\xi)^\top)^{-\frac{1}{2}} \right)^{-1} \\ &\text{s.t.} \quad \boldsymbol{g}(\xi) = \text{constant} \end{aligned} \tag{4.31}$$

Proof : case of $\nu \to 0$

By the same considerations, we can obtain the optimal noise that minimizes the asymptotic error in distribution space in the all-data limit, using equation 4.25 with a multiplication by I_F inside the the trace. This leads to the result.

4.7.6 . Numerical Validation of the Predicted Distribution Error



.

Figure 4.7 – KL vs. the noise parameter (Gaussian Mean). The noise proportion is fixed at 50%.

We here numerically validate our formulae predicting the asymptotic estimation error in distribution space $\mathcal{D}_{\mathrm{KL}}(p_d, p_{\hat{\theta}_{\mathrm{NCE}}})$, when the noise is constrained within a parametric family containing the data ; here, the model is a one-dimensional centered Gaussian with unit variance, parameterized by its mean.

5 - Annealed Noise-Contrastive Estimation

This section presents the work published in :
O. Chehab, A. Hyvarinen, A. Risteski. *Provable benefits of annealing for estimating normalizing constants*. Neural Information Processing Systems (NeurIPS), 2023. Spotlight.

5.1 . Summary

Recent research has developed several Monte Carlo methods for estimating the normalization constant (partition function) based on the idea of annealing. This means sampling successively from a path of distributions which interpolate between a tractable “proposal" distribution and the unnormalized “target" distribution. Prominent estimators in this family include annealed importance sampling and annealed noise-contrastive estimation (NCE). Such methods hinge on a number of design choices : which estimator to use, which path of distributions to use and whether to use a path at all ; so far, there is no definitive theory on which choices are efficient. Here, we evaluate each design choice by the asymptotic estimation error it produces. First, we show that using NCE is more efficient than the importance sampling estimator, but in the limit of infinitesimal path steps, the difference vanishes. Second, we find that using the geometric path brings down the estimation error from an exponential to a polynomial function of the parameter distance between the target and proposal distributions. Third, we find that the arithmetic path, while rarely used, can offer optimality properties over the universally-used geometric path. In fact, in a particular limit, the optimal path is arithmetic. Based on this theory, we finally propose a two-step estimator to approximate the optimal path in an efficient way.

5.2 . Introduction

Recent progress in generative modeling has sparked renewed interest in models of data that are defined by an unnormalized distribution. A prominent example is energy-based models, which are increasingly used in deep learning [232], and for which there are a variety of parameter estimation procedures [41, 11, 233, 225]. Another example comes from Bayesian statistics, where the posterior model of parameters given data is frequently known only up to a proportionality constant. Such models can be evaluated and compared by the probability they assign to a dataset, yet this requires computing their normalization constants (partition functions) which are typically high-dimensional, intractable integrals.

Monte-Carlo techniques have been successful at computing these integrals using sampling methods [234]. The most common is importance sampling [234] which draws a sample from a tractable, "proposal" distribution to integrate the unnormalized "target" density. Noise-contrastive estimation (NCE) [11] uses a sample from *both* the proposal and the target, to compute the integral. Yet such methods suffer from high variance, especially when the "gap" between the proposal and target densities

is large [235, 20, 236]. This has motivated various approaches to gradually bridge the gap with intermediate distributions, which is loosely referred to as "annealing". Among them, annealed importance sampling (AIS) [237–239] is widely adopted : it has been used to compute the normalization constants of deep stochastic models [240, 241] or to motivate a lower-bound for learning objectives [242, 243]. To integrate the unnormalized "target" density, it draws a sample from an entire path of distributions between the proposal and the target. While annealed importance sampling has been shown to be effective empirically, its theoretical understanding remains limited [244, 245] : it is yet unclear when annealing is effective, for which annealing paths, and whether AIS is a statistically efficient way to do it.

In this paper, we define a family of *annealed Bregman estimators (ABE)* for the normalization constant. We show that it is general enough to recover many classical estimators as a special case, including importance sampling, noise-contrastive estimation, umbrella sampling [246], bridge sampling [247] and annealed importance sampling. We provide a statistical analysis of its hyperparameters such as the choice of paths, and show the following :

1. First, we establish that using NCE is more asymptotically statistically efficient — in the sense of how many samples from the intermediate distribution need to be generated — than the importance sampling estimator, but in the limit of infinitesimal path steps, the difference vanishes.
2. Second, we find that the near-universally used *geometric path* brings down the estimation error from an exponential to a polynomial function of the parameter distance between the target and proposal distributions.
3. Third, we find that using the recently introduced *arithmetic path* [248] is exponentially inefficient in its basic form, yet it can be reparameterized to be in some sense optimal. Based on this optimality result, we finally propose a two-stage estimation procedure which first finds an approximation of the optimal (arithmetic) path, then uses it to estimate the normalization constant.

5.3 . Background

Importance sampling and NCE The problem considered here is computing the normalization constant[1], *i.e.* the integral of some unnormalized density $f_1(\boldsymbol{x})$ called "target".

Importance sampling and noise-contrastive estimation are two common estimators for that integral which use a random sample drawn from a tractable density $p_0(\boldsymbol{x})$ called "proposal"(Table 5.1, column 3). In fact, they are part of a larger family of Monte-Carlo estimators of the normalizing constant which can be interpreted as solving a binary classification task, aiming to distinguish between a sample drawn from the proposal and another from the target [92], originating from a line of research by Pihlaja et al. [63]. These estimators are summarized in Table 5.1. Each estimator is obtained by minimizing a specific binary classification loss that is identified by a convex function $\phi(\boldsymbol{x})$. For example, minimizing the classification loss identified by $\phi_{\mathrm{IS}}(x) = x \log x$ yields the importance sampling estimator. Similarly, $\phi_{\mathrm{RevIS}}(x) = -\log x$ leads to the reverse importance sampling estimator [249], and $\phi_{\mathrm{NCE}}(x) = x \log x - (1+x)\log((1+x)/2)$ to the noise-contrastive estimator.

1. in this paper we also say we "estimate" the normalization constant, though in classical statistics it is more traditional to use "estimation" when referring to parameters of a statistical model

Table 5.1 – Some estimators of the normalization obtained by minimizing a classification loss, and their estimation error in terms of well-known divergences [92]. For details and definitions, see Appendix 5.8.1.

Name	Loss identified by $\phi(\boldsymbol{x})$	Estimator $\hat{Z}_1$	MSE
IS	$x \log x$	$\mathbb{E}_{p_0} \frac{f_1}{p_0}$	$\frac{1+\nu}{\nu N} \mathcal{D}_{\chi^2}(p_1, p_0)$
RevIS	$-\log x$	$\left(\mathbb{E}_{p_1} \frac{p_0}{f_1}\right)^{-1}$	$\frac{1+\nu}{N} \mathcal{D}_{\chi^2}(p_0, p_1)$
NCE	$x \log x - (1+x) \log(\frac{1+x}{2})$	implicit	$\frac{(1+\nu)^2}{\nu N} \frac{\mathcal{D}_{\mathrm{HM}}(p_1,p_0)}{1-\mathcal{D}_{\mathrm{HM}}(p_1,p_0)}$

Annealed estimators Annealing extends the above "binary" setup, by introducing a sequence of $K+1$ distributions from the proposal to the target (included). The idea will be to draw a sample from *all* these distributions to estimate the integral of the target $f_1(\boldsymbol{x})$.

These intermediate distributions are obtained from a path $(f_t)_{t\in[0,1]}$, defined by interpolating between the proposal p_0 and unnormalized target f_1 : this path is therefore unnormalized. Different interpolation schemes can be chosen. A general one, explained in Figure 1, is to take the q-mean of the proposal and target [248]. Two values of q are of particular interest : $q \to 0$ defines a a near-universal path [245], obtained by taking the geometric mean of the target and proposal, while $q = 1$ defines a path obtained by the arithmetic mean.

Once a path is chosen, it can be uniformly[2] discretized into a sequence of $K+1$ unnormalized densities, denoted by $(f_{k/K})_{k\in[0,K]}$ with corresponding normalizations $(Z_{k/K})_{k\in[0,K]}$. In practice, samples are drawn from the corresponding normalized densities $(p_{k/K})_{k\in[0,K]}$ using Markov Chain Monte Carlo (MCMC). This sampling step incurs a computational cost, which is paid in the hope of reducing the variance of the estimation. It is common in the literature [244, 245] to assume *perfect sampling*, meaning the MCMC has converged and produced exact and independent samples from the distributions along the path, which simplifies the analysis.

Estimation error A measure of "quality" is required to compare different estimation choices, such as whether to anneal and which path to use. Such a measure is given by the Mean Squared Error (MSE), which is generally tractable when written at the first order in the asymptotic limit of a large sample size [54, Eq. 5.20]. These expressions have been derived for estimators obtained by minimimizing a classification loss [92] and are included in Table 5.1. They measure the "gap" between the proposal and target distributions using statistical divergences. Note also that the estimation error depends on the *normalized* target density (column 4), while the estimators are computed using the *unnormalized* target density (column 3). Further details are available in Appendix 5.8.1.

5.4 . Annealed Bregman Estimators of the normalization constant

2. other discretization schemes can be equivalently achieved by re-parameterizing the path [244]

General ($q \in]0,1]$)	$f_t(\boldsymbol{x}) = \left((1-t)p_0(\boldsymbol{x})^q + t f_1(\boldsymbol{x})^q\right)^{\frac{1}{q}}$
Geometric ($q \to 0$)	$f_t(\boldsymbol{x}) = p_0(\boldsymbol{x})^{1-t} \times f_1(\boldsymbol{x})^t$
Arithmetic ($q = 1$)	$f_t(\boldsymbol{x}) = (1-t)p_0(\boldsymbol{x}) + t f_1(\boldsymbol{s})$

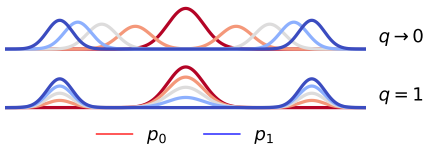


Figure 5.1 – q-mean paths between the proposal and target distributions. The geometric and arithmetic paths are obtained as limit cases. Here, the proposal (red) is a standard Gaussian. The target (blue) is a Gaussian mixture with two modes, and same first and second moments as the proposal. The path of distributions interpolates between blue and red.

The question that we try to answer in this paper is : *How should we choose the $K+1$ distributions in annealing, and how are their samples best used?* To answer this, we will study the error produced by different estimation choices. But first we define the set of estimators for which the analysis is done.

Definition of Annealed Bregman Estimators We now define a new family of estimators, which we call *annealed Bregman estimators (ABE)*; the motivation for this terminology will become clear in the coming paragraphs. We will show that this is a general class of estimators for computing the normalization using a sample drawn from the sequence of $K+1$ distributions. For ABE, the log normalization $\log Z_1$ is estimated additively along the sequence of distributions

$$\widehat{\log Z_1} = \sum_{k=0}^{K-1} \log \widehat{\left(\frac{Z_{(k+1)/K}}{Z_{k/K}}\right)} + \log Z_0 \ . \tag{5.1}$$

Defining the estimation in log-space is analytically convenient, as it is easier to analyze a sum of estimators than a product. Exponentiating the result leads to an estimator of Z_1. We naturally extend the binary setup ($K = 1$) of Chehab et al. [92] and propose to compute each of the intermediate log-ratios, by solving a classification task between samples drawn from their corresponding densities $p_{k/K}$ and $p_{(k+1)/K}$. This is a specific case of a more general framework where each (log) ratio of densities (not just their normalizing constants) is estimated by solving a binary classification task [61]. Each binary classification loss is now identified by a convex function $\phi_k(\boldsymbol{x})$ and defined as

$$\mathcal{L}_k(\beta_k) := \mathbb{E}_{\boldsymbol{x}\sim p_{k/K}}[\phi_k^{'}(r_k(\boldsymbol{x};\beta_k)) \times r_k(\boldsymbol{x};\beta_k) - \phi_k(r_k(\boldsymbol{x};\beta_k))] - \mathbb{E}_{\boldsymbol{x}\sim p_{(k+1)/K}}[\phi_k^{'}(r_k(\boldsymbol{x};\beta_k))], \tag{5.2}$$

where the regression function $r_k(\boldsymbol{x};\beta_k)$ is parameterized by the unknown log-ratio β_k

$$r_k(\boldsymbol{x};\beta_k) = \exp(-\beta_k) \times f_{(k+1)/K}(\boldsymbol{x})/f_{k/K}(\boldsymbol{x}) \ . \tag{5.3}$$

For the true β_k^*, it holds $\beta_k^* = \log(Z_{(k+1)/K}/Z_{k/K})$ and $r_k(\boldsymbol{x};\boldsymbol{\beta}_k^*) = p_{(k+1)/K}(\boldsymbol{x})/p_{k/K}(\boldsymbol{x})$. The convex functions $(\phi_k)_{k\in[0,K-1]}$ which identify the classification losses are called "Bregman" generators, hence ABE. As mentioned above, we assume perfect sampling and allocate the total sample size N equally among the K estimators in the sum.

Hyperparameters The annealed Bregman estimator depends on the following hyperparameters : (1) the choice of path q; (2) the number of distributions along that path $K+1$ (including the proposal and the target); (3) the classification losses identified by the convex functions $(\phi_k)_{k\in[0,K-1]}$.

Different combinations of these hyperparameters recover several common estimators of the log-partition function. In binary case of $K=1$ this includes importance sampling, reverse importance sampling, and noise-contrastive estimation, each obtained for a different choice of the classification loss [92]. To build intuition, consider $K=2$ so that we add a single intermediate distribution $p_{1/2}$ to the sequence. Minimizing the importance sampling loss ($\phi_0 = x\log x$) provides a closed-form estimator of the first ratio $\log(Z_1/Z_{1/2})$, and minimizing the reverse importance sampling loss ($\phi_1 = -\log x$) provides a closed-form estimator of the second ratio $\log(Z_{1/2}/Z_0)$. Combining these recovers the *bridge sampling estimator* as a special case [247]

$$\widehat{\log Z_1} = -\log \mathbb{E}_{p_1}\frac{f_{1/2}}{f_1} + \log \mathbb{E}_{p_0}\frac{f_{1/2}}{f_0} + \log Z_0 \ . \tag{5.4}$$

Alternatively, we can use these classification losses in reverse order : reverse importance sampling ($\phi_0 = -\log x$) for the first ratio, and importance sampling ($\phi_1 = x\log x$) for the second ratio, and recover the *umbrella sampling estimator* [246] also known as the *ratio sampling estimator* [250]

$$\widehat{\log Z_1} = \log \mathbb{E}_{p_{1/2}}\frac{f_1}{f_{1/2}} - \log \mathbb{E}_{p_{1/2}}\frac{f_0}{f_{1/2}} + \log Z_0 \ . \tag{5.5}$$

Another option yet, is to use the same classification loss for all ratios. With importance sampling ($\phi_k = x\log x,\ \forall k \in [\![0, K-1]\!]$), we recover the *annealed importance sampling* estimator [237–239]

$$\widehat{\log Z_1} = \sum_{k=1}^{K} \log \mathbb{E}_{\boldsymbol{x}\sim p_{k-1}}\left[\frac{f_k}{f_{k-1}}(\boldsymbol{x})\right] + \log Z_0 \ . \tag{5.6}$$

The family of annealed Bregman estimators is visibly large enough to include many existing estimators, obtained for different hyperparmeter choices. This raises the fundamental question of how these hyperparameters should be chosen, in particular in the challenging case where the *target and proposal have little overlap and the data is high dimensional*. To answer this question, we will study the estimation error produced by different hyperparameter choices.

5.5 . Statistical analysis of the hyperparameters

We consider a fixed data budget N and investigate how the remaining hyperparameters are best chosen for statistical efficiency. The starting point for the analysis is that as ABE estimates the normalization in log-space, the estimator is obtained by a sum of independent and asymptotically unbiased estimators [91] given in Eq. 5.1 and thus the mean squared errors written in table 5.1 are additive. (Recall, the independence of these estimators is because new samples are drawn for each estimation task.) Each individual error actually measures an overlap between two consecutive distributions along the path, and annealing integrates these overlaps.

Classification losses, ϕ_k Given the popularity of annealed importance sampling, we should first ask if the importance sampling loss is really an acceptable default. We recall an important limitation of importance sampling : its estimation error is notoriously sensitive to distribution tails [251]. Without annealing, it is infinite when the target p_1 has a heavier tail than the proposal p_0. When annealing with a geometric path, for example between two Gaussians with different covariances $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{Id})$ and $p_1 = \mathcal{N}(\mathbf{0}, 2\,\mathbf{Id})$, the geometric path produces Gaussians with increasing variances $\Sigma_t = (1 - t/2)^{-1}\,\mathbf{Id}$ and therefore increasing tails. Hence, the same tail mismatch holds along the path. Note that this concern is a realistic one for natural image data, as the target distribution over images is typically super-Gaussian [252] while the proposal is usually chosen as Gaussian.

This warrants a better choice for the loss : In the binary setup ($K = 1$), the NCE loss is optimal [247, 92] and its error can be orders of magnitude less than importance sampling [247]. This optimality result has been extended to a sequence of distributions $K > 1$ [253, eq. 16]. We further show that in the limit of a continuous path, the gap between annealed IS and annealed NCE is closed and we provide their estimation error :

Theorem 3 (Estimation error and the Fisher-Rao path length) *For a finite value of K, the optimal loss is NCE*

$$\mathrm{MSE}(p_0, p_1; q, K, N, \phi_{\mathrm{NCE}}) \leq \mathrm{MSE}(p_0, p_1; q, K, N, \phi), \qquad \forall q, K, N, \phi \ . \tag{5.7}$$

In the limit of $K \to \infty$ (such that $K^2/N \to 0$), NCE, IS, and revIS converge to the same estimation error, given by the Fisher-Rao path length from the proposal to the target

$$\mathrm{MSE}(p_0, p_1; q, K, N, \phi) = \frac{1}{N}\int_0^1 I(t)dt + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big), \forall \phi \in \{\phi_{\mathrm{NCE}}, \phi_{\mathrm{IS}}, \phi_{\mathrm{RevIS}}\} \tag{5.8}$$

where the Fisher-Rao metric $I(t) := \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}[(\frac{d}{dt}\log p_t(\boldsymbol{x}))^2]$ defined as the Fisher information over the path, using time t as the parameter.

This is proven in Appendix 5.8.2.Note that in this theorem, to take limits successively in $N \to \infty$ then in $K \to \infty$, we assume that N grows at least as fast as K^2. While the NCE estimator requires solving a (potentially non-convex) scalar optimization problem in Eq. 5.2 which itself requires samples from the target distribution and IS does not, this is the (possibly steep) computational price to pay for statistical optimality. In the following we will keep the optimal NCE loss and will indicate the dependency of the estimation error on ϕ_{NCE} with a subscript, instead. We highlight that our theorems in this paper apply to the MSE in the limit of $K \to \infty$: their results hold the same for the IS and RevIS losses by virtue of theorem 3. Just as in the binary case, while the estimator is computed with the *unnormalized* path of densities (Eq. 5.2), the estimation error depends on the *normalized* path of densities (Eq. 5.8).

Number of distributions, $K + 1$ It is known that estimating the normalization constant using plain importance sampling ($K = 1$) can produce a statistical error than is exponential in the distance between the target and the proposal [254]. We show that in the binary case, NCE also suffers from an estimation error that scales exponentially with the parameter-distance between the target and proposal dimension.

In the following, we consider a proposal p_0 and target p_1 that are in an exponential family with sufficient statistics $\boldsymbol{t}(\boldsymbol{x})$. Note that certain exponential families have universal approximation capabilities [255, 256]. The exponential family is defined as

$$p(\boldsymbol{x};\boldsymbol{\theta}) := \exp(\langle \boldsymbol{\theta}, \boldsymbol{t}(\boldsymbol{x})\rangle - \log Z(\boldsymbol{\theta})) \tag{5.9}$$

where $Z(\boldsymbol{\theta}) = \int \exp(\langle \boldsymbol{\theta}_1, \boldsymbol{t}(\boldsymbol{x})\rangle)$ is the partition function. We will consider that the (unnormalized) target density f_1 is what we call a *simply unnormalized model* defined as

$$f_1(\boldsymbol{x}) = \exp(\langle \boldsymbol{\theta}_1, \boldsymbol{t}(\boldsymbol{x})\rangle) \tag{5.10}$$

Note that in general, a pdf can be unnormalized in many ways : one can multiply an unnormalized density by any positive function of $\boldsymbol{\theta}$ and it will still be unnormalized. However, the simple and intuitive case defined above is what we base the analysis below on.

For exponential families, the log-normalization $\log Z(\boldsymbol{\theta})$ is a convex function ("log-sum-exp") of the parameter $\boldsymbol{\theta}$ [257], which implies $0 \preccurlyeq \nabla^2_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$. In our theorems we use the further assumptions of strong convexity with constant M, and/or smoothness with constant L (gradient is L-Liptschitz) :

$$\nabla^2_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) \succcurlyeq M\, \mathbf{Id} \tag{5.11}$$

$$\nabla^2_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) \preccurlyeq L\, \mathbf{Id} \tag{5.12}$$

For exponential families, the derivatives of the log partition function yield moments of the sufficient statistics, and the Hessian $\nabla^2_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) = \mathrm{Cov}_{\boldsymbol{x}\sim p}[\boldsymbol{t}(\boldsymbol{x})]$ is in fact the Fisher matrix. We can interpret our two assumptions : Eq. 5.11 can be seen as a form of "strong identifiability". Namely, positive-definiteness is required of the Fisher matrix, for the Maximum-Likelihood loss to have a unique minimum : we further assume a lower-bound on the smallest eigenvalue, which can be viewed as a strong identifiability condition. Eq. 5.12 can be interpreted as a bound on the second-order moments of the distribution $p(\boldsymbol{x};\boldsymbol{\theta})$, which is equivalent to the variance in every direction being bounded, which will be the case for parameters in a bounded domain $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. An example along with the proofs of the following Theorems 4 and 5, are provided in Appendix 5.8.2.

Theorem 4 (Exponential error of binary NCE) *Assume the proposal p_0 is from the normalized exponential family, while the (unnormalized) target f_1 is from the simply unnormalized exponential family (Eq. 5.10). The log-partition function $\log Z(\boldsymbol{\theta})$ is assumed to be strongly convex (Eq. 5.11).*
Then in the binary case $K = 1$, the estimation error of NCE is (at least) exponential in the parameter-distance between the proposal and the target :

$$\mathrm{MSE}_{\mathrm{NCE}}(p_0, p_1; q, K, N) \geq \frac{4}{N} \exp\left(\frac{1}{8} M \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2\right) - 1 + o\left(\frac{1}{N}\right), \qquad \text{when } K = 1 \tag{5.13}$$

where M is the strong convexity constant of $\log Z(\boldsymbol{\theta})$.

Annealing the importance sampling estimator (increasing K) was proposed in hope that we can trade the statistical cost in the dimension for a computational cost (number of classification tasks)

which is more acceptable. Yet, there is no definitive theory on the ability of annealing to reduce the statistical cost in a general setup [245, 248]. For both importance sampling and noise-contrastive estimation, we prove that annealing with the near-universally used geometric path brings down the estimation error, from exponential to polynomial in the parameter-distance between the proposal and target. Given that we expect $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|_2$ to scale as $\sqrt{D}$ with the dimension, using these paths effectively makes annealed estimation amenable to high-dimensional problems. This corroborates empirical [258] and theoretical [244] results which suggested in simple cases that annealing with an appropriate path can reduce the estimator error up to several orders of magnitude.

Theorem 5 (Polynomial error of annealed NCE with a geometric path) *Assume the proposal p_0 is from the normalized exponential family, while the (unnormalized) target f_1 is from the simply unnormalized exponential family (Eq. 5.10). The log-partition function $\log Z(\boldsymbol{\theta})$ is assumed to be strongly convex and smooth (Eq. 5.11, Eq. 5.12).*
Then in the annealing limit of a continuous path $K \to \infty$, the estimation error of annealed NCE with the geometric path is (at most) polynomial in the parameter-distance between the proposal and the target

$$\mathrm{MSE}_{\mathrm{NCE}}(p_0, p_1; q, K, N) \leq \frac{L^2}{MN}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right), \text{ when } q = 0 \tag{5.14}$$

where M and L are respectively the strong convexity and smoothness constants of $\log Z(\boldsymbol{\theta})$.

To our knowledge, this is the first result building on Gelman and Meng [244, Table 1] and Grosse et al. [245] which showcases the benefits of annealed estimation for a general target distribution.

We conclude that annealing with the near-universally used geometric path provably benefits noise-contrastive estimation, as well as importance sampling and reverse importance sampling, when the proposal and target distributions have little overlap.

Path parameter, q — geometric vs. arithmetic Despite the near-universal popularity of the geometric path ($q \to 0$), it is worth asking if there are other simple paths that are more optimal. Interpolating moments of exponential families was shown to outperform the geometric path by Grosse et al. [245], yet building such a path requires knowing the exponential family of the target. Other paths based on the arithmetic mean (and generalizations) of the target and proposal, were proposed in Masrani et al. [248], without a definitive theory of the estimation error.

Next, we analyze the error of the arithmetic path. We prove that the arithmetic path ($q = 1$) does *not* exhibit the same benefits as the geometric path : in general, its estimation error grows exponentially in the parameter-distance between the target and proposal distributions. However, in the case where an oracle gives us the normalization Z_1 to be used only in the construction of the path (we will discuss what this means in practice below), the arithmetic path can be reparameterized so as to bring down the estimation error to polynomial, even constant, in the parameter-distance. We start by the negative result.

Theorem 6 (Exponential error of annealed NCE with an arithmetic path) *Assume the proposal p_0 is from the normalized exponential family, while the (unnormalized) target f_1 is from the simply unnormalized exponential family (Eq. 5.10). The log-partition function $\log Z(\boldsymbol{\theta})$ is assumed to be strongly convex (Eq. 5.11).*

Consider the annealing limit of a continuous path $K \to \infty$ path and a far-away target with large enough $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\| > 0$. For estimating the log normalization of the (unnormalized) target density f_1, the estimation error of annealed NCE with the arithmetic path is (at least) exponential in the parameter-distance between the proposal and the target.

$$\mathrm{MSE}_{\mathrm{NCE}}(p_0, p_1; q, K, N) > \frac{C}{N} \exp\left(\frac{M}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2\right) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right), \text{ when } q = 1 \tag{5.15}$$

where C is constant defined in Appendix 5.8.2.

We suggest an intuitive explanation for this negative result. We begin with the observation that the estimation error (Eq. 5.8) depends on the *normalized* path of densities. Suppose the target model is rescaled by a constant 100, so that the new unnormalized target density is $f_1(\boldsymbol{x}) \times 100$ and its new normalization is $Z_1 \times 100$. Looking at table 5.2, this rescaling does not modify the geometric path of normalized densities, while it does the arithmetic path of normalized densities. Because the estimation error depends on path of normalized densities, this makes the arithmetic choice sensitive to target normalization, even more so as the parameter distance grows and the log-normalization with it, as a strongly convex function of it (Appendix, Eq. 5.94). This suggests making the arithmetic path of normalized distributions "robust" to the choice of Z_1. We will show this can be achieved by re-parameterizing the path in terms of Z_1.

We next prove that certain reparameterizations can bring down the error to a polynomial and even constant function of the parameter-distance between the target and proposal. The following theorems may seem purely theoretical, as if necessitating an oracle for Z_1, but they will actually lead to an efficient estimation algorithm later.

Theorem 7 (Polynomial error of annealed NCE with an arithmetic path and oracle) *Assume the same as in Theorem 6, replacing the strong convexity of the log-partition by smoothness (Eq. 5.12). Additionally, suppose an oracle gives the normalization constant Z_1 to be used only in the reparameterization of the arithmetic path with $t \to \frac{t}{t+Z_1(1-t)}$ (see Table 5.2). This brings down the estimation error of annealed NCE to (at most) polynomial in the parameter-distance*

$$\mathrm{MSE}_{\mathrm{NCE}}(p_0, p_1; q, K, N) \leq \frac{1}{N}(2 + L\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right), \qquad \text{when } q = 1 \tag{5.16}$$

where L is the smoothness constant of $\log Z(\boldsymbol{\theta})$.

In fact, supposing we have (oracle) access to the normalizing constant Z_1, the arithmetic path can even be reparameterized such that it is the optimal path in a certain limit. We next prove such optimality in the limits of a continuous path $K \to \infty$ and "far-away" target and proposal [244] :

Theorem 8 (Constant error of annealed NCE with an arithmetic path and oracle) *Consider the limits of a continuous annealing path $K \to \infty$, and of a target distribution whose overlap with the proposal goes to zero. Namely, consider the quantities :*

$$\epsilon^{'}(\boldsymbol{x}) := \sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})} \qquad\qquad \epsilon \;\; := \int_{\mathbb{R}^D} \epsilon^{'}(\boldsymbol{x}) d\boldsymbol{x} \tag{5.17}$$

Table 5.2 – Geometric and arithmetic paths, defined in the space of unnormalized densities (second column); “oracle" and “oracle-trig" are reparameterizations of the arithmetic path which depend on the true normalization Z_1. The corresponding normalized densities (third column) produce an estimation error (fourth column) which we quantify.

Path name	Unnormalized density	Normalized density	Error
Geometric	$f_t(\boldsymbol{x}) = p_0(\boldsymbol{x})^{1-t} f_1(\boldsymbol{x})^t$	$p_t(\boldsymbol{x}) \propto p_0(\boldsymbol{x})^{1-t} p_1(\boldsymbol{x})^t$	poly
Arithmetic	$f_t(\boldsymbol{x}) = (1-w_t)p_0(\boldsymbol{x}) + w_t f_1(\boldsymbol{x})$	$p_t(\boldsymbol{x}) = (1-\tilde{w}_t)p_0(\boldsymbol{x}) + \tilde{w}_t p_1(\boldsymbol{x})$	
vanilla	$w_t = t$	$\tilde{w}_t = \frac{tZ_1}{(1-t)+tZ_1}$	exp
oracle	$w_t = \frac{t}{t+Z_1(1-t)}$	$\tilde{w}_t = t$	poly
oracle-trig	$w_t = \frac{\sin^2\left(\frac{\pi t}{2}\right)}{\sin^2\left(\frac{\pi t}{2}\right)+Z_1\cos^2\left(\frac{\pi t}{2}\right)}$	$\tilde{w}_t = \sin^2\left(\frac{\pi t}{2}\right)$	const

$$\epsilon^{''}(\boldsymbol{x}) := \frac{\epsilon^{'}(\boldsymbol{x}) - \epsilon \sin(\pi t) p_t^{\text{oracle}}(\boldsymbol{x})}{p_t^{\text{oracle-trig}}(\boldsymbol{x})} \quad \epsilon^{'''} := \int_{\mathbb{R}^D} \int_0^1 \epsilon^{''}(\boldsymbol{x}) \Big(1 + \frac{\big(\partial_t p_t^{\text{oracle-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{oracle-trig}}(\boldsymbol{x})}\Big) dt d\boldsymbol{x} \ . \tag{5.18}$$

Assume $\sup_{\boldsymbol{x}\in\mathbb{R}^D} \epsilon'(x) \to 0, \sup_{\boldsymbol{x}\in\mathbb{R}^D} \epsilon''(x) \to 0, \epsilon \to 0, \epsilon''' \to 0$*, and consider the distributions* $p_t^{\text{arith-trig}}$ *and* p_t^{oracle} *as defined in Table 5.2.*

Then, the optimal annealing path convergences pointwise to an arithmetic path reparameterized trigonometrically with $t \to \frac{t}{\sin^2(\frac{\pi t}{2})+Z_1(1-\sin^2(\frac{\pi t}{2}))}$ *and the estimation error tends to the optimal estimation error (which is constant with respect to the parameter-distance) :*

$$\text{MSE}_{\text{NCE}}(p_0, p_1; q, K, N) = \frac{1}{N}\pi^2 + O\Big(\frac{\epsilon + \epsilon^{'''}}{N}\Big) + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \quad \text{when } q = 1 \ . \tag{5.19}$$

By way of remarks, we note that the assumptions that the quantities $\sup \epsilon'(x), \sup \epsilon''(x), \epsilon, \epsilon'''$ go to 0 are mutually incomparable (i.e. none of them implies the others). For example, $\sup_{\boldsymbol{x}\in\mathbb{R}^D} \epsilon'(\boldsymbol{x})$ and ϵ going 0 require that the function $\sqrt{p_0(x)p_1(x)}$ goes to 0 in the L_∞ and L_1 sense, respectively — and these two norms are not equivalent in the Lebesgue measure.

Two-step estimation Thus, we see that, perhaps unsurprisingly, the "optimal" mixture weights in the space of unnormalized densities depends on the true Z_1 : however, this dependency is simple. We propose a two-step estimation method : first, Z_1 is pre-estimated, for example using the geometric path; second, the estimate of Z_1 is plugged into the "oracle" or "oracle-trig" weight of the arithmetic path (table 5.2, column 2), and which is used to obtain a second estimation of Z_1. Note that pre-estimating a problematic (hyper)parameter, here Z_1, has proved beneficial to reduce the estimation error of NCE in a related context [259].

5.6 . Numerical results

We now present numerical evidence for our theory and validate our two-step estimators. Importantly, we do *not* claim to achieve state of the art in terms of practical evaluation of the normalization

constants; our goal is to support our theoretical analysis. We follow the evaluation methods of importance sampling literature [245] and evaluate our methods on synthetic Gaussians. This setup is specially convenient for validating our theory : the optimal estimation error can conveniently be computed in closed-form, so too can the geometric and arithmetic paths which avoids a sampling error from MCMC algorithms. These derivations are included in the Appendix 5.8.2. We specifically consider the high-dimensional setting, where the computation of the determinant of a high-dimensional (covariance) matrix which appears in the normalization of a Gaussian, can in fact be challenging [260].

Numerical Methods The proposal distribution is always a standard Gaussian, while the target differs by the second moment : $p_1 = \mathcal{N}(\mathbf{0}, 2\,\mathbf{Id})$ in Figure 5.2, $p_1 = \mathcal{N}(\mathbf{0}, 0.25\,\mathbf{Id})$ in Figure 5.3b and $p_1 = \mathcal{N}(\mathbf{0}, \sigma^2\,\mathbf{Id})$ in Figure 5.3a, where the target variance decreases as $\sigma(i) = i^{-1}$ so that the (natural) parameter distance grows linearly [257, Part II-4]. We use a sample size of $N = 50000$ points, and, unless otherwise mentioned, $K + 1 = 10$ distributions from the annealing paths and the dimensionality is 50. To compute an estimator of the normalization constant using the non-convex NCE loss, we used a non-linear conjugate gradient scheme implemented in Scipy [229]. We chose the conjugate-gradient algorithm as it is deterministic (no residual variance like in SGD). The empirical Mean-Squared Error was computed over 100 random seeds, parallelized over 100 CPUs. The longest experiment was for Figure 5.3b and took 7 wall-clock hours to complete. For the two-step estimators ("two-step" and "two-step (trig)"), a pre-estimate of the normalization was first computed using the geometric path with 10 distributions. Then, this estimate was used to to re-parameterize an arithmetic path with 10 distributions which produced the second estimate.

Results Figure 5.2 numerically supports the optimality of the NCE loss for a finite K (here, $K = 2$ so three distributions are used) proven in Theorem 3. Figure 5.3 validates our main results for annealing paths. It shows how the estimation error scales with the proposal and target distributions growing apart, either with the parameter-distance in Figure 5.3a or with the dimensionality in Figure 5.3b. Using no annealing path ($K = 1$) produces an estimation error which grows linearly in log space; this numerically supports the exponential growth predicted by Theorem 4. Meanwhile, annealing ($K \to \infty$) sets the estimation error on different trends, depending on the choice of path. Choosing the geometric path brings the growth down to sub-exponential, as predicted by Theorem 5, while choosing the (basic) arithmetic path does not as in Theorem 6. To alleviate this, our two-step estimation methods consist in reparameterizing the arithmetic path so that it actually does bring down the estimation error. In fact, our two-step estimators in table 5.2 empirically approach the optimal estimation error in Figure 5.3. While this requires more computation, it has the appeal of making the estimation error *constant* with respect to the parameter-distance between the target and proposal distributions. Practically, this means that in Figure 5.3a, regular Noise-Contrastive Estimation (black, full line) fails when the parameter-distance between the target and proposal distributions is higher than 20, while our two-step estimators remain optimal.

We next explain interesting observations in Figure 5.3 which are actually coherent with our theory. First, in Figure 5.3a, the "two-step (trig)" estimator is only optimal when the parameter-distance between the target and proposal distributions is larger than 10. This is because the optimality of this two-step estimator was derived in Theorem 8 conditionally on non-overlapping distributions, here

achieved by a large parameter-distance. Second, in both Figures 5.3a and 5.3b, the "two-step" estimator empirically achieves the optimal estimation error that was predicted for the "two-step (trig)" estimator. This suggests our polynomial upper bound from Theorem 7 may be loose in certain cases. This further explains why, in Figure 5.3a, the arithmetic path is near-optimal for a single value of the parameter-distance. At this value of 20, the partition function happens to be equal to one $Z(\theta_1) = 1$, so that the arithmetic path is effectively the same as the "two-step" estimator.

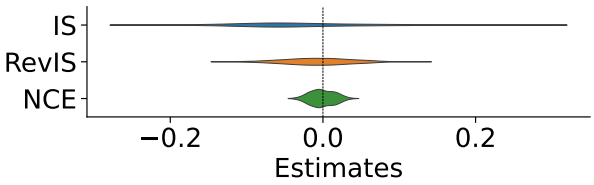
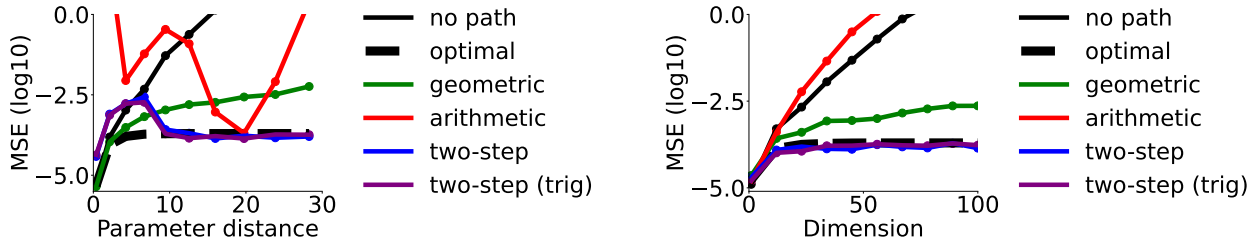


Figure 5.2 – Optimality of the NCE loss, using the geometric path with $K = 2$. NCE has the smallest deviation from zero, the true value of the log normalizing constant.



(a) MSE against parameter distance.

(b) MSE against dimensionality.

Figure 5.3 – Estimation error as the target and proposal distributions grow apart. Without annealing, the error is exponential in the parameter distance (diagonal in log-scale). Annealing with the geometric path and our two-step methods brings down the error to slower growth, as predicted by our theorems.

5.7 . Discussion

Previous work has mainly focused on annealed importance sampling [245, 261], which is a special case of our annealed Bregman estimator. They have evaluated the merits of different paths empirically, using an approximation of the estimation error called Effective Sample Size (ESS) and the consistency-gap. In our analysis, we consider consistent estimators and derive and optimize the exact estimation error of the optimal Noise-Contrastive Estimation. Liu et al. [251] considered the NCE estimate for Z (not $\log Z$) with the name "discriminance sampling", and annealed the estimator using an extended state-space construction similar to Neal [237]. Their analysis of the estimation error is relevant but does not deal with hyperparameters other than the classification loss.

We made the common assumption of perfect sampling [244, 245, 262, 263] in order to make the estimation error tractable and obtain practical guidelines to reduce it. We note however, that this leaves a gap to bridge with a practical setup where the sampling error cannot be ignored; in fact, annealed importance sampling [237] was originally proposed such that the samples can be obtained from a Markov Chain that has not converged. In this original formulation, AIS is a special case of a larger framework called Sequential Monte Carlo (SMC) [264] in which the path of distributions is implicitly defined (by Markov transitions), sometimes even "on the go" [263]. Yet even within that theory, it seems that analyzing the estimation error for an inexplicit path of distributions is challenging [265, Eq. 38]. In particular, samples from MCMC will typically follow marginal distributions that are not analytically tractable, thus the stronger assumption of "perfect sampling" is often used to make estimation error depend explicitly on the path of distributions [266, Eq. 3.2]. Even then, results on how the estimation error scales with dimensionality are heuristic [237] or limited by assumptions such as an essentially log-concave path of distributions or a factorial target distribution [266].

It might also be argued that the limit of almost no overlap between proposal and target, which we use a lot, is unrealistic. To see why it can be realistic, consider the case of natural image data. A typical proposal in high dimensions is a Gaussian, since it is both tractable and principled : it is the distribution which spreads the most mass among those sharing the same (finite) mean and variance as the target [147, Section 4.1.4]. However, there is almost no overlap between Gaussian data and natural images, which is seen in the fact that a human observer can effortlessly discriminate between the two.

More generally, note that a number of methods based on "annealing" were developed to deal with sampling issues. In fact, the path costs for two such methods, parallel tempering [267, eq. 17] and tempered transitions [268, eq. 18], are equal (or upper bounded) by a a sum of f-divergences which in the limit of a continuous path is the same cost function as in our Theorem 3. This suggests our results may be applicable to more practical methods in the literature.

Conclusion We defined a class of estimators of the normalization constant, annealed Bregman estimation (ABE), which relies on a sampling phase from a path of distributions, and an estimation phase where these samples are used to estimate the log-normalization of the target distribution. Our results suggest a number of simple recommendations regarding hyperparameter choices in annealing. First, if the path has very few intermediate distributions, it is better to choose NCE due to its statistical optimality (Theorem 3). If however, the path has many intermediate distributions and approaches the annealing limit, then IS enjoys the same statistical optimality as NCE but has the advantage of its computational simplicity. Annealing can always provide substantial benefits (Theorem 4). Moreover, if we have a reasonable a priori estimate of Z_1, the arithmetic path achieves very low error (Theorem 7) — sometimes even approaching optimality (Theorem 8). On the other hand, even absent an initial estimate of Z_1, the geometric path can exponentially reduce the estimation error compared with no annealing (Theorems 4 and 5).

Acknowledgemenets This work was supported by the French ANR-20-CHIA-0016. Aapo Hyvärinen was supported by funding from the Academy of Finland and a Fellowship from CIFAR. Andrej Risteski was supported in part by NSF awards IIS-2211907, CCF-2238523, and an Amazon Research

Award.

5.8 . Supplemental material

In the following, we will study the estimation error of of annealed Bregman estimation (ABE) in two important setups : the log-normalization is computed using two distributions ($K = 1$), the proposal and the target, or else using a path of distributions ($K \to \infty$).

The anonymized code used for the experiments is available at `https://github.com/l-omar-chehab/annealing-normalizing-constants`.

5.8.1 . No annealing, $K = 1$

We use [92, Eq.21] for the estimation error of any suitably parameterized [3] classifier $F(\boldsymbol{x}; \boldsymbol{\beta})$ between two distributions p_1 and p_0. The estimation error is measured by the asymptotic Mean-Squared Error (MSE)

$$\mathrm{MSE}_{\hat{\boldsymbol{\beta}}}(p_n, \nu, \phi, N) = \frac{\nu + 1}{N} \mathrm{tr}(\boldsymbol{\Sigma}) + o\Big(\frac{1}{N}\Big) \tag{5.20}$$

which depends on the sample sizes $N = N_1 + N_0$, their ratio $\nu = N_1/N_0$, the Bregman classification loss indexed by the convex function $\phi(x)$, and the asymptotic variance matrix

$$\boldsymbol{\Sigma} = \boldsymbol{I}_w^{-1}\big(\boldsymbol{I}_v - (1 + \frac{1}{\nu})\boldsymbol{m}_w \boldsymbol{m}_w^\top\big)\boldsymbol{I}_w^{-1} \quad . \tag{5.21}$$

We suppose the standard technical conditions of van der Vaart [54, Th. 5.23] apply so that the remainder term $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|^2$ can indeed be written independently of the parameterization, as $o(N^{-1})$. Here, $\boldsymbol{m}_w(\boldsymbol{\beta}^*)$, $\boldsymbol{I}_w(\boldsymbol{\beta}^*)$ and $\boldsymbol{I}_v(\boldsymbol{\beta}^*)$ are the reweighted mean and covariances of the paramete-gradient of the classifier, also known as the “relative" Fisher score $\nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)$,

$$\boldsymbol{m}_w(\boldsymbol{\beta}^*) = \mathbb{E}_{\boldsymbol{x}\sim p_d}\big[w(\boldsymbol{x})\nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)\big] \tag{5.22}$$

$$\boldsymbol{I}_w(\boldsymbol{\beta}^*) = \mathbb{E}_{\boldsymbol{x}\sim p_d}\big[w(\boldsymbol{x})\nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)\, \nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)^\top\big] \tag{5.23}$$

$$\boldsymbol{I}_v(\boldsymbol{\beta}^*) = \mathbb{E}_{\boldsymbol{x}\sim p_d}\big[v(\boldsymbol{x})\nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)\, \nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*)^\top\big] \tag{5.24}$$

where the reweighting of data points is by $w(\boldsymbol{x}) := \frac{p_1}{\nu p_0}(\boldsymbol{x})\phi''\big(\frac{p_1}{\nu p_0}(\boldsymbol{x})\big)$ and by $v(\boldsymbol{x}) = w(\boldsymbol{x})^2 \frac{\nu p_0(\boldsymbol{x}) + p_1(\boldsymbol{x})}{\nu p_0(\boldsymbol{x})}$, which are all evaluated at the true parameter value $\boldsymbol{\beta}^*$.

Scalar parameterization We now consider a specific parameterization of the classifier :

$$F(\boldsymbol{x}; \beta) = \log\Big(\frac{f_1(\boldsymbol{x})}{\nu f_0(\boldsymbol{x})}\Big) - \beta \tag{5.25}$$

where the optimal parameter is the log-ratio of normalizations $\beta^* = \log(Z_1/Z_0)$. Consequently, we have $\nabla_{\boldsymbol{\beta}} F(\boldsymbol{x}; \beta^*) = -1$ and plugging this into the above quantities yields

$$\mathrm{MSE} = \frac{1+\nu}{T}\Big(\frac{\mathbb{E}_{\boldsymbol{x}\sim p_1}\big[w^2(\boldsymbol{x})\frac{\nu p_0(\boldsymbol{x}) + p_1(\boldsymbol{x})}{\nu p_0(\boldsymbol{x})}\big]}{\mathbb{E}_{\boldsymbol{x}\sim p_1}[w(\boldsymbol{x})]^2} - (1 + \frac{1}{\nu})\Big) + o\Big(\frac{1}{N}\Big)$$

3. technically, the formula was derived in [63, 11] assuming the classifier was parameterized as $F(\boldsymbol{x}; \boldsymbol{\beta}) = \log p_1(\boldsymbol{x}; \boldsymbol{\beta})/\nu p_0(\boldsymbol{x})$ but the proof seems to generalize to any well-defined parameterization $F(\boldsymbol{x}; \boldsymbol{\beta})$.

which matches the formula found in [247, Eq 3.2]. For different choices of the Bregman classification loss, the estimation error is written using a divergence between the two distributions

Name	Loss identified by $\phi(\boldsymbol{x})$	Estimator	MSE up to $o(N^{-1})$
IS	$x \log x$	$\log \mathbb{E}_{p_0} \frac{f_1}{f_0}$	$\frac{1+\nu}{\nu N} \mathcal{D}_{\chi^2}(p_1, p_0)$
RevIS	$-\log x$	$-\log \mathbb{E}_{p_1} \frac{f_0}{f_1}$	$\frac{1+\nu}{N} \mathcal{D}_{\chi^2}(p_0, p_1)$
NCE	$x \log x - (1+x) \log(\frac{1+x}{2})$	implicit	$\frac{(1+\nu)^2}{\nu N} \frac{\mathcal{D}_{\mathrm{HM}}(p_1,p_0)}{1-\mathcal{D}_{\mathrm{HM}}(p_1,p_0)}$
IS-RevIS	$(1-\sqrt{x})^2$	$\log \mathbb{E}_{p_0} \frac{f_1}{f_0} - \log \mathbb{E}_{p_1} \frac{f_0}{f_1}$	$\frac{(1+\nu)^2}{\nu N} \frac{1-(1-\mathcal{D}_{H^2}(p_d,p_n))^2}{(1-\mathcal{D}_{H^2}(p_d,p_n))^2}$

where

$\mathcal{D}_{\chi^2}(p_1, p_0) := \left(\int \frac{p_1^2}{p_0}\right) - 1$ is the chi-squared divergence
$D_{H^2}(p_1, p_0) := 1 - \left(\int \sqrt{p_1 p_0}\right) \in [0, 1]$ is the squared Hellinger distance
$\mathcal{D}_{\mathrm{HM}}(p_1, p_0) := 1 - \int \left(\pi p_1^{-1} + (1-\pi) p_0^{-1}\right)^{-1} = 1 - \frac{1}{\pi} \mathbb{E}_{p_1} \frac{\pi p_0}{(1-\pi)p_1 + \pi p_0} \in [0, 1]$
is the harmonic divergence with weight $\pi \in [0, 1]$.
Here, the weight $\pi = P(Y = 0) = \frac{T_n}{T} = \frac{\nu}{1+\nu}$.

Proof of Theorem 4 *Exponential error of binary NCE*

In the following, we will drop the remainder term in $o(N^{-1})$ given that no other limits are taken and that we will study the dominant term only. The estimation error of binary NCE is expressed in terms of the harmonic divergence

$$\mathrm{MSE} = \frac{4}{N} \frac{\mathcal{D}_{\mathrm{HM}}(p_1, p_0)}{1 - \mathcal{D}_{\mathrm{HM}}(p_1, p_0)} \tag{5.26}$$

which is intractable for general exponential families. Instead, we can lower-bound the estimation error. To do so, we lower-bound the harmonic divergence using the inequality of means (harmonic vs. geometric)

$$\mathcal{D}_{\mathrm{HM}}(p_1, p_0) = 1 - \int \frac{2 p_0 p_1}{p_0 + p_1} \geq 1 - \int \sqrt{p_0 p_1} = \mathcal{D}_{H^2}(p_0, p_1) \tag{5.27}$$

and therefore

$$\mathrm{MSE}_{\mathrm{LB}} = \frac{4}{N} \frac{\mathcal{D}_{H^2}(p_1, p_0)}{1 - \mathcal{D}_{H^2}(p_1, p_0)} \quad . \tag{5.28}$$

This lower bound is expressed in terms of the squared Hellinger distance, that is tractable for exponential families :

$$\mathcal{D}_{H^2}(p_1, p_0) := 1 - \int_{\boldsymbol{x} \in \mathbb{R}^D} \sqrt{p_1 p_0} d\boldsymbol{x} \tag{5.29}$$

$$= 1 - \int_{\boldsymbol{x} \in \mathbb{R}^D} \frac{1}{Z(\boldsymbol{\theta}_1)^{\frac{1}{2}} Z(\boldsymbol{\theta}_0)^{\frac{1}{2}}} \exp\left(\frac{1}{2}(\boldsymbol{\theta}_1 + \boldsymbol{\theta}_0)^\top \boldsymbol{t}(\boldsymbol{x})\right) d\boldsymbol{x} \tag{5.30}$$

$$= 1 - \frac{Z(\frac{1}{2}\boldsymbol{\theta}_1 + \frac{1}{2}\boldsymbol{\theta}_0)}{Z(\boldsymbol{\theta}_1)^{\frac{1}{2}} Z(\boldsymbol{\theta}_0)^{\frac{1}{2}}} \tag{5.31}$$

$$= 1 - \exp\left(\log Z\left(\frac{1}{2}\boldsymbol{\theta}_1 + \frac{1}{2}\boldsymbol{\theta}_0\right) - \frac{1}{2}\log Z(\boldsymbol{\theta}_1) - \frac{1}{2}\log Z(\boldsymbol{\theta}_0)\right) \ . \tag{5.32}$$

We now wish to lower bound $\mathrm{MSE}_{\mathrm{LB}}$, and therefore $\mathcal{D}_{H^2}(p_1, p_0)$, by an expression which is exponential in the parameter distance $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|$. To do so, we note that for exponential families, the log-normalization is convex in the parameters. Here, we further assume strong convexity, so that

$$\log Z\left(\frac{1}{2}\boldsymbol{\theta}_1 + \frac{1}{2}\boldsymbol{\theta}_0\right) \leq \frac{1}{2}\log Z(\boldsymbol{\theta}_1) + \frac{1}{2}\log Z(\boldsymbol{\theta}_0) - \frac{1}{8}M\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 \tag{5.33}$$

where M is the strong convexity constant. Plugging this back into the squared Hellinger distance, we obtain

$$\mathcal{D}_{H^2}(p_1, p_0) \geq 1 - \exp\left(-\frac{1}{8}M\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2\right) \tag{5.34}$$

so that the MSE

$$\mathrm{MSE} \geq \frac{4}{N}\frac{\mathcal{D}_{H^2}(p_1, p_0)}{1 - \mathcal{D}_{H^2}(p_1, p_0)} \geq \frac{4}{N}\exp\left(\frac{1}{8}M\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2\right) - 1 \tag{5.35}$$

grows exponentially with the euclidean distance between the parameters.

5.8.2 . Annealing limit, $K \to \infty$

We now consider annealing paths $(p_t)_{t\in[0,1]}$ that interpolate between between the proposal p_0 and the target p_1.

Estimation error

We first show the optimality of the NCE loss within the family of Annealed Bregman Estimators, in the sense that it produces the smallest estimation error. We then study the estimation error of different annealed Bregman estimators in the annealing limit of a continuous path ($K \to \infty$).

Proof of Theorem 3 *Optimality of the NCE loss and the estimation error in the annealing limit $K \to \infty$*

- Optimality of the NCE loss
  Because the annealed Bregman estimator is built by adding independent estimators

$$\widehat{\log Z_1} = \sum_{k=0}^{K-1} \log \widehat{\left(\frac{Z_{(k+1)/K}}{Z_{k/K}}\right)} + \log Z_0 \ . \tag{5.36}$$

  the total Mean Squared Error (MSE) is the sum of each MSEs for each estimator (indexed by $k \in [\![0, K-1]\!]$)

$$\mathrm{MSE}((\phi_k)_{k\in[\![0,K]\!]}) = \sum_{k=0}^{K-1} \mathrm{MSE}_k(\phi_k) \tag{5.37}$$

where we highlighted the dependency on the classification losses identified by $(\phi_k)_{k\in[\![0,K]\!]}$. The MSEs follow Eq. 5.26. It was shown by Meng and Wong [247] that for any of these MSEs, the optimal loss is identified by $\phi_k(x) = x\log x - (1+x)\log(\frac{1+x}{2})$ and is in fact the NCE loss [92]. Thus the sum of MSEs is minimized for the same loss.

- Annealed Noise-Contrastive Estimation (NCE)
  We are interested in the estimation error (asymptotic MSE) obtained for the NCE loss. Based off table 5.1, it is written as

$$\mathrm{MSE} = \sum_{k=0}^{K-1}\left(\frac{4K}{N}\frac{\mathcal{D}_{\mathrm{HM}}(p_{k/K}, p_{(k+1)/K})}{1-\mathcal{D}_{\mathrm{HM}}(p_{k/K}, p_{(k+1)/K})} + o\left(\frac{K}{N}\right)\right) \tag{5.38}$$

$$= \frac{4K}{N}\sum_{k=0}^{K-1}\frac{\mathcal{D}_{\mathrm{HM}}(p_{k/K}, p_{(k+1)/K})}{1-\mathcal{D}_{\mathrm{HM}}(p_{k/K}, p_{(k+1)/K})} + o\left(\frac{K^2}{N}\right) \ . \tag{5.39}$$

where a sample budget of N/K is used for each estimator that is summed. The estimation error of balanced ($\nu = 1$) NCE-JS between two "close" distributions p_t and p_{t+h}, is

$$\mathrm{MSE}(p_t, p_{t+h}) \propto \frac{\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h})}{1-\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h})} \tag{5.40}$$

up to the remainder term. The estimation error can be simplified using a Taylor expansion. To do so, we recall that D_{HM} is an f-divergence generated by $\phi(x) = 1 - \frac{x}{\pi+(1-\pi)x}$ [269, 270] ($\pi = \frac{1}{2}$ here) and its expansion is therefore [271, Eq.7.64]

$$\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h}) = \frac{1}{2}h^2\nabla_t^2\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h}) + o(h^2) \tag{5.41}$$

$$= \frac{1}{2}\phi''(1)h^2 I(t) + o(h^2) = \frac{1}{4}h^2 I(t) + o(h^2) \ . \tag{5.42}$$

It follows that

$$\frac{\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h})}{1-\mathcal{D}_{\mathrm{HM}}(p_t, p_{t+h})} = \frac{1}{4}I(t)h^2 + o(h^2) \ . \tag{5.43}$$

Summing these estimation errors along the path of distributions with $h = 1/K$,

$$\mathrm{MSE} = \frac{4K}{N}\sum_{k=0}^{K-1}\left(\frac{1}{4}I(t)\frac{1}{K^2} + o\left(\frac{1}{K^2}\right)\right) + o\left(\frac{K^2}{N}\right) \tag{5.44}$$

$$= \left(\frac{1}{NK}\sum_{k=0}^{K-1} I(t)\right) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \tag{5.45}$$

$$= \frac{1}{N}\int_0^1 I(t)dt + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \quad . \tag{5.46}$$

In the case of a parametric path $p(\boldsymbol{x}|\boldsymbol{\theta}(t))_{t\in[0,1]}$, the proof is the same. Simply, the second-order term in the Taylor expansion of Eq. 5.42 is computed using the chain rule

$$\nabla_t^2\mathrm{MSE}(p_{\boldsymbol{\theta}(t)}, p_{\boldsymbol{\theta}(t+h)}) \tag{5.47}$$

$$= \dot{\boldsymbol{\theta}}(t)^\top \nabla^2_{\boldsymbol{\theta}} \mathrm{MSE}(p_{\boldsymbol{\theta}(t)}, p_{\boldsymbol{\theta}(t+h)}) \dot{\boldsymbol{\theta}}(t) + \ddot{\boldsymbol{\theta}}(t)^\top \nabla_{\boldsymbol{\theta}} \mathrm{MSE}(p_{\boldsymbol{\theta}(t)}, p_{\boldsymbol{\theta}(t+h)}) \tag{5.48}$$

$$= \dot{\boldsymbol{\theta}}(t)^\top \nabla^2_{\boldsymbol{\theta}} \mathrm{MSE}(p_{\boldsymbol{\theta}(t)}, p_{\boldsymbol{\theta}(t+h)}) \dot{\boldsymbol{\theta}}(t)^\top + 0 \tag{5.49}$$

$$= \dot{\boldsymbol{\theta}}(t)^\top \boldsymbol{I}(\theta(t)) \dot{\boldsymbol{\theta}}(t) \tag{5.50}$$

- Annealed importance sampling (IS)
  Similarly, for the choice of the importance sampling base estimator,

$$\mathrm{MSE} = \sum_{k=0}^{K-1} \left(\frac{2K}{N} \mathcal{D}_{\chi^2}(p_{(k+1)/K}, p_{k/K}) + o\left(\frac{K}{N}\right) \right) \tag{5.51}$$

$$= \frac{2K}{N} \sum_{k=0}^{K-1} \mathcal{D}_{\chi^2}(p_{(k+1)/K}, p_{k/K}) + o\left(\frac{K^2}{N}\right) \tag{5.52}$$

$$= \frac{2K}{N} \sum_{k=0}^{K-1} \mathcal{D}_{\mathrm{rev}\chi^2}(p_{k/K}, p_{(k+1)/K}) + o\left(\frac{K^2}{N}\right) \tag{5.53}$$

$$= \frac{2K}{N} \sum_{k=0}^{K-1} \left(\frac{1}{2}\phi''(1) I(t) \frac{1}{K^2} + o\left(\frac{1}{K^2}\right) \right) + o\left(\frac{K^2}{N}\right) \tag{5.54}$$

$$= \frac{1}{NK} \sum_{k=0}^{K-1} \phi''(1) I(t) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \tag{5.55}$$

$$= \frac{1}{N} \int_0^1 I(t) dt + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \ . \tag{5.56}$$

  given that $\phi(x) = -\log(x)$ and therefore $\phi''(1) = 1$ for the reverse χ^2 divergence.
- Annealed reverse importance sampling (RevIS)
  Similarly, for the choice of the reverse importance sampling base estimator,

$$\mathrm{MSE} = \sum_{k=0}^{K-1} \left(\frac{2K}{N} \mathcal{D}_{\chi^2}(p_{k/K}, p_{(k+1)/K}) + o\left(\frac{K}{N}\right) \right) \tag{5.57}$$

$$= \frac{2K}{N} \sum_{k=0}^{K-1} \left(\frac{1}{2}\phi''(1) I(t) \frac{1}{K^2} + o\left(\frac{1}{K^2}\right) \right) + o\left(\frac{K^2}{N}\right) \tag{5.58}$$

$$= \frac{1}{NK} \sum_{k=0}^{K-1} I(t) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) = \frac{1}{N} \int_0^1 I(t) dt + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \ . \tag{5.59}$$

  given that $\phi(x) = x\log(x)$ and therefore $\phi''(1) = 1$ for the χ^2 divergence.

Examples of paths

Geometric path The geometric path is defined in the space of unnormalized densities by

$$f_t(\boldsymbol{x}) := p_0(\boldsymbol{x})^{1-t} f_1(\boldsymbol{x})^t = p_0(\boldsymbol{x})^{1-t} p_1(\boldsymbol{x})^t Z_1^t \propto p_0(\boldsymbol{x})^{1-t} p_1(\boldsymbol{x})^t \tag{5.60}$$

so in the space of normalized densities, the path is

$$p_t := \frac{p_0(\boldsymbol{x})^{1-t} p_1(\boldsymbol{x})^t}{Z_t} \tag{5.61}$$

where the normalization is

$$Z_t := \int_{\boldsymbol{x}\in\mathbb{R}^d} p_0(\boldsymbol{x})^{1-t} p_1(\boldsymbol{x})^t d\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}\sim p_1}\left[\left(\frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}\right)^t\right] = \mathbb{E}_{\boldsymbol{x}\sim p_0}\left[\left(\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\right)^{1-t}\right] \ . \tag{5.62}$$

Arithmetic path The arithmetic path is defined in the space of unnormalized densities by

$$f_t(\boldsymbol{x}) := (1-t)p_0(\boldsymbol{x}) + t f_1(\boldsymbol{x}) = (1-t)p_0 + tZ_1 p_1 \tag{5.63}$$

$$\propto \frac{(1-t)}{(1-t)+tZ_1} p_0 + \frac{tZ_1}{(1-t)+tZ_1} p_1 \tag{5.64}$$

so in the space of normalized densities, the path is actually a mixture between the target and the proposal, where the weight of the mixture is a nonlinear function of the target normalization

$$p_t := (1-\tilde{w}_t)p_0 + \tilde{w}_t p_1, \quad \tilde{w}_t = \frac{tZ_1}{(1-t)+tZ_1} \ . \tag{5.65}$$

Optimal path We know (*e.g.* from Gelman and Meng [244, Eq. 49]) that the optimal path is

$$p_t(\boldsymbol{x}) = \left(a(t)\sqrt{p_0(\boldsymbol{x})} + b(t)\sqrt{p_1(\boldsymbol{x})}\right)^2 \tag{5.66}$$

where the coefficients $a(t)$ and $b(t)$

$$a(t) = \frac{\cos((2t-1)\alpha_H)}{2\cos(\alpha_H)} - \frac{\sin((2t-1)\alpha_H)}{2\sin(\alpha_H)} \tag{5.67}$$

$$b(t) = \frac{\cos((2t-1)\alpha_H)}{2\cos(\alpha_H)} + \frac{\sin((2t-1)\alpha_H)}{2\sin(\alpha_H)} \tag{5.68}$$

are simple functions of the squared Hellinger distance $\mathcal{D}_{H^2}(p_0, p_1)$ between the proposal and the target [4]

$$\alpha_H = \arctan\left(\sqrt{\frac{\mathcal{D}_{H^2}(p_0,p_1)}{2-\mathcal{D}_{H^2}(p_0,p_1)}}\right) \in [0, \frac{\pi}{4}] \ . \tag{5.69}$$

The estimation error produced by that optimal path is [244, Eq. 48]

$$\mathrm{MSE} = \frac{1}{N}\int_0^1 I(t)dt = \frac{1}{N}16\alpha_H^2 \ . \tag{5.70}$$

4. In Gelman and Meng [244, Eq. 49], the Hellinger distance is defined such that it is in $[0, \sqrt{2}]$. We here instead use the conventional definition of the squared Hellinger distance which is normalized so that it is in $[0, 1]$.

For two Gaussians

$$p_0 := \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{5.71}$$
$$p_1 := \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \tag{5.72}$$

the squared Hellinger distance can be written in closed-form

$$\mathcal{D}_{H^2}(p_0, p_1) = 1 - \frac{|\boldsymbol{\Sigma}_0|^{\frac{1}{4}}|\boldsymbol{\Sigma}_1|^{\frac{1}{4}}}{|\frac{1}{2}\boldsymbol{\Sigma}_0 + \frac{1}{2}\boldsymbol{\Sigma}_1|^{\frac{1}{2}}} \exp\Big(-\frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \big(\frac{1}{2}\boldsymbol{\Sigma}_0 + \frac{1}{2}\boldsymbol{\Sigma}_1\big)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\Big) \tag{5.73}$$

and plugs into the optimal path formula, which is also obtained in closed-form.

Estimation error from taking different paths

Proof of Theorem 5 *Polynomial error of annealed NCE with the geometric path*

We next study the estimation error produced by the geometric path (Figure 1). In the annealing limit $K \to \infty$, the MSE is written as

$$\text{MSE} = \frac{1}{N}\int_0^1 I(t)dt + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \ . \tag{5.74}$$

We recall from Grosse et al. [245] that the geometric path is closed for distributions in the exponential family : all distributions along the path remain in the exponential family. Furthermore, their Fisher information can be written in terms of the terms parameters [245, Eq. 17]; this is based off a a result of exponential families from [272, Section 3.3]

$$I(t) = \dot{\boldsymbol{\theta}}(t)^\top \boldsymbol{I}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{\theta}}(t)^\top \dot{\boldsymbol{\mu}}(t) \tag{5.75}$$

where $\boldsymbol{\mu}(t)$ are the generalized moments, defined as $\boldsymbol{\mu}(t) = \mathbb{E}_{\boldsymbol{x}\sim p_t(\boldsymbol{x})}[\boldsymbol{t}(\boldsymbol{x})] = \nabla_{\boldsymbol{\theta}} \log Z_t(\boldsymbol{\theta})$. It follows,

$$\frac{1}{N}\int_0^1 I(t)dt = \frac{1}{N}\int_0^1 \dot{\boldsymbol{\theta}}(t)^\top \dot{\boldsymbol{\mu}}(t)dt \ . \tag{5.76}$$

The geometric path is defined in parameter space by $\boldsymbol{\theta}_t = t\boldsymbol{\theta}_1 + (1-t)\boldsymbol{\theta}_0$, therefore

$$\frac{1}{N}\int_0^1 I(t)dt = \frac{1}{N}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^\top \int_0^1 \dot{\boldsymbol{\mu}}(t)dt = \frac{1}{N}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{5.77}$$

as in [245, Eq. 17]. For exponential families, $\log Z(\boldsymbol{\theta})$ is convex in $\boldsymbol{\theta}$. Here, we further assume strong convexity (with constant M) and smoothness (with constant L) so that

$$(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^\top(\nabla_{\boldsymbol{\theta}} \log Z_t(\boldsymbol{\theta}_1) - \nabla_{\boldsymbol{\theta}} \log Z_t(\boldsymbol{\theta}_0)) \tag{5.78}$$
$$\leq \frac{1}{M}\|\nabla_{\boldsymbol{\theta}} \log Z_t(\boldsymbol{\theta}_1) - \nabla_{\boldsymbol{\theta}} \log Z_t(\boldsymbol{\theta}_0)\|^2 \leq \frac{L^2}{M}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 \tag{5.79}$$

so that the MSE

$$\text{MSE} \leq \frac{L^2}{MN}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.80}$$

is polynomial in the euclidean distance between the parameters.

Proof of Theorem 6 *Exponential error of annealed NCE with the arithmetic path and "vanilla" schedule*

We now study the estimation error produced by the arithmetic path with "vanilla" schedule (table 5.2, line 3). Similarly, we start with the formula of the estimation error of NCE in the limit of a continuous path

$$\text{MSE} = \frac{1}{N}\int_0^1 I(t)dt + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.81}$$

where $I(t) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}[(\frac{d}{dt}\log p(\boldsymbol{x},t))^2]$ is the Fisher information over the path, using time t as the parameter. The arithmetic path is a Gaussian mixture (see table 5.2) so we will conveniently use the parametric form of the path to compute the Fisher information

$$I(t) = \dot{\tilde{w}}_t^\top I(\tilde{w}_t)\dot{\tilde{w}}_t \tag{5.82}$$

where the parameter here is the weight of the Gaussian mixture $\tilde{w}_t = tZ_1/(tZ_1 + 1 - t)$. We will need to compute two quantites : the Fisher information to that mixture parameter (not the time), and the parameter speed $\dot{\tilde{w}}_t$.

$$I(\tilde{w}_t) := \mathbb{E}_{\boldsymbol{x}\sim p_{\tilde{w}_t}}\Big[\Big(\frac{\partial \log p_{\tilde{w}_t}}{\partial \tilde{w}_t}(\boldsymbol{x})\Big)^2\Big] = \mathbb{E}_{\boldsymbol{x}\sim p_{\tilde{w}_t}}\Big[\Big(\frac{1}{p_{\tilde{w}_t}(\boldsymbol{x})}\frac{\partial p_{\tilde{w}_t}}{\partial \tilde{w}_t}(\boldsymbol{x})\Big)^2\Big] \tag{5.83}$$

$$= \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{(p_1(\boldsymbol{x}) - p_0(\boldsymbol{x}))^2}{p_{\tilde{w}_t}(\boldsymbol{x})}d\boldsymbol{x} = \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{(p_1(\boldsymbol{x}) - p_0(\boldsymbol{x}))^2}{(1-\tilde{w}_t)p_0(\boldsymbol{x}) + \tilde{w}_t p_1(\boldsymbol{x})}d\boldsymbol{x} \tag{5.84}$$

$$\geq \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{(p_1(\boldsymbol{x}) - p_0(\boldsymbol{x}))^2}{p_0(\boldsymbol{x}) + p_1(\boldsymbol{x})}d\boldsymbol{x} = \int p_0(\boldsymbol{x})\frac{\big(1 - \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\big)^2}{1 + \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}} = \mathcal{D}_\phi(p_1, p_0) \tag{5.85}$$

which is an f-divergence with generator $\phi(\boldsymbol{x}) = (1-x)^2/(1+x)$ that provides a t-independent lower bound. This will allow us to factor this quantity out of the integral defining the MSE, and simplify computations. We also have

$$\dot{\tilde{w}}_t := \frac{\partial}{\partial t}\tilde{w}_t = \frac{1}{t(1-t)} \times \sigma\Big(\log\frac{tZ_1}{1-t}\Big) \times \Big(1 - \sigma\Big(\log\frac{tZ_1}{1-t}\Big)\Big) \tag{5.86}$$

$$= \frac{1}{t(1-t)} \times \frac{tZ_1}{(1-t) + tZ_1} \times \frac{(1-t)}{(1-t) + tZ_1} = \frac{Z_1}{((1-t) + tZ_1)^2} \ . \tag{5.87}$$

where we choose to keep the dependency on t. The intuition is that integrating this quantity will yield a function of Z_1, which will drive the MSE toward high values. We next show this rigorously and finally compute the estimation error.

$$\frac{1}{N}\int_0^1 I(t)dt = \frac{1}{N}\int_0^1 \dot{\tilde{w}}(t)I(\tilde{w}(t))\dot{\tilde{w}}(t)dt \tag{5.88}$$

$$\geq \frac{1}{N} \times \mathcal{D}_\phi(p_1, p_0) \times \int_0^1 \dot{\tilde{w}}(t)^2 dt \tag{5.89}$$

$$= \frac{1}{N} \times \mathcal{D}_\phi(p_1, p_0) \times Z_1^2 \times \int_0^1 \frac{1}{(t(Z_1 - 1) + 1)^4}dt \tag{5.90}$$

$$= \frac{1}{N} \times \mathcal{D}_\phi(p_1, p_0) \times Z_1^2 \times \frac{Z_1^2 + Z_1 + 1}{3Z_1^3} \tag{5.91}$$

$$= \frac{1}{3N} \times \mathcal{D}_\phi(p_1, p_0) \times (Z_1^{-1} + 1 + Z_1) \ . \tag{5.92}$$

We would like to write Z_1 in terms of the parameters. To do so, we now suppose the unnormalized target is in a simply unnormalized exponential family. Consequently,

$$Z_1 := \exp(\log Z(\boldsymbol{\theta}_1) - \log Z(\boldsymbol{\theta}_0) + \log Z(\boldsymbol{\theta}_0)) \tag{5.93}$$

$$\geq \exp\left(\nabla \log Z(\boldsymbol{\theta}_0)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) + \frac{M}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 + \log Z(\boldsymbol{\theta}_0)\right) \ . \tag{5.94}$$

using the strong convexity of the log-partition function. For large enough $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\| > 0$, the quadratic term in the exponential is larger than the linear term, and the divergence $\mathcal{D}_\phi(p_1, p_0)$ is larger than a constant. It follows that for large enough $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\| > 0$, there exists a constant $C > 0$ such that the MSE grows (at least) exponentially with the parameter-distance

$$\text{MSE} > \frac{C}{3N} \times \exp\left(\frac{M}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2\right) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \ . \tag{5.95}$$

Proof of Theorem 7 *Polynomial error of annealed NCE with the arithmetic path and "oracle" schedule*

We now study the estimation error produced by the arithmetic path with "oracle" schedule (table 5.2, line 4). Similarly, we start with the formula of the estimation error of NCE annealed over a continuous path

$$\text{MSE} = \frac{1}{N}\int_0^1 I(t)dt + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \tag{5.96}$$

where $I(t) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}[(\frac{d}{dt}\log p(\boldsymbol{x}, t))^2]$ is the Fisher information over the path, using time t as the parameter. The arithmetic path is the Gaussian mixture $p_t(\boldsymbol{x}) = tp_1(\boldsymbol{x}) + (1-t)p_0(\boldsymbol{x})$ (see table 5.2). The Fisher information is therefore

$$I(t) := \mathbb{E}_{\boldsymbol{x}\sim p_t}\left[\left(\frac{\partial \log p_t}{\partial t}(\boldsymbol{x})\right)^2\right] = \mathbb{E}_{\boldsymbol{x}\sim p_t}\left[\left(\frac{1}{p_t(\boldsymbol{x})}\frac{\partial p_t}{\partial t}(\boldsymbol{x})\right)^2\right] \tag{5.97}$$

$$= \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{(p_1(\boldsymbol{x}) - p_0(\boldsymbol{x}))^2}{p_t(\boldsymbol{x})}d\boldsymbol{x} = \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{(p_1(\boldsymbol{x}) - p_0(\boldsymbol{x}))^2}{(1-t)p_0(\boldsymbol{x}) + tp_1(\boldsymbol{x})}d\boldsymbol{x} \tag{5.98}$$

$$\leq \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{p_1(\boldsymbol{x})^2 + p_0(\boldsymbol{x})^2}{(1-t)p_0(\boldsymbol{x}) + tp_1(\boldsymbol{x})}d\boldsymbol{x} \tag{5.99}$$

where we choose to keep the dependency on t in the bound.

We briefly justify this choice. We had first tried a t-independent bound, which led to an upper bound of the MSE that was too loose. We share insight as to why : first, recognize that the fraction can be broken in two terms, each of them a chi-square divergence between an endpoint of the path (p_0 or p_1) and the mixture p_t. Each of them admits a t-independent upper bound given by the chi-square divergence between the endpoints p_0 and p_1, using lemma 1. However, the chi-square divergence between two Gaussians, for example, is exponential (not polynomial) in the natural parameters [271,

eq 7.41]. In fact, plotting $I(t)$ for a univariate Gaussian model revealed that it took high values at the endpoints $t = 0$ and $t = 1$, and was near zero almost everywhere else in the interval $t \in [0, 1]$, which again suggested that dropping the dependency on t was unreasonable.

Now we can compute the estimation error, as

$$\frac{1}{N}\int_0^1 I(t)dt \leq \frac{1}{N}\int_{\mathbb{R}^d}\int_0^1 \frac{p_1(\boldsymbol{x})^2 + p_0(\boldsymbol{x})^2}{(1-t)p_0(\boldsymbol{x}) + tp_1(\boldsymbol{x})}dtd\boldsymbol{x} = \frac{1}{N}(J_1 + J_2) \ . \tag{5.100}$$

Let us try to solve one of these integrals, say J_1.

$$J_1 = \int_{\mathbb{R}^d}\int_0^1 \frac{p_1(\boldsymbol{x})^2}{(1-t)p_0(\boldsymbol{x}) + tp_1(\boldsymbol{x})}dtd\boldsymbol{x} = \int_{\mathbb{R}^d}\frac{p_1(\boldsymbol{x})^2}{p_0(\boldsymbol{x})}\Big(\int_0^1 \frac{1}{1 + t(\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} - 1)}dt\Big)d\boldsymbol{x} \tag{5.101}$$

$$= \int_{\mathbb{R}^d}\frac{p_1(\boldsymbol{x})^2}{p_0(\boldsymbol{x})}\Big(\frac{1}{\frac{p_1}{p_0} - 1}\log\frac{p_1}{p_0}\Big)d\boldsymbol{x} = 1 + \mathbb{E}_{p_1}\Big[\frac{1}{1 - \frac{p_0}{p_1}}\log\frac{p_1}{p_0} - 1\Big] = 1 + \mathcal{D}_\phi(p_0, p_1) \ . \tag{5.102}$$

which we rewrote using an f-divergence defined by $\phi(x) = \frac{-\log(x)}{1-x} - 1$. Similarly, we obtain

$$J_2 = \int_{\mathbb{R}^d}\int_0^1 \frac{p_0(\boldsymbol{x})^2}{(1-t)p_0(\boldsymbol{x}) + tp_1(\boldsymbol{x})}dtd\boldsymbol{x} = \int_{\mathbb{R}^d}\frac{p_0(\boldsymbol{x})^2}{p_1(\boldsymbol{x})}\Big(\int_0^1 \frac{1}{\frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})} + t(1 - \frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})})}dt\Big)d\boldsymbol{x} \tag{5.103}$$

$$= \int_{\mathbb{R}^d}\frac{p_0(\boldsymbol{x})^2}{p_1(\boldsymbol{x})}\Big(\frac{1}{\frac{p_0}{p_1} - 1}\log\frac{p_0}{p_1}\Big)d\boldsymbol{x} = 1 + \mathbb{E}_{p_0}\Big[\frac{1}{1 - \frac{p_1}{p_0}}\log\frac{p_0}{p_1} - 1\Big] = 1 + \mathcal{D}_\phi(p_1, p_0) \ . \tag{5.104}$$

Putting this together, we get

$$\frac{1}{N}\int_0^1 I(t)dt \leq \frac{1}{N}(2 + \mathcal{D}_\phi(p_0, p_1) + \mathcal{D}_\phi(p_1, p_0)) \ . \tag{5.105}$$

How does this divergence depend on the parameter-distance $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|$? Does it bring down the dependency from exponential to something lower? We next analyze this :

$$\mathcal{D}_\phi(p_0, p_1) + 1 = \mathbb{E}_{p_1}\frac{1}{1 - \frac{p_0}{p_1}}\log\frac{p_1}{p_0}$$

which looks like a Kullback-Leibler divergence, where the integrand is reweighted by $\frac{1}{1-\frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}}$. Note that

$$\begin{cases} \frac{1}{1-\frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}} \geq 1 & p_0(\boldsymbol{x}) \leq p_1(\boldsymbol{x}) \\ \frac{1}{1-\frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}} < 1 & p_0(\boldsymbol{x}) > p_1(\boldsymbol{x}) \end{cases} \tag{5.106}$$

which motivates separating the integral over both domains

$$1 + \mathcal{D}_\phi(p_0, p_1) = \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p_0(\boldsymbol{x}) \leq p_1(\boldsymbol{x})\}} p_1(\boldsymbol{x})\log\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\frac{1}{1 - \frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}} \tag{5.107}$$

$$+ \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p_0(\boldsymbol{x}) > p_1(\boldsymbol{x})\}} p_1(\boldsymbol{x})\log\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\frac{1}{1 - \frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}} \tag{5.108}$$

$$\leq \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p_0(\boldsymbol{x})\leq p_1(\boldsymbol{x})\}} p_1(\boldsymbol{x}) + \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p_0(\boldsymbol{x})> p_1(\boldsymbol{x})\}} p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \tag{5.109}$$

$$\leq 1 + D_{\mathrm{KL}}(p_1, p_0) \tag{5.110}$$

Hence we get

$$\frac{1}{N}\int_0^1 I(t)dt \leq \frac{1}{N} \times (2 + \mathcal{D}_{\mathrm{KL}}(p_0, p_1) + \mathcal{D}_{\mathrm{KL}}(p_1, p_0)) \ . \tag{5.111}$$

We now suppose the proposal and target are distributions in an exponential family. The KL divergence between exponential distributions with parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$, is given by the Bregman divergence of the log-partition on the swapped parameters [257, Eq. 29]

$$\mathcal{D}_{\mathrm{KL}}(p_0, p_1) = \mathcal{D}^{\mathrm{Bregman}}_{\log Z}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_0) := \log Z(\boldsymbol{\theta}_1) - \log Z(\boldsymbol{\theta}_0) - \nabla \log Z(\boldsymbol{\theta}_0)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) \tag{5.112}$$

$$\leq \frac{L}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2 \tag{5.113}$$

Hence

$$\mathrm{MSE} \leq \frac{1}{N} \times (2 + L\|\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\|^2) + o\left(\frac{1}{N}\right) + o\left(\frac{K^2}{N}\right) \tag{5.114}$$

using the L-smoothness of the log-partition function $\log Z(\boldsymbol{\theta})$.

Discussion on the assumptions for theorems 4, 5, 6, 7 For these theorems, we have supposed that the target and proposal distributions are in an exponential family with a log parition that verifies

$$M\,\mathbf{Id} \preccurlyeq \nabla^2_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) \preccurlyeq L\,\mathbf{Id} \ . \tag{5.115}$$

We now look at the validity of this assumption for a simple example : the univariate Gaussian, which is in an exponential family. The canonical parameters are its mean and variance (μ, v). Written as an exponential family,

$$p(\boldsymbol{x}) := \exp(\langle \boldsymbol{\theta}, \boldsymbol{t}(\boldsymbol{x})\rangle - \log Z(\boldsymbol{\theta})) \tag{5.116}$$

the natural parameters are $\boldsymbol{\theta} = (\mu/v, -1/(2v))$, associated with the sufficient statistics $\boldsymbol{t}(x) = (x, x^2)$ [257]. The log-partition function and its derivatives are

$$\log Z(\boldsymbol{\theta}) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2) \tag{5.117}$$

$$\nabla \log Z(\boldsymbol{\theta}) = \mathbb{E}_{x\sim p}[t(x)] = \left(-\frac{\theta_1}{2\theta_2}, -\frac{1}{2\theta_2} + \frac{\theta_1^2}{4\theta_2^2}\right) \tag{5.118}$$

$$\nabla^2 \log Z(\boldsymbol{\theta}) = \mathrm{Var}_{x\sim p}[t(x)] = \frac{1}{2\theta_2}\begin{pmatrix} -1 & \frac{\theta_1}{\theta_2} \\ \frac{\theta_1}{\theta_2} & \frac{1}{\theta_2} - \frac{1}{2}\frac{\theta_1^2}{\theta_2} \end{pmatrix} = \begin{pmatrix} v & 2\mu v \\ 2\mu v & 2v^2 - \mu^2 \end{pmatrix} \tag{5.119}$$

When the mean is zero, the eigenvalues of the Hessian are in fact the diagonal values $(v, 2v^2)$, and they are bounded if and only if the variance v is bounded.

Proof of Theorem 8 *Constant error of annealed NCE with the arithmetic path and "oracle-trig" schedule*

We now study the estimation error produced by the arithmetic path with "oracle-trig" schedule (table 5.2, line 5). We write the optimal path of Eq. 5.66 in the limit when the distributions have little overlap, pointwise (with error $\epsilon^{'}$) and on average (with error ϵ) :

$$\sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})} = \epsilon^{'}(\boldsymbol{x}) \tag{5.120}$$

$$\int \sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})}d\boldsymbol{x} = \epsilon \tag{5.121}$$

We briefly note that there exist certain conditions, given by the dominated convergence theorem, where the first error (pointwise) going to zero implies the second (on average) going to zero as well, but this is outside the scope of this proof. Now, many relevant quantities involved in the optimal distribution and its estimation error simplify as $\epsilon^{'}(\boldsymbol{x}) \to 0$ pointwise and $\epsilon \to 0$. The notation $O(\cdot)$ will hide dependencies on absolute constants only.

$$\mathcal{D}_{H^2}(p_0, p_1) := 1 - \int \sqrt{p_0 p_1} = 1 - \epsilon \tag{5.122}$$

$$\alpha_H := \arctan\left(\sqrt{\frac{\mathcal{D}_{H^2}(p_0,p_1)}{2 - \mathcal{D}_{H^2}(p_0,p_1)}}\right) = \frac{\pi}{4} - \frac{\epsilon}{2} + o(\epsilon) \tag{5.123}$$

$$a_t := \frac{\cos((2t-1)\alpha_H)}{2\cos(\alpha_H)} - \frac{\sin((2t-1)\alpha_H)}{2\sin(\alpha_H)} = \cos\left(\frac{\pi t}{2}\right) + \epsilon(t-1)\sin\left(\frac{\pi t}{2}\right) + o(\epsilon) \tag{5.124}$$

$$b_t = \frac{\cos((2t-1)\alpha_H)}{2\cos(\alpha_H)} + \frac{\sin((2t-1)\alpha_H)}{2\sin(\alpha_H)} = \sin\left(\frac{\pi t}{2}\right) - \epsilon t\cos\left(\frac{\pi t}{2}\right) + o(\epsilon) \tag{5.125}$$

$$\partial_t a_t := -\alpha_H\left(\frac{\sin((2t-1)\alpha_H)}{\cos(\alpha_H)} + \frac{\cos((2t-1)\alpha_H)}{\sin(\alpha_H)}\right) \tag{5.126}$$

$$= -\frac{\pi}{2}\sin\left(\frac{\pi t}{2}\right) + \epsilon\left(\sin\left(\frac{\pi t}{2}\right) + \frac{\pi}{2}(t-1)\cos\left(\frac{\pi t}{2}\right)\right) + o(\epsilon) \tag{5.127}$$

$$\partial_t b_t := -\alpha_H\left(\frac{\sin((2t-1)\alpha_H)}{\cos(\alpha_H)} - \frac{\cos((2t-1)\alpha_H)}{\sin(\alpha_H)}\right) \tag{5.128}$$

$$= \frac{\pi}{2}\cos\left(\frac{\pi t}{2}\right) + \epsilon\left(\frac{\pi}{2}t\sin\left(\frac{\pi t}{2}\right) - \cos\left(\frac{\pi t}{2}\right)\right) + o(\epsilon) \tag{5.129}$$

$$a_t \times \partial_t a_t := -\frac{\pi}{4}\sin(\pi t) + \epsilon\left(\frac{1}{2}\sin(\pi t) + \frac{\pi}{2}(t-1)\cos(\pi t)\right) + o(\epsilon) \tag{5.130}$$

$$a_t \times \partial_t b_t := \frac{\pi}{2}\cos^2\left(\frac{\pi t}{2}\right) + \epsilon\left(\frac{\pi}{2}(1-2t)\sin(\pi t) + \cos(\pi t) + 1\right) + o(\epsilon) \tag{5.131}$$

$$b_t \times \partial_t a_t := -\frac{\pi}{2}\sin^2\left(\frac{\pi t}{2}\right) + \epsilon\left(\frac{1}{2} - \frac{1}{2}\cos(\pi t) + \frac{1}{2}\frac{\pi}{2}\sin(\pi t)(2t-1)\right) + o(\epsilon) \tag{5.132}$$

$$b_t \times \partial_t b_t := \frac{\pi}{4}\sin(\pi t) + \epsilon\left(-\frac{1}{2}\sin(\pi t) - \frac{\pi}{2}t\cos(\pi t)\right) + o(\epsilon) \tag{5.133}$$

This leads to the following simplification of the optimal path

$$p_t^{\text{opt}}(\boldsymbol{x}) := \left(a_t\sqrt{p_0(\boldsymbol{x})} + b_t\sqrt{p_1(\boldsymbol{x})}\right)^2 = a(t)^2 p_0(\boldsymbol{x}) + b(t)^2 p_1(\boldsymbol{x}) + 2a(t)b(t)\epsilon^{'}(\boldsymbol{x}) \tag{5.134}$$

$$= a(t)^2 p_0(\boldsymbol{x}) + b(t)^2 p_1(\boldsymbol{x}) + 2a(t)b(t)\epsilon^{'}(\boldsymbol{x}) \tag{5.135}$$

$$= \cos^2\big(\frac{\pi t}{2}\big)p_0(\boldsymbol{x}) + \sin^2\big(\frac{\pi t}{2}\big)p_1(\boldsymbol{x}) + \epsilon^{'}(\boldsymbol{x})\sin(\pi t) \tag{5.136}$$

$$+ \epsilon\sin(\pi t)(p_0(\boldsymbol{x})(t-1) - p_1(\boldsymbol{x})t) + \epsilon\,\epsilon^{'}(\boldsymbol{x})((1-2t)\cos(\pi t) - 1) + o(\epsilon) \tag{5.137}$$

$$= p_t^{\text{arith}-\text{trig}}(\boldsymbol{x}) + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_1(\boldsymbol{x})) \tag{5.138}$$

where we denoted by

$$p_t^{\text{arith}-\text{trig}}(\boldsymbol{x}) := \cos^2\big(\frac{\pi t}{2}\big)p_0(\boldsymbol{x}) + \sin^2\big(\frac{\pi t}{2}\big)p_1(\boldsymbol{x}) \tag{5.139}$$

the arithmetic path with "oracle-trig" schedule defined in table 5.2 (line 5); the trigonometric weights in evolve slowly at the end points $t = 0$ and $t = 1$. We also define $g_1(\boldsymbol{x}) = \sin(\pi t)(p_0(\boldsymbol{x})(t-1) - p_1(\boldsymbol{x})t)$ which is an integrable function. This proves the first part of this theorem, which is that the optimal path p^{opt} is close to a certain arithmetic path (with trigonometric weights) $p^{\text{arith}-\text{trig}}$, and that closeness is controlled by how little overlap there is between the endpoint distributions p_0 and p_1, on average and pointwise.

Similarly, we can control how close these two paths are in terms of estimation errors. The estimation error in Eq. 5.70 produced by a path is

$$\text{MSE} := \frac{1}{N}\int_0^1 I(t)dt + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.140}$$

$$= \frac{1}{N}\int_{\mathbb{R}^D}\int_0^1 \big(\partial_t \log p_t(\boldsymbol{x})\big)^2 p_t(\boldsymbol{x})dtd\boldsymbol{x} + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.141}$$

$$= \frac{1}{N}\int_{\mathbb{R}^D}\int_0^1 \Big(\frac{\partial_t p_t(\boldsymbol{x})}{p_t(\boldsymbol{x})}\Big)^2 p_t(\boldsymbol{x})dtd\boldsymbol{x} + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.142}$$

We denote by $\text{MSE}_{\text{optimal}}$ and $\text{MSE}_{\text{arith}-\text{trig}}$ the estimation errors produced respectively by the optimal path and the arithmetic path with trigonometric weights. The estimation error of the optimal path can be Taylor-expanded in terms of the overlap (ϵ and $\epsilon^{'}(\boldsymbol{x})$) as well. We next compute the intermediate quantities that are required to obtain the estimation error.

$$\partial_t p_t^{\text{opt}}(\boldsymbol{x}) := \partial_t\Big(\big(a_t\sqrt{p_0(\boldsymbol{x})} + b_t\sqrt{p_1(\boldsymbol{x})}\big)^2\Big) \tag{5.143}$$

$$= 2(\partial_t a_t\sqrt{p_0(\boldsymbol{x})} + \partial_t b_t\sqrt{p_1(\boldsymbol{x})})(a_t\sqrt{p_0(\boldsymbol{x})} + b_t\sqrt{p_1(\boldsymbol{x})}) \tag{5.144}$$

$$= 2p_0(\boldsymbol{x})\big(a_t \times \partial_t a_t\big) + 2p_1(\boldsymbol{x})\big(b_t \times \partial_t b_t\big) + \tag{5.145}$$

$$2\sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})}\big(\partial_t a_t \times b_t + a_t \times \partial_t b_t\big) \tag{5.146}$$

$$= \pi\sin(\pi t)\big(\frac{p_1(\boldsymbol{x}) - p_0(\boldsymbol{x})}{2}\big) + \pi\cos(\pi t)\sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})} + \epsilon\Big(\tag{5.147}$$

$$p_0(\boldsymbol{x})\big(\sin(\pi t) + \pi(t-1)\cos(\pi t)\big) + p_1(\boldsymbol{x})\big(-\sin(\pi t) - \pi t\cos(\pi t)\big) + \tag{5.148}$$

$$\sqrt{p_0(\boldsymbol{x})p_1(\boldsymbol{x})}\big(\sin(\pi t)(1-2t)\frac{\pi}{2} + \cos(\pi t) + 3\big)\Big) + o(\epsilon) \tag{5.149}$$

$$= \pi\sin(\pi t)\big(\frac{p_1(\boldsymbol{x}) - p_0(\boldsymbol{x})}{2}\big) + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x})) \tag{5.150}$$

$$= \partial_t p_t^{\text{arith-trig}}(\boldsymbol{x}) + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x})) \tag{5.151}$$

where $g_2(\boldsymbol{x}) = p_0(\boldsymbol{x})\big(\sin(\pi t)+\pi(t-1)\cos(\pi t)\big)+p_1(\boldsymbol{x})\big(-\sin(\pi t)-\pi t\cos(\pi t)\big)$ is an integrable function. It follows,

$$\big(\partial_t p_t(\boldsymbol{x})\big)^2 = \big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x}) + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x}))\big)^2 \tag{5.152}$$

$$= \big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2 + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x})) \tag{5.153}$$

by expanding and using that $\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x}) = \pi\sin(\pi t)\big(\frac{p_1(\boldsymbol{x})-p_0(\boldsymbol{x})}{2}\big)$ is bounded so that $\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\times \epsilon^{'}(\boldsymbol{x}) = O(\epsilon^{'}(\boldsymbol{x}))$. Moreover,

$$\frac{1}{p_t^{\text{opt}}(\boldsymbol{x})} = \frac{1}{p_t^{\text{arith-trig}}(\boldsymbol{x}) + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_1(\boldsymbol{x}))} = \frac{1}{p_t^{\text{arith-trig}}(\boldsymbol{x})}\frac{1}{1+\frac{O(\epsilon^{'}(\boldsymbol{x}))+O(\epsilon g_1(\boldsymbol{x}))}{p_t^{\text{arith-trig}}(\boldsymbol{x})}} \quad . \tag{5.154}$$

Denoting by $\epsilon^{''}(\boldsymbol{x}) := (O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_1(\boldsymbol{x})))/p_t^{\text{arith-trig}}(\boldsymbol{x})$ a third quantity which we assume goes to zero, we get

$$\frac{1}{p_t^{\text{opt}}(\boldsymbol{x})} = \frac{1}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + \frac{O(\epsilon^{''}(\boldsymbol{x}))}{p_t^{\text{arith-trig}}(\boldsymbol{x})} \quad . \tag{5.155}$$

We can now write

$$(\partial_t \log p_t(\boldsymbol{x}))^2 \times p_t(\boldsymbol{x}) = \frac{\big(\partial_t p_t(\boldsymbol{x})\big)^2}{p_t(\boldsymbol{x})} \tag{5.156}$$

$$= \Big(\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2 + O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x}))\Big) \times \Big(\frac{1}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + \frac{O(\epsilon^{''}(\boldsymbol{x}))}{p_t^{\text{arith-trig}}(\boldsymbol{x})}\Big) \tag{5.157}$$

$$= \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + \frac{O(\epsilon^{'}(\boldsymbol{x})) + O(\epsilon g_2(\boldsymbol{x}))}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})}O(\epsilon^{''}(\boldsymbol{x})) \tag{5.158}$$

$$= \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + O(\epsilon^{''}(\boldsymbol{x}))\Big(1 + \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})}\Big) \tag{5.159}$$

by expanding. We can then write

$$\text{MSE}_{\text{optimal}} \tag{5.160}$$

$$= \frac{1}{N}\int_{\mathbb{R}^D}\int_0^1 \frac{(\partial_t p_t^{\text{opt}}(\boldsymbol{x}))^2}{p_t^{\text{opt}}(\boldsymbol{x})}dtd\boldsymbol{x} + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.161}$$

$$= \frac{1}{N}\int_{\mathbb{R}^D}\int_0^1 \Big(\frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})} + O(\epsilon^{''}(\boldsymbol{x}))\Big(1 + \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})}\Big)\Big)dtd\boldsymbol{x} \tag{5.162}$$

$$+ o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.163}$$

$$= \text{MSE}_{\text{arith-trig}} + \frac{1}{N}\int_{\mathbb{R}^D}\int_0^1 O(\epsilon^{''}(\boldsymbol{x}))\Big(1 + \frac{\big(\partial_t p_t^{\text{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\text{arith-trig}}(\boldsymbol{x})}\Big)dtd\boldsymbol{x} + \tag{5.164}$$

$$o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.165}$$

$$= \mathrm{MSE}_{\mathrm{arith-trig}} + O\Big(\frac{\epsilon^{'''}}{N}\Big) + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.166}$$

Additionally, we assume that the remainder term denoted by

$$\epsilon^{'''} = \int_{\mathbb{R}^D}\int_0^1 O(\epsilon^{''}(\boldsymbol{x}))\Big(1 + \frac{\big(\partial_t p_t^{\mathrm{arith-trig}}(\boldsymbol{x})\big)^2}{p_t^{\mathrm{arith-trig}}(\boldsymbol{x})}\Big)dtd\boldsymbol{x} \tag{5.167}$$

is integrable and also goes to zero. On the other hand, the estimation error produced by the optimal path is known [244, Eq. 48] and the result can be Taylor-expanded as well

$$\mathrm{MSE}_{\mathrm{optimal}} = \frac{1}{N}16\alpha_H^2 + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.168}$$

$$= \frac{1}{N}16\Big(\frac{\pi^2}{16} + O(\epsilon)\Big) + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) \tag{5.169}$$

$$= \frac{1}{N}\pi^2 + O\Big(\frac{\epsilon}{N}\Big) + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big)\ . \tag{5.170}$$

This means that

$$\mathrm{MSE}_{\mathrm{arith-trig}} = \frac{1}{N}\pi^2 + O\Big(\frac{\epsilon}{N}\Big) + o\Big(\frac{1}{N}\Big) + o\Big(\frac{K^2}{N}\Big) + O\Big(\frac{\epsilon^{'''}}{N}\Big)\ . \tag{5.171}$$

5.8.3 . Useful Lemma

Lemma 1 (Chi-square divergence of between a density and a mixture) *We wish to upper bound the chi-square divergence between a distribution $p(\boldsymbol{x})$ and a mixture $wp(\boldsymbol{x}) + (1-w)q(\boldsymbol{x})$, where $0 < w < 1$.*

$$\mathcal{D}_{\chi^2}(p, wp + (1-w)q) = \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{p(\boldsymbol{x})^2}{wp(\boldsymbol{x}) + (1-w)q(\boldsymbol{x})}d\boldsymbol{x} - 1 \tag{5.172}$$

$$= \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p(\boldsymbol{x})<q(\boldsymbol{x})\}} \frac{p(\boldsymbol{x})^2}{wp(\boldsymbol{x}) + (1-w)q(\boldsymbol{x})}d\boldsymbol{x} \tag{5.173}$$

$$+ \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p(\boldsymbol{x})>q(\boldsymbol{x})\}} \frac{p(\boldsymbol{x})^2}{wp(\boldsymbol{x}) + (1-w)q(\boldsymbol{x})}d\boldsymbol{x} - 1 \tag{5.174}$$

$$\leq \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p(\boldsymbol{x})<q(\boldsymbol{x})\}} \frac{p(\boldsymbol{x})^2}{wp(\boldsymbol{x}) + (1-w)p(\boldsymbol{x})}d\boldsymbol{x} \tag{5.175}$$

$$+ \int_{\{\boldsymbol{x}\in\mathbb{R}^D | p(\boldsymbol{x})>q(\boldsymbol{x})\}} \frac{p(\boldsymbol{x})^2}{wq(\boldsymbol{x}) + (1-w)q(\boldsymbol{x})}d\boldsymbol{x} - 1 \tag{5.176}$$

$$\leq \int_{\boldsymbol{x}\in\mathbb{R}^D} p(\boldsymbol{x})d\boldsymbol{x} + \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{p(\boldsymbol{x})^2}{q(\boldsymbol{x})}d\boldsymbol{x} - 1 \tag{5.177}$$

$$= \int_{\boldsymbol{x}\in\mathbb{R}^D} \frac{p(\boldsymbol{x})^2}{q(\boldsymbol{x})}d\boldsymbol{x} = \mathcal{D}_{\chi^2}(p, q) + 1 \tag{5.178}$$

Conclusion

This thesis manuscript studies self-supervised learning as a topic of increasing interest in machine learning literature. In section 1, we explained that self-supervised learning effectively reframes an unsupervised problem into a supervised problem, so a prediction task (classification or regression). Part I explored what prediction tasks are able to learn on brain imaging data, using deep neural networks with a focus on interpretability. Part II pays greater interest to the question of what self-supervised learning actually learns. We chose a basic prediction task, binary classification that is related to estimating an unnormalized density model. We analyzed how the estimation error is impacted by the design of the task and how annealing can formally help. Yet much remains to be done : we next discuss some research questions and lay out the blueprint to naturally extend this manuscript.

Extension to other prediction tasks In this manuscript we focused on binary classification which despite its apparent simplicity, is shown to be a very rich framework. The main argument relating binary classification to unsupervised learning is the Bayes-predictor learns (a ratio of) densities. A natural question is : can the Bayes-predictor for other prediction tasks be used for unsupervised learning ?

Extension to multi-class classification. Similar to the binary case, a multi-class classification loss that is a proper and composite can also be expressed in terms of a Bregman divergence between the true and model class-probabilities [273] or equivalently between the true and model density-ratios [274, Eq.7] (with respect to the reference distribution for class 0). From that viewpoint, our estimation error analysis could be extended to popular losses (*e.g.* InfoNCE [26] and RankingNCE [42]) in the multi-class case.

Extension to regression. Other popular losses in self-supervised learning (*e.g.* InfoNCE [26] and RankingNCE [42]) are based on a multi-class classification problem. The Bayes-predictor for regression tasks with a Bregman loss is the conditional expectation of the target given the input $f(\boldsymbol{x}) = \mathbb{E}[Y|X = \boldsymbol{x}]$ [275]. This is a well-known setup where this Bayes-regressor is tractable : when the input is a gaussian-noised version of the data and the target is the original data. For this denoising task, the Bayes-regressor is related to the Stein score of the noised density $\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y})$ by Tweedie's formula [276, 277]. Hence, solving the regression task can be used to learn an unnormalized model of the perturbed data distribution. Works by collaborators are proving how this estimation method, like NCE, can suffer from an error that is exponentially large in relevant quantities [278] and that annealing this estimation method provably brings down that error [279].

Last, we note that self-supervised learning is a broad field that extends well beyond the scope of this thesis. Its success has been explored [280, 281], even questioned [282], through the lens of information theory, representation learning [26, 283], statistical inference [42, 16–18, 284], and from other perspectives as well. Future theory will be needed to determine how relevant these perspectives are to explain the practical success of self-supervised learning.

Synthèse en français

L'apprentissage auto-supervisé a gagné en popularité en tant que méthode d'apprentissage à partir de données non annotées. Il s'agit essentiellement de créer puis de résoudre un problème de prédiction qui utilise les données ; par exemple, de retrouver l'ordre de données qui ont été mélangées. Ces dernières années, cette approche a été utilisée avec succès pour entraîner des réseaux de neurones qui extraient des représentations utiles des données, le tout sans aucune annotation. Cependant, notre compréhension de ce qui est appris et de la qualité de cet apprentissage est limitée. Ce document éclaire ces deux aspects de l'apprentissage auto-supervisé.

Empiriquement, nous évaluons ce qui est appris en résolvant des tâches auto-supervisés. Nous spécialisons des tâches de prédiction lorsque les données sont des enregistrements d'activité cérébrale, par magnétoencéphalographie (MEG) ou électroencephalographie (EEG). Ces tâches partagent un objectif commun : reconnaître la structure temporelle dans les ondes cérébrales. Nos résultats montrent que les représentations apprises en résolvant ces tâches-là comprennent des informations neurophysiologiques, cognitives et cliniques, interprétables.

Théoriquement, nous explorons également la question de la qualité de l'appretissage, spécifiquement pour les tâches de prédiction qui peuvent s'écrire comme un problème de classification binaire. Nous poursuivons une trâme de recherche qui utilise des problèmes de classification binaire pour faire de l'inférence statistique, alors que cela peut nécessiter de sacrifier une notion d'efficacité statistique pour une autre notion d'efficacité computationnelle. Nos contributions visent à améliorer l'efficacité statistique. Nous analysons théoriquement l'erreur d'estimation statistique et trouvons des situations lorsque qu'elle peut rigoureusement être réduite. Spécifiquement, nous caractérisons des hyperparametres optimaux de la tâche de classification binaire et prouvons également que la populaire heuristique de "recuit" peut rendre l'estimation plus efficace, même en grandes dimensions.

Bibliographie

[1] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning : From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning : data mining, inference and prediction*. Springer, 2 edition, 2009.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[4] A. Hyvärinen, I. Khemakhem, and H. Morioka. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning, 2023.

[5] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR)*, 2017.

[6] V. Escorcia, J. Carlos Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1256–1264, 2015.

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection : Quantifying interpretability of deep visual representations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017.

[8] J. Oramas, K. Wang, and T. Tuytelaars. Visual explanation by interpretation : Improving visual feedback capabilities of deep neural networks. In *International Conference on Learning Representations*, 2019.

[9] M. Nava, J. Guzzi, R.O. Chavez-Garcia, L.M. Gambardella, and A. Giusti. Learning long-range perception using self-supervision from short-range sensors and odometry. *IEEE Robotics and Automation Letters*, 4 :1279–1286, 2018.

[10] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2070–2079, 2017.

[11] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(11) :307–361, 2012.

[12] B. Liu, D.J. Hsu, P. Ravikumar, and A. Risteski. Masked prediction : A parameter identifiability view. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 21241–21254. Curran Associates, Inc., 2022.

[13] I. Misra Y. LeCun. Self-supervised learning : The dark matter of intelligence. https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/, 2021. [Online; accessed 24-09-2023].

[14] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *International Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates Inc., 2020.

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

[16] A. Hyvärinen and H. Morioka. Nonlinear ICA of temporally dependent stationary sources. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 460–469. PMLR, 20–22 Apr 2017.

[17] A. Hyvärinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29. Curran Associates, Inc., 2016.

[18] A. Hyvärinen, H. Sasaki, and R.E. Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89, pages 859–868. PMLR, 2019.

[19] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. Gordon Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum. A cookbook of self-supervised learning, 2023.

[20] H. Lee, C. Pabbaraju, A. Sevekari, and A. Risteski. Pitfalls of gaussians as a noise distribution in NCE. In *International Conference on Learning Representations (ICLR)*. arXiv, 2022.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

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

[24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008.

[25] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23 :1661–1674, 2011.

[26] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018.

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

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[29] W. Stuetzle A. Buja and Y. Shen. *Loss functions for binary class probability estimation and classification : Structure and applications.* Technical Report, University of Pennsylvania, November 2005, Philadelphia, PA, 2005.

[30] T. Gneiting and A. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102 :359–378, 2007.

[31] M. Reid and R. Williamson. Composite binary losses. *Journal of Machine Learning Research (JMLR)*, 11(83) :2387–2422, 2010.

[32] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7 :200–217, 1967.

[33] A. Menon and C.S. Ong. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning (ICML)*, 2016.

[34] L. Wasserman. *All of Statistics : A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010. ISBN 1441923225.

[35] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5) :359–366, 1989. ISSN 0893-6080.

[36] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22 (57) :1–64, 2021.

[37] L. Gresele, G. Fissore, A. Javaloy, B. Schölkopf, and A. Hyvärinen. Relative gradient optimization of the jacobian term in unsupervised deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[38] M. Gutmann and J. Hirayama. Bregman divergence as general framework to estimate unnormalized statistical models. In *Uncertainty in Artificial Intelligence (UAI)*, 2011.

[39] C. Ceylan and M.U. Gutmann. Conditional noise-contrastive estimation of unnormalised models. In *International Conference on Machine Learning (ICML)*, volume 80, pages 726–734. PMLR, 2018.

[40] A. Hinrichs, J. Prochno, and M. Ullrich. The curse of dimensionality for numerical integration on general domains. *J. Complex.*, 50 :25–42, 2018.

[41] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24) :695–709, 2005.

[42] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models : Consistency and statistical efficiency. In *EMNLP*, 2018.

[43] G. Roeder, L. Metz, and D. Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning (ICML)*, volume 139, pages 9030–9039. PMLR, 2021.

[44] A. Srivastava, S. Han, K. Xu, B. Rhodes, and M. U. Gutmann. Estimating the density ratio between distributions with high discrepancy using multinomial logistic regression. *Transactions on Machine Learning Research (TMLR)*, 2023.

[45] A. Hyvärinen and E. Oja. Independent component analysis : algorithms and applications. *Neural Networks*, 13(4) :411–430, 2000.

[46] G. Deco and W. Brauer. Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures. *Neural Networks*, 8 :525–535, 1995.

[47] L. Gresele, G. Fissore, A. Javaloy, B. Schölkopf, and A. Hyvärinen. Relative gradient optimization of the jacobian term in unsupervised deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[48] A. Mnih and Y.W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *International Conference on Machine Learning (ICML)*, page 419–426. Omnipress, 2012.

[49] O. Thomas, R. Dutta, J. Corander, S. Kaski, and M. U. Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2016.

[50] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162 4 :2025–35, 2002.

[51] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences (PNAS)*, 117(48) :30055–30062, 2020.

[52] C. Durkan, I. Murray, and G. Papamakarios. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning (ICML)*, volume 119, pages 2771–2781. PMLR, 2020.

[53] B.K. Miller, C. Weniger, and P. Forré. Contrastive neural ratio estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 3262–3278. Curran Associates, Inc., 2022.

[54] A.W. van der Vaart. *Asymptotic Statistics*. Asymptotic Statistics. Cambridge University Press, 2000.

[55] A. van der Vaart and J.A. Wellner. *Weak Convergence and Empirical Processes : With Applications to Statistics*. Springer Series in Statistics. Springer, 1996. ISBN 9780387946405.

[56] F. Bach. *Learning theory from first principles*. Arxiv (last accessed on Nov 22 2023), 2020.

[57] L. Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387251456.

[58] M. Celentano, C. Cheng, and A. Montanari. The high-dimensional asymptotics of first order methods with random data, 2021.

[59] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma. Explaining neural scaling laws, 2021.

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

[61] B. Rhodes, K. Xu, and M.U. Gutmann. Telescoping density-ratio estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 4905–4916. Curran Associates, Inc., 2020.

[62] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[63] M. Pihlaja, M. Gutmann, and A. Hyvärinen. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Uncertainty in Artificial Intelligence (UAI)*, 2010.

[64] P. Dayan and L.F. Abbott. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, 2005. ISBN 9780262541855.

[65] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. *IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

[66] *Sensory coding : information maximization and redundancy reduction*, volume 7 of *World Scientific Series in Mathematical Biology and Medecine*, 1999.

[67] J.-P. Nadal and N. Parga. Sensory coding : information maximization and redundancy reduction. http://www.lps.ens.fr/~nadal/documents/proceedings/carg97/carg97.html, 1999. [Online ; accessed 24-09-2023].

[68] F. A. C. Azevedo, L. R.B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513, 2009.

[69] G. Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006. ISBN 9780198041252.

[70] J-R King, L Gwilliams, C Holdgraf, J Sassenhagen, A Barachant, D Engemann, E Larson, and A Gramfort. Encoding and decoding framework to uncover the algorithms of cognition. *The Cognitive Neurosciences VI, MIT Press*, 6 :691–702, 2020.

[71] Rufin VanRullen. Perceptual cycles. *Trends in cognitive sciences*, 20(10) :723–735, 2016.

[72] R. A. Poldrack, J. A. Mumford, and T. E. Nichols. *Handbook of Functional MRI Data Analysis*. Cambridge University Press, 2011.

[73] Hämäläinen, M., R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.*, 65 :413–497, Apr 1993.

[74] E. Niedermeyer and F.H.L. da Silva. *Electroencephalography : Basic Principles, Clinical Applications, and Related Fields*. LWW Doody's all reviewed collection. Lippincott Williams & Wilkins, 2005. ISBN 9780781751261.

[75] H. Berger. *Hans Berger on the Electroencephalogram of Man : The Fourteen Original Reports on the Human Electroencephalogram*. EEG Journals Supplement. Elsevier Publishing Company, 1969. ISBN 9780444407399.

[76] R. Hari and A. Puce. *MEG-EEG Primer*. Oxford University Press, 2017. ISBN 9780190497774.

[77] D. Cohen. Magnetoencephalography : Evidence of magnetic fields produced by alpha-rhythm currents. *Science*, 161 :784–786, 1968.

[78] C. Pernet, M. Garrido, A. Gramfort, N. Maurits, C. Michel, E. Pang, R. Salmelin, J.-M. Schoffelen, P. Valdés-Sosa, and A. Puce. Issues and recommendations from the OHBM COBIDAS MEEG committee for reproducible EEG and MEG research. *Nature Neuroscience*, 23 :1473–1483, 2020.

[79] S. A. Imtiaz. A systematic review of sensing technologies for wearable sleep staging. *Sensors (Basel, Switzerland)*, 21, 2021.

[80] A. Défossez, C. Caucheteux, J. Rapin, O. Kabeli, and J.-R. King. Decoding speech from non-invasive brain recordings, 2022.

[81] Grand View Research. *Magnetoencephalography Market Size, Share & Trends Analysis Report By Application (Clinical, Research), By End Use (Hospitals, Imaging Centers, Academic & Research Institutes), By Region, And Segment Forecasts, 2023 - 2030.* Grand View Research, 2021.

[82] S. Baillet. Magnetoencephalography for brain electrophysiology and imaging. *Nature Neuroscience*, 20 :327–339, 2017.

[83] T.C Ferree, M.T Clay, and D.M Tucker. The spatial resolution of scalp EEG. *Neurocomputing*, 38-40 :1209–1216, 2001. ISSN 0925-2312. Computational Neuroscience : Trends in Research 2001.

[84] B. C. Jobst, F. Bartolomei, B. Diehl, B. Frauscher, P. Kahane, L. Minotti, A. Sharan, N. Tardy, G. Worrell, and J. Gotman. Intracranial EEG in the 21st century. *Epilepsy Currents*, 20(4) :180–188, July 2020.

[85] D. Ulrich. Sleep spindles as facilitators of memory formation and learning. *Neural Plasticity*, 2016, 2016.

[86] Christina Jayne Bathgate and Jack D Edinger. Diagnostic criteria and assessment of sleep disorders. In *Handbook of Sleep Disorders in Medical Conditions*, pages 3–25. Elsevier, 2019.

[87] SJM Smith. EEG in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl 2) :ii2–ii7, 2005.

[88] Christina Micanovic and Suvankar Pal. The diagnostic utility of EEG in early-onset dementia : a systematic review of the literature with narrative analysis. *Journal of Neural Transmission*, 121(1) : 59–69, 2014.

[89] J. Brogger, T. Eichele, E. Aanestad, H. Olberg, I. Hjelland, and H. Aurlien. Visual EEG reviewing times with SCORE EEG. *Clinical Neurophysiology Practice*, 3 :59–64, 2018.

[90] O. Chehab, A. Hyvärinen, and A. Risteski. Provable benefits of annealing for computing normalizing constants. In *Submitted*, 2023.

[91] M. Uehara, T. Matsuda, and F. Komaki. Analysis of noise contrastive estimation from the perspective of asymptotic variance. *ArXiv*, 2018. doi: 10.48550/ARXIV.1808.07983.

[92] O. Chehab, A. Gramfort, and A. Hyvärinen. Optimizing the noise in self-supervised learning : from importance sampling to noise-contrastive estimation, 2023.

[93] Eric Maris and Robert Oostenveld. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, 164(1) :177–190, 2007. ISSN 0165-0270.

[94] Thomas Naselaris, Kendrick N Kay, Shinji Nishimoto, and Jack L Gallant. Encoding and decoding in fMRI. *Neuroimage*, 56(2) :400–410, 2011.

[95] Shivangi Mahto, Vy A. Vo, Javier Turek, and Alexander G. Huth. Multi-timescale representation learning in lstm language models. *ArXiv*, abs/2009.12727, 2021.

[96] Menoua Keshishian, Hassan Akbari, Bahar Khalighinejad, Jose L Herrero, Ashesh D Mehta, and Nima Mesgarani. Estimating and interpreting nonlinear receptive field of sensory neural responses with deep neural network models. *eLife*, 9 :e53445, jun 2020. ISSN 2050-084X.

[97] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3) :356, 2016.

[98] Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1) :1–10, 2022.

[99] Juliette Millet and Jean-Remi King. Inductive biases, pretraining and fine-tuning jointly account for brain responses to speech. *arXiv preprint arXiv :2103.01032*, 2021.

[100] Biljana Petreska, Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. In *Advances in Neural Information Processing Systems 24 : 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 756–764, 2011.

[101] Lea Duncker, Gergo Bohner, Julien Boussard, and Maneesh Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1726–1734. PMLR, 09–15 Jun 2019.

[102] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[103] Adam H. Marblestone, Greg Wayne, and Konrad Paul Kording. Towards an integration of deep learning and neuroscience. *bioRxiv*, 2016.

[104] Katja Seeliger, Matthias Fritsche, Umut Güçlü, Sanne Schoenmakers, J-M Schoffelen, Sander E Bosch, and MAJ Van Gerven. Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, 180 :253–266, 2018.

[105] Katja Seeliger, Luca Ambrogioni, Yağmur Güçlütürk, Leonieke M van den Bulk, Umut Güçlü, and Marcel AJ van Gerven. End-to-end neural system identification with neural information flow. *PLOS Computational Biology*, 17(2) :e1008558, 2021.

[106] A. M. H. J. Aertsen and P. I. M. Johannesma. The spectro-temporal receptive field. *Biological Cybernetics*, 42(2) :133–143, Nov 1981.

[107] N.J. Smith and M. Kutas. Regression-based estimation of ERP waveforms : I. the rERP framework. *Psychophysiology*, 52(2) :157–168, 2015.

[108] N.J. Smith and M. Kutas. Regression-based estimation of ERP waveforms : Ii. nonlinear effects, overlap correction, and practical considerations. *Psychophysiology*, 52(2) :169–181, 2015.

[109] Christopher R. Holdgraf, Jochem W. Rieger, Cristiano Micheli, Stephanie Martin, Robert T. Knight, and Frederic E. Theunissen. Encoding and decoding models in cognitive electrophysiology. *Frontiers in Systems Neuroscience*, 11 :61, 2017.

[110] Michael J. Crosse, Giovanni M. Di Liberto, Adam Bednar, and Edmund C. Lalor. The multivariate temporal response function (mTRF) toolbox : A matlab toolbox for relating neural signals to continuous stimuli. *Frontiers in Human Neuroscience*, 10 :604, 2016.

[111] Nuno R. Gonçalves, Robert Whelan, John J. Foxe, and Edmund C. Lalor. Towards obtaining spatiotemporally precise responses to continuous sensory stimuli in humans : A general linear modeling approach to EEG. *NeuroImage*, 97 :196–205, 2014.

[112] Stéphanie Martin, Peter Brunner, Chris Holdgraf, Hans-Jochen Heinze, Nathan E. Crone, Jochem Rieger, Gerwin Schalk, Robert T. Knight, and Brian N. Pasley. Decoding spectrotemporal features of overt and covert speech from the human cortex. *Frontiers in Neuroengineering*, 7 :14, 2014.

[113] Edmund Lalor and John Foxe. Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution. *European Journal of Neuroscience*, 31(1) :189–193, 2009.

[114] Nai Ding and Jonathan Z. Simon. Adaptive temporal encoding leads to a background-insensitive cortical representation of speech. *Journal of Neuroscience*, 33(13) :5728–5735, 2013.

[115] Jona Sassenhagen. How to analyse electrophysiological responses to naturalistic language with time-resolved multiple regression. *Language, Cognition and Neuroscience*, 34(4) :474–490, 2019.

[116] Jean-Baptiste Poline and Matthew Brett. The general linear model and fmri : Does love last forever ? *NeuroImage*, 62(2) :871–880, 2012. ISSN 1053-8119. 20 YEARS OF fMRI.

[117] Jeff B Cromwell and Michel Terraza. *Multivariate tests for time series models*. Sage, 1994.

[118] Yasuki Noguchi, Koji Inui, and Ryusuke Kakigi. Temporal dynamics of neural adaptation effect in the human visual ventral stream. *Journal of Neuroscience*, 24(28) :6283–6290, 2004. ISSN 0270-6474.

[119] John Foxe and Adam Snyder. The role of alpha-band brain oscillations as a sensory suppression mechanism during selective attention. *Frontiers in Psychology*, 2 :154, 2011.

[120] Iñaki Iturrate, Ricardo Chavarriaga, Luis Montesano, Javier Minguez, and JdR Millán. Latency correction of event-related potentials between different experimental protocols. *Journal of neural engineering*, 11(3) :036005, 2014.

[121] Alexander G Huth, Wendy A De Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532 (7600) :453–458, 2016.

[122] Ricardo Vigário, Veikko Jousmäki, Matti Hämäläinen, Riitta Hari, and Erkki Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 229–235. MIT Press, 1998.

[123] Mikko Uusitalo and Risto Ilmoniemi. Signal-space projection method for separating MEG or EEG into components. *Medical and biological engineering and computing*, 35(2) :135–40, 04 1997.

[124] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4) :920–928, 2011.

[125] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A. Engemann. Predictive regression modeling with MEG/EEG : from source power to signals and cognitive states. *NeuroImage*, 222 :116893, 2020.

[126] Lucas C. Parra, Clay D. Spence, Adam D. Gerson, and Paul Sajda. Recipes for the linear analysis of EEG. *NeuroImage*, 28(2) :326–341, 2005.

[127] Alain de Cheveigné, Giovanni M Di Liberto, Dorothée Arzounian, Daniel DE Wong, Jens Hjortkjær, Søren Fuglsang, and Lucas C Parra. Multiway canonical correlation analysis of brain data. *NeuroImage*, 186 :728–740, 2019.

[128] Hao Xu, Alexander Lorbert, Peter J Ramadge, J Swaroop Guntupalli, and James V Haxby. Regularized hyperalignment of multi-set fMRI data. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 229–232. IEEE, 2012.

[129] Michael Lim, Justin Ales, Benoit Cottereau, Trevor Hastie, and Anthony Norcia. Sparse EEG/MEG source estimation via a group lasso. *PLOS ONE*, 12 :e0176835, 06 2017.

[130] Jean-Rémi King, François Charton, David Lopez-Paz, and Maxime Oquab. Back-to-back regression : Disentangling the influence of correlated factors from multivariate observations. *NeuroImage*, 220 :117028, 2020.

[131] Thomas Bazeille, Elizabeth Dupre, Hugo Richard, J B Poline, and Bertrand Thirion. An empirical evaluation of functional alignment using inter-subject decoding. *Neuroimage*, 2020.

[132] H. Richard, L. Gresele, A. Hyvärinen, B. Thirion, A. Gramfort, and P. Ablin. Modeling shared responses in neuroimaging studies through multiview ica. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19149–19162. Curran Associates, Inc., 2020.

[133] Russell A Poldrack, Paul C Fletcher, Richard N Henson, Keith J Worsley, Matthew Brett, and Thomas E Nichols. Guidelines for reporting an fMRI study. *Neuroimage*, 40(2) :409–414, 2008.

[134] Ari S. Benjamin, Hugo L. Fernandes, Tucker Tomlinson, Pavan Ramkumar, Chris VerSteeg, Raeed H. Chowdhury, Lee E. Miller, and Konrad P. Kording. Modern machine learning as a benchmark for fitting neural responses. *Frontiers in Computational Neuroscience*, 12 :56, 2018.

[135] Anna A Ivanova, Martin Schrimpf, Stefano Anzellotti, Noga Zaslavsky, Evelina Fedorenko, and Leyla Isik. Is it that simple ? linear mapping models in cognitive neuroscience. *bioRxiv*, 2021.

[136] Gareth James. *An introduction to statistical learning : with applications in R*. Springer, New York, NY, 2013. ISBN 978-1-4614-7138-7.

[137] Matti Hämäläinen, Riitta Hari, Risto J Ilmoniemi, Jukka Knuutila, and Olli V Lounasmaa. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2) :413, 1993.

[138] Hugh R Wilson and Jack D Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, 12(1) :1–24, 1972.

[139] K J Friston, L Harrison, and W Penny. Dynamic causal modelling. *Neuroimage*, 19(4) :1273–1302, 2003.

[140] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2) :223–311, 2018.

[141] Stephen A. Billings. *Nonlinear System Identification : NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley, 2013.

[142] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) : 1735–1780, 1997.

[143] Alexandre Défossez, Neil Zeghidour, Nicolas Usunier, Leon Bottou, and Francis Bach. Sing : Symbol-to-instrument neural generator. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9041–9051. Curran Associates, Inc., 2018.

[144] Jonathan Masci, Ueli Meier, Dan C. Ciresan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, 2011.

[145] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786) :504–507, 2006.

[146] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1863–1871, Bejing, China, 22–24 Jun 2014. PMLR.

[147] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020.

[148] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 1171–1179. Curran Associates, Inc., 2015.

[149] T. P. Jung, S. Makeig, M. J. McKeown, A. J. Bell, T. W. Lee, and T. J. Sejnowski. Imaging brain dynamics using independent component analysis. *Proceedings of the IEEE*, 89(7) :1107–1122, 2001.

[150] Alain de Cheveigné, Daniel DE Wong, Giovanni M Di Liberto, Jens Hjortkjaer, Malcolm Slaney, and Edmund Lalor. Decoding the auditory brain with canonical component analysis. *NeuroImage*, 172 :206–216, 2018.

[151] Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452 :352, 2008.

[152] Umut Güçlü and M. V. Gerven. Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in Computational Neuroscience*, 11, 2017.

[153] Leo Breiman. Random forests. *Machine Learning*, 45(1) :5–32, Oct 2001.

[154] Diederik Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[155] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch : An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019.

[156] Po-Hsuan (Cameron) Chen, Janice Chen, Yaara Yeshurun, Uri Hasson, James Haxby, and Peter J Ramadge. A reduced-dimension fmri shared response model. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[157] Hejia Zhang, Po-Hsuan Chen, and Peter Ramadge. Transfer learning on fmri datasets. In *International Conference on Artificial Intelligence and Statistics*, pages 595–603. PMLR, 2018.

[158] James V. Haxby, J. Swaroop Guntupalli, Andrew C. Connolly, Yaroslav O. Halchenko, Bryan R. Conroy, Maria Ida Gobbini, Michael Hanke, and Peter J. Ramadge. A common, high-dimensional model of the representational space in human ventral temporal cortex. *Neuron*, 72, 2011.

[159] Maneesh Sahani and Jennifer F. Linden. How linear are auditory cortical responses? In *NIPS*, 2002.

[160] Kendrick Norris Kay, Jonathan A. Winawer, Aviv A. Mezer, and Brian A. Wandell. Compressive spatial summation in human visual cortex. *Journal of neurophysiology*, 110 2 :481–94, 2013.

[161] Jan-Mathijs Schoffelen, Robert Oostenveld, Nietzsche Lam, Julia Udden, Annika Hulten, and Peter Hagoort. A 204-subject multimodal neuroimaging dataset to study language processing. *Scientific Data*, 6, 12 2019.

[162] Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. Luminosoinsight/wordfreq : v2.2. https ://github.com/LuminosoInsight/wordfreq, October 2018.

[163] Catherine Tallon-Baudry and Olivier Bertrand. Oscillatory gamma activity in humans and its role in object representation. *Trends in Cognitive Sciences*, 3(4) :151–162, 1999.

[164] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, et al. MEG and EEG data analysis with MNE-Python. *Frontiers in neuroscience*, 7 :267, 2013.

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

[166] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[167] Saskia Haegens and Elana Zion Golumbic. Rhythmic facilitation of sensory processing : A critical review. *Neuroscience & Biobehavioral Reviews*, 86 :150–165, 2018.

[168] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3) :31–57, 2018.

[169] Anna Ivanova, Martin Schrimpf, Leyla Isik, Stefano Anzellotti, Noga Zaslavsky, and Evelina Fedorenko. Is it that simple ? the use of linear models in cognitive neuroscience. Workshop Proposal, 2020.

[170] Kara D. Federmeier and Marta Kutas. A rose by any other name : Long-term memory structure and sentence processing. *Journal of Memory and Language*, 41(4) :469–495, 11 1999.

[171] Evelina Fedorenko, Terri L Scott, Peter Brunner, William G Coon, Brianna Pritchett, Gerwin Schalk, and Nancy Kanwisher. Neural correlate of the construction of sentence meaning. *Proceedings of the National Academy of Sciences*, 113(41) :E6256–E6262, 2016.

[172] Tong He, Ru Kong, Avram J Holmes, Minh Nguyen, Mert R Sabuncu, Simon B Eickhoff, Danilo Bzdok, Jiashi Feng, and BT Thomas Yeo. Deep neural networks and kernel regression achieve comparable accuracies for functional connectivity prediction of behavior and demographics. *NeuroImage*, 206 :116276, 2020.

[173] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T.H. Falk, and J. Faubert. Deep learning-based electroencephalography analysis : a systematic review. *Journal of neural engineering*, 16(5) : 051001, 2019.

[174] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11) :1761–1770, 2019.

[175] Umut Güçlü and Marcel van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, 35 :10005–10014, 07 2015.

[176] Ilya Kuzovkin, Raul Vicente, Mathilde Petton, Jean-Philippe Lachaux, Monica Baciu, Philippe Kahane, Sylvain Rheims, Juan Vidal, and Jaan Aru. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications Biology*, 1, 12 2018.

[177] Santiago A. Cadena, George H Denfield, Edgar Y. Walker, Leon A. Gatys, A. Tolias, M. Bethge, and Alexander S. Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS Computational Biology*, 15, 2019.

[178] Michael Eickenberg, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion. Seeing it all : Convolutional network layers map the function of the human visual system. *NeuroImage*, 152 : 184–194, 2017.

[179] D. Klindt, A. S. Ecker, T. Euler, and M. Bethge. Neural system identification for large populations separating "what" and "where". In *Advances in Neural Information Processing Systems 31*, Sep 2017.

[180] Chengxu Zhuang, Siming Yan, Aran Nayebi, Martin Schrimpf, Michael C. Frank, James J. DiCarlo, and Daniel L. K. Yamins. Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3), 2021.

[181] Shailee Jain and Alexander Huth. Incorporating context into language encoding models for fmri. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6629–6638, 2018.

[182] Nikolaus Kriegeskorte and Tal Golan. Neural network models and deep learning. *Current Biology*, 29(7) :R231–R236, 2019.

[183] P. Sun, G. K. Anumanchipalli, and E. Chang. Brain2char : A deep architecture for decoding text from brain recordings. *Journal of Neural Engineering*, 2020.

[184] Yağmur Güçlütürk, Umut Güçlü, Katja Seeliger, Sander Bosch, Rob van Lier, and Marcel van Gerven. Deep adversarial neural decoding. arXiv Preprint 1705.07109, 2017.

[185] Ana Luísa Pinho, Alexis Amadon, Torsten Ruest, Murielle Fabre, Elvis Dohmatob, Isabelle Denghien, Chantal Ginisty, Séverine-Becuwe, Séverine Roger, Laurence Laurier, Véronique Joly-Testault, Gaëlle Médiouni-Cloarec, Christine Doublé, Bernadette Martins, Philippe Pinel, Evelyn Eger, Gaël Varoquaux, Christophe Pallier, Stanislas Dehaene, Lucie Hertz-Pannier, and Bertrand Thirion. Individual Brain Charting, a high-resolution fMRI dataset for cognitive mapping. *Sci Data*, 5 :180105, 2018.

[186] Charlotte Caucheteux and Jean-Rémi King. Language processing in brains and deep neural networks : computational convergence and its limits. *bioRxiv*, 2020.

[187] Martin Schrimpf, Idan Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua Tenenbaum, and Evelina Fedorenko. Artificial neural networks accurately predict language processing in the brain. *bioRxiv*, 2020.

[188] Mariya Toneva and Leila Wehbe. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[189] Luca Ambrogioni, Marcel A. J. van Gerven, and Eric Maris. Dynamic decomposition of spatio-temporal neural signals. *PLOS Computational Biology*, 13(5) :1–37, 05 2017.

[190] Diego Vidaurre, Andrew J. Quinn, Adam P. Baker, David Dupret, Alvaro Tejero-Cantero, and Mark W. Woolrich. Spectrally resolved fast transient brain states in electrophysiological data. *NeuroImage*, 126 :81–95, 2016. ISSN 1053-8119.

[191] Marcel A.J. van Gerven. A primer on encoding models in sensory neuroscience. *Journal of Mathematical Psychology*, 76 :172–183, 2017. Model-based Cognitive Neuroscience.

[192] Sanne Schoenmakers, Markus Barth, Tom Heskes, and Marcel van Gerven. Linear reconstruction of perceived images from human brain activity. *NeuroImage*, 83 :951–961, 2013.

[193] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86, 10 2013.

[194] John D Hunter. Matplotlib : A 2d graphics environment. *Computing in science & engineering*, 9 (3) :90–95, 2007.

[195] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernandez del Rio, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825) :357–362, September 2020.

[196] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy

1.0 Contributors. SciPy 1.0 : Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17 :261–272, 2020.

[197] H. Banville, O. Chehab, A. Hyvärinen, D.-A. Engemann, and A. Gramfort. Uncovering the structure of clinical EEG signals with self-supervised learning. *Journal of Neural Engineering*, 18(4) : 046020, 2021.

[198] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, aug 2017. ISSN 1097-0193. doi: 10.1002/hbm.23730. URL http://dx.doi.org/10.1002/hbm.23730.

[199] David Sabbagh, Pierre Ablin, Gael Varoquaux, Alexandre Gramfort, and Denis A. Engemann. Manifold-regression to predict from MEG/EEG brain signals without source modeling. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[200] Suzanna Becker. Learning to categorize objects using temporal coherence. In *Advances in neural information processing systems*, pages 361–368, 1993.

[201] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis : Unsupervised learning of invariances. *Neural Computation*, 14(4) :715–770, 2002.

[202] Bruce M Altevogt and Harvey R Colten. *Sleep disorders and sleep deprivation : an unmet public health problem*. National Academies Press, 2006.

[203] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[204] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn : unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016.

[205] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, pages 4650–4661, 2019.

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

[207] Mohammad M Ghassemi, Benjamin E Moody, Li-Wei H Lehman, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win : the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[208] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet : components of a new research resource for complex physiologic signals. *Circulation*, 101(23) :e215–e220, 2000.

[209] Silvia López, I Obeid, and J Picone. Automated interpretation of abnormal adult electroencephalograms. *MS Thesis, Temple University*, 2017.

[210] Stanislas Chambon, Mathieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4) : 758–769, 2018.

[211] Shayan Motamedi-Fakhr, Mohamed Moshrefi-Torbati, Martyn Hill, Catherine M. Hill, and Paul R. White. Signal processing techniques applied to human sleep EEG signals—a review. *Biomedical Signal Processing and Control*, 10 :21–33, 2014. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2013.12.003. URL http://www.sciencedirect.com/science/article/pii/S174680941300178X.

[212] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis : a systematic review. *Journal of Neural Engineering*, 16(5) :051001, 2019.

[213] Magdy Younes, Jill Raneri, and Patrick Hanly. Staging sleep in polysomnograms : Analysis of inter-scorer variability. *Journal of Clinical Sleep Medicine*, 12(06) :885–894, 2016. doi: 10.5664/jcsm.5894. URL https://jcsm.aasm.org/doi/abs/10.5664/jcsm.5894.

[214] Raman K Malhotra and Alon Y Avidan. Sleep stages and scoring technique. *Atlas of Sleep Medicine*, pages 77–99, 2013.

[215] S Lopez, G Suarez, D Jungreis, I Obeid, and J Picone. Automated identification of abnormal adult EEGs. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–5. IEEE, 2015.

[216] Robin Tibor Schirrmeister, Lukas Gemein, Katharina Eggensperger, Frank Hutter, and Tonio Ball. Deep learning with convolutional neural networks for decoding and visualization of EEG pathology. *arXiv preprint arXiv :1708.08012*, 2017.

[217] Lukas AW Gemein, Robin T Schirrmeister, Patryk Chrabąszcz, Daniel Wilson, Joschka Boedecker, Andreas Schulze-Bonhage, Frank Hutter, and Tonio Ball. Machine-learning-based diagnostics of EEG pathology. *Neuroimage*, page 117021, 2020.

[218] D. Kobak and G. C. Linderman. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature Biotechnology*, 39 :156–157, 2021.

[219] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[220] A. Mnih and Y.W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *International Conference on Machine Learning (ICML)*, 2012.

[221] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[222] A. Menon and C.S. Ong. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning (ICML)*, 2016.

[223] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, 2014.

[224] L. Riou-Durand and N. Chopin. Noise contrastive estimation : Asymptotic properties, formal comparison with MC-MLE. *Electronic Journal of Statistics*, 12(2) :3473 – 3518, 2018.

[225] R. Gao, E. Nijkamp, D.P. Kingma, Z. Xu, A.M. Dai, and Y. Nian Wu. Flow contrastive estimation of energy-based models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7515–7525, 2020.

[226] A.B. Dieng, F.J.R. Ruiz, D.M. Blei, and M.K. Titsias. Prescribed generative adversarial networks. *ArXiv*, abs/1910.04302, 2019.

[227] Jonathan P. Lorraine, David Acuna, Paul Vicol, and David Duvenaud. Complex momentum for optimization in games. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[228] S. Mohamed and B. Lakshminarayanan. Learning in implicit generative models. *ArXiv*, abs/1610.03483, 2016.

[229] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, J.K. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0 : Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17 :261–272, 2020.

[230] J.D. Hunter. Matplotlib : A 2d graphics environment. *Computing in science & engineering*, 9(3) : 90–95, 2007.

[231] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane,

J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T.E. Oliphant. Array programming with NumPy. *Nature*, 585 (7825) :357–362, 2020.

[232] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021.

[233] G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8) :1771–1800, 2002.

[234] A.B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[235] B. Liu, E. Rosenfeld, P. Ravikumar, and A. Risteski. Analyzing and improving the optimization landscape of noise-contrastive estimation. In *International Conference on Learning Representations (ICLR)*, 2022.

[236] O. Chehab, A. Gramfort, and A. Hyvärinen. The optimal noise in noise-contrastive learning is not what you think. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 180, pages 307–316. PMLR, 2022.

[237] R.M. Neal. Annealed importance sampling. *Statistics and Computing*, 11 :125–139, 1998.

[238] C. Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78 : 2690–2693, 1996.

[239] C. Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements : A master-equation approach. *Physical Review E*, 56 :5018–5035, 1997.

[240] R. Salakhutdinov and G. Hinton. Replicated softmax : an undirected topic model. In *Neural Information Processing Systems (NIPS)*, 2009.

[241] Y. Dauphin and Y. Bengio. Stochastic ratio matching of rbms for sparse high-dimensional inputs. In *Neural Information Processing Systems (NIPS)*, volume 26. Curran Associates, Inc., 2013.

[242] V. Masrani, T.A. Le, and F.D. Wood. The thermodynamic variational objective. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[243] R. Brekelmans, S. Huang, M. Ghassemi, G. Ver Steeg, R.B. Grosse, and A. Makhzani. Improving mutual information estimation with annealed and energy-based bounds. In *International Conference on Learning Representations (ICLR)*, 2022.

[244] A. Gelman and X.-L. Meng. Simulating normalizing constants : From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13 :163–185, 1998.

[245] R. Grosse, C. Maddison, and R. Salakhutdinov. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26. Curran Associates, Inc., 2013.

[246] G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation : Umbrella sampling. *Journal of Computational Physics*, 23(2) :187–199, 1977.

[247] X.-L. Meng and W.H. Wong. Simulating ratios of normalizing constants via a simple identity : a theoretical exploration. *Statistica Sinica*, pages 831–860, 1996.

[248] V. Masrani, R. Brekelmans, T. Bui, F. Nielsen, A. Galstyan, G. Ver Steeg, and F. Wood. q-paths : Generalizing the geometric annealing path using power means. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 161, pages 1938–1947. PMLR, 27–30 Jul 2021.

[249] M.A. Newton. Approximate bayesian-inference with the weighted likelihood bootstrap. *Journal of the royal statistical society series b-methodological*, 1994.

[250] M.-H. Chen and Q.-M. Shao. On monte carlo methods for estimating ratios of normalizing constants. *The Annals of Statistics*, 25(4) :1563–1594, 1997.

[251] Q. Liu, J. Peng, A.T. Ihler, and J.W. Fisher III. Estimating the partition function by discriminance sampling. In *Uncertainty in Artificial Intelligence (UAI)*, 2015.

[252] A. Hyvärinen, J. Hurri, and P.O. Hoyer. Natural image statistics - a probabilistic approach to early computational vision. In *Computational Imaging and Vision*, 2009.

[253] O. Krause, A. Fischer, and C. Igel. Algorithms for estimating the partition function of restricted boltzmann machines. *Artificial Intelligence*, 278 :103195, 2020.

[254] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *arXiv : Probability*, 2015.

[255] B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar. Density estimation in infinite dimensional exponential families. *Journal of Machine Learning Research*, 18(57) :1–59, 2017.

[256] O. Krause, A. Fischer, T. Glasmachers, and C. Igel. Approximation properties of dbns with binary hidden units and real-valued visible units. In *International Conference on Machine Learning (ICML)*, 2013.

[257] F. Nielsen and V. Garcia. Statistical exponential families : A digest with flash cards, 2011.

[258] X.-L. Meng and S. Schilling. Fitting full-information item factor models and an empirical investigation of bridge sampling. *Journal of the American Statistical Association*, 91(435) :1254–1267, 1996.

[259] M. Uehara, T. Kanamori, T. Takenouchi, and T. Matsuda. A unified statistically efficient estimation framework for unnormalized models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108, pages 809–819. PMLR, 2020.

[260] L. Ellam, H. Strathmann, M.A. Girolami, and I. Murray. A determinant-free method to simulate the parameters of large gaussian fields. *Stat*, 6 :271–281, 2017.

[261] T. Kiwaki. Variational optimization of annealing schedules, 2015.

[262] R. Brekelmans, V. Masrani, F.D. Wood, G. Ver Steeg, and A.G. Galstyan. All in the exponential family : Bregman duality in thermodynamic variational inference. In *International Conference on Machine Learning (ICML)*, 2020.

[263] S. Goshtasbpour, V. Cohen, and F. Perez-Cruz. Adaptive annealed importance sampling with constant rate progress. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 11642–11658. PMLR, 2023.

[264] N. Chopin and O. Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer International Publishing, 2020. ISBN 9783030478452.

[265] P. Del Moral and A. Doucet. Sequential monte carlo samplers. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68, 2002.

[266] C. Dai, J. Heng, P.E. Jacob, and N. Whiteley. An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, 117 :1587–1600, 2020.

[267] S. Syed, V. Romaniello, T. Campbell, and A. Bouchard-Cote. Parallel tempering on optimized paths. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 10033–10042. PMLR, 2021.

[268] G. Behrens, N. Friel, and M. Hurn. Tuning tempered transitions. *Statistics and Computing*, 22 : 65–78, 2010.

[269] L. Wang, D.E. Jones, and X.-L. Meng. Warp bridge sampling : The next generation. *Journal of the American Statistical Association*, 117(538) :835–851, 2022.

[270] H. Xing. Improving bridge estimators via f-GAN. *Statistics and Computing*, 32, 2022.

[271] Y. Polyanskiy and Y. Wu. *Information Theory : From Coding to Learning*. Cambridge University Press, 2022.

[272] S.-I. Amari and H. Nagaoka. *Methods of information geometry*. AMS, 2000.

[273] E. Vernet, M.D. Reid, and R.C. Williamson. Composite multiclass losses. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Neural Information Processing Systems (NIPS)*, volume 24. Curran Associates, Inc., 2011.

[274] R. Nock, A. Menon, and C.S. Ong. A scaled bregman theorem with applications. In *Neural Information Processing Systems (NIPS)*, volume 29. Curran Associates, Inc., 2016.

[275] Arindam Banerjee, Xin Guo, and Hui Wang. On the optimality of conditional expectation as a bregman predictor. *IEEE Trans. Inf. Theory*, 51 :2664–2669, 2005.

[276] B. Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106 :1602–1614, 2011.

[277] A. Hyvärinen. Estimation theory and information geometry based on denoising. In *Proceedings of the First Workshop on Information Theoretic Methods in Science and Engineering, August 18-20, 2008, Tampere, Finland*, 2008.

[278] F. Koehler, A. Heckett, and A. Risteski. Statistical efficiency of score matching : The view from isoperimetry. In *International Conference on Learning Representations (ICLR)*, 2023.

[279] Y. Qin and A. Risteski. Fit like you sample : Sample-efficient generalized score matching from fast mixing markov chains, 2023.

[280] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel L. K. Yamins, and Noah D. Goodman. On mutual information in contrastive learning for visual representations. *ArXiv*, abs/2005.13149, 2020.

[281] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. *arXiv : Learning*, 2021.

[282] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *ArXiv*, abs/1907.13625, 2020.

[283] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning (ICML)*, 2020.

[284] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning (ICML)*, 2021.