*Regular Article*

# Fisher Information Estimation using Neural Networks

**Tran Trong Duy**[1], **Nguyen Van Ly**[1,2], **Nguyen Linh Trung**[1], **Karim Abed-Meraim**[3,4]

[1] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[2] University of California, Irvine, CA, USA
[3] PRISME Laboratory, University of Orléans, Orléans, France
[4] Academic Institute of France (IUF), 1 rue Descartes, 75005 Paris, France

Correspondence: Nguyen Linh Trung, linhtrung@vnu.edu.vn

Communication: received 15 July 2022, revised 11 December 2022, accepted 9 May 2023
Online publication: 19 June 2023, Digital Object Identifier: 10.21553/rev-jec.322
The associate editor coordinating the review of this article and recommending it for publication was Prof. Vo Nguyen Quoc Bao.

*Abstract*– **In estimation theory, the Fisher information matrix (FIM) is a fundamental concept from which we can infer the well-known Cramér-Rao bound. A closed-form expression of the FIM is often intractable due to the lack or sophistication of statistical models. In this paper, we propose a Fisher Information Neural Estimator (FINE) based on neural networks and a relation between the $f$-divergence and the Fisher information. The proposed method produces an estimate of the FIM directly from observed data. It does not require knowledge or an estimate of the probability density function (pdf), and is therefore universally applicable. The proposed FINE is applicable for not only deterministic parameters but also random parameters. We show via numerical results that the proposed FINE can provide a highly-accurate FIM estimate with a low-computational complexity. Furthermore, we also propose an accelerated FINE version which can be used for scenarios with a high parameter dimension. Finally, we develop an algorithm to choose an appropriate size of the employed neural networks.**

*Keywords*– **Fisher information estimation, Cramér-Rao bound, neural networks.**

## 1 Introduction

Fisher information is a well-known and well-defined concept in mathematical statistics, which is defined as a measure of the amount of information that a random variable carries about some unknown parameters. In estimation theory, the inverse of the Fisher information directly gives us the Cramér-Rao bound (CRB), which is a well-known lower bound on the variance of any unbiased estimator of the unknown parameters. There are many other areas in which the Fisher information is applied to, e.g., Bayesian statistics, frequentist statistics, optimal experimental design, computational neuroscience, physical laws, biology, and machine learning [1–3].

Analytically, a closed-form expression of the Fisher information matrix (FIM) might be obtained by taking the expectation of the Hessian matrix of the log likelihood function (the score function). Unfortunately, such a straightforward computation is often impossible due to unknown statistical models [4, 5]. Even in circumstances where the statistical model is available, a closed-form expression of the FIM can still be intractable due to model sophistication. This difficulty raises the significance of developing FIM estimation methods. Donoho [6] showed that one cannot generally give a confidence interval for a functional of some unknown density without prior information. However, one can give a lower confidence bound, e.g., the Fisher information, for such functionals. Hence, the lower confidence bound can be used as an estimate of the true Fisher information where the confidence level is computed using the Kolmogorov distance. Another bound on the Fisher information is studied in [7]. The author used a spline interpolation to obtain the minimum Fisher information among all the distributions from which the observations were made.

The estimation of the FIM can be divided into two categories: plug-in and non-plug-in [8, 9]. In the plug-in category, the strategy is to first estimate the probability density function (pdf) based on the observed data and then use the pdf estimate for a numerical computation of the FIM. Plug-in estimators had not been available until the introduction of a kernel family of pdf estimators by Rosenblatt [10]. After that, kernel-based methods for estimating the derivative of the pdf or the functionals have been studied extensively in the literature [11–14]. Particularly, Bahattacharya proposed the first Fisher information estimator in [11] and introduced some error bounds on density and its derivative estimation. More recently, in [15], Spall proposed a Monte Carlo resampling-based (MCR) method for FIM estimation. The MCR method first estimates the pdf for each of a set of perturbed experiments and then numerically computes the gradient of the log density function before sample averaging. Another method for FIM estimation was introduced in [16], which also estimates the pdf using the observed data and then obtains the derivatives of the pdf based on finite-difference approximation. This method is based on an algorithm called Density Estimation using Field Theory, which suffers from "the curse of dimensionality", and the implementation is only suitable for a small dimension. Unlike the plug-in methods, the strategy of non-plug-in methods is to directly estimate the FIM based on the observed data. This non-plug-in strategy is particularly suitable for circumstances where the system is a black box whose operating parameters are tunable, e.g. controlled experiments [17–19]. One can observe data from the system for various settings of the parameters. An example of non-plug-in FIM estimation methods is in [20], which is based on a relation between the $f$-divergence and the FIM.

The strategy of plug-in methods is straightforward, but an accurate estimate of the pdf may not always be possible or is very difficult to obtain in scenarios where the underlying pdf is sophisticated. Non-plug-in methods do not rely on pdf estimation since the FIM is directly estimated from the observed data, and thus they are relieved of the difficulties in pdf estimation. However, existing non-plug-in methods like the one in [20] often suffer from problems of having a high computational complexity, requiring very large data sets for accurate estimation, or being specifically developed for systems where the operating parameters are deterministic.

Motivated by the above discussion and a recently developed mutual information estimation method in [21], we propose a non-plug-in FIM estimator, referred to as Fisher Information Neural Estimator (FINE), which has several advantages such as having a low computational complexity, high estimation accuracy, and applicable for both cases of deterministic and random parameters with high dimensions. It should also be noted that FINE employs neural networks and thus takes advantages of their nice properties such as the ability to learn non-linear and complex relationships and having low computational complexities.

The contributions of this paper are summarized as follows. First, we propose FINE – a Fisher information estimator based on neural networks for the case of deterministic parameters. The proposed FINE is based on a relation between the Fisher information and the $f$-divergence, which was exploited in a previous work [20]. However, unlike [20] which computed the $f$-divergence by using the Friedman-Rafsky (FR) statistic [22], FINE computes the $f$-divergence by neural networks. Compared to [20], FINE has not only higher estimation accuracy but also a lower computational complexity.

Second, we show that the proposed FINE framework can be used for the case of random parameters, i.e., FINE is applicable for the Bayesian Fisher information estimation problem. We prove that the relation between the Bayesian Fisher information matrix (B-FIM) and the $f$-divergence follows an expression that is similar to the case of deterministic parameters. To validate the efficacy of the proposed FINE in the Bayesian framework, we carry out some simulations about dynamical phase offset estimation in a communication system. Numerical results show that the proposed FINE gives a better estimation accuracy compared to an existing asymptotic bound.

Third, we propose an accelerated FINE version which is suitable for scenarios where the parameter dimension is high. More specifically, a minimum number of parameter perturbations is used and the least-square (LS) estimator in the FINE version is not required. These significantly reduce the computational complexity. Finally, we develop an algorithm for choosing an appropriate size of the neural networks used in the proposed FINE and accelerated FINE.

The remainder of this paper is organized as follows. In Section 2, we start with a brief review of Fisher information, $f$-divergence, and their relationship. Then we conclude the section with a problem statement and related work. Section 3 presents the proposed FINE for both cases of deterministic and random variables. A computational complexity analysis is given in Section 4. The accelerated FINE is also proposed in this section. Simulations are carried out in Section 5 for validation and performance comparison. Section 5 also presents an algorithm for choosing the neural network size. Finally, Section 6 concludes the paper.

*Notation:* Upper-case and lower-case boldface letters denote matrices and column vectors, respectively. $\mathbb{E}[\cdot]$ represents expectation. The operator $|\cdot|$ denotes the absolute value of a number and the operator $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix. The transpose is denoted by $[\cdot]^T$. The symbol $\mathcal{N}(\cdot,\cdot)$ represents the normal distribution, where the first argument is the mean and the second argument is the variance or the covariance matrix. $\mathbb{R}$ denotes the set of real numbers.

## 2 Background and Problem Statement

### 2.1 Fisher Information and $f$-divergence

*2.1.1 Fisher Information:* Consider a random variable $X$ whose pdf $p(x|\boldsymbol{\theta})$ is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, a vector of $d$ unknown parameters. When $\boldsymbol{\theta}$ is deterministic, the FIM $\mathbf{F}(\boldsymbol{\theta})$ is defined as follows [23]:

$$\begin{aligned}\mathbf{F}(\boldsymbol{\theta}) &= \mathbb{E}_{X|\boldsymbol{\theta}}\left[\left(\nabla_{\boldsymbol{\theta}} \log p(x|\boldsymbol{\theta})\right)\left(\nabla_{\boldsymbol{\theta}} \log p(x|\boldsymbol{\theta})\right)^T\right] \\ &= -\mathbb{E}_{X|\boldsymbol{\theta}}\left[\mathbf{H}_{\boldsymbol{\theta}}\left(\log p(x|\boldsymbol{\theta})\right)\right], \qquad (1)\end{aligned}$$

where $\nabla_{\boldsymbol{\theta}} \log p(x|\boldsymbol{\theta})$ and $\mathbf{H}_{\boldsymbol{\theta}}\left(\log p(x|\boldsymbol{\theta})\right)$ respectively denote the gradient and the Hessian matrix of the score function $\log p(x|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. In case the parameter vector $\boldsymbol{\theta}$ is random, the B-FIM $\mathbf{B}$ is used instead [24, 25]

$$\begin{aligned}\mathbf{B} &= \mathbb{E}_{X,\boldsymbol{\theta}}\left[\left(\nabla_{\boldsymbol{\theta}} \log p(x,\boldsymbol{\theta})\right)\left(\nabla_{\boldsymbol{\theta}} \log p(x,\boldsymbol{\theta})\right)^T\right] \\ &= -\mathbb{E}_{X,\boldsymbol{\theta}}\left[\mathbf{H}_{\boldsymbol{\theta}}\left(\log p(x,\boldsymbol{\theta})\right)\right], \qquad (2)\end{aligned}$$

where $p(x,\boldsymbol{\theta})$ is the joint pdf of $X$ and $\boldsymbol{\theta}$.

*2.1.2 $f$-divergence:* For any convex function $f$ such that $f(1) = 0$, the $f$-divergence between two probability distributions $p(x)$ and $q(x)$ is defined as a function $D_f(p\|q)$ that measures the difference between $p(x)$ and $q(x)$ [26]:

$$D_f(p\|q) = \mathbb{E}_q\left[f\left(\frac{p(x)}{q(x)}\right)\right] = \int q(x) f\left(\frac{p(x)}{q(x)}\right) dx. \quad (3)$$

The Kullback-Leibler (KL) divergence is a special case of the $f$-divergence where $f(t)=t\log(t)$ and is given as

$$D_{\mathrm{KL}}(p\|q)=\mathbb{E}_p\left[\log\left(\frac{p(x)}{q(x)}\right)\right]=\int p(x)\log\left(\frac{p(x)}{q(x)}\right)dx. \tag{4}$$

*2.1.3 Relation Between Fisher Information and f-divergence:* For notational simplicity, let $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\eta}}$ denote the probability distribution of a random variable $X$ parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\eta}=\boldsymbol{\theta}+\delta$, respectively. Here, $\delta$ is a small perturbation around $\boldsymbol{\theta}$. This means $p_{\boldsymbol{\theta}}=p(x|\boldsymbol{\theta})$ and $p_{\boldsymbol{\eta}}=p(x|\boldsymbol{\theta}+\delta)$. The relation between the Fisher information $\mathbf{F}(\boldsymbol{\theta})$ and the $f$-divergence between $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}+\delta}$ is given in a quadratic form as [20, 27]

$$D_f(p_{\boldsymbol{\theta}}||p_{\boldsymbol{\eta}})\approx\frac{1}{2}\delta^T\mathbf{F}(\boldsymbol{\theta})\delta. \tag{5}$$

The above relation can be obtained by applying the Taylor expansion to the $f$-divergence. This relation indicates that if $D_f(p_{\boldsymbol{\theta}}\|p_{\boldsymbol{\eta}})$ can be computed for at least $d(d+1)/2$ different perturbations $\delta$, then the FIM $\mathbf{F}(\boldsymbol{\theta})$ can be obtained by solving (5). This is due to the fact that $\mathbf{F}(\boldsymbol{\theta})\in\mathbb{R}^{d\times d}$ is a symmetric matrix and thus contains $d(d+1)/2$ different elements. It should be noted that the relation in (5) is for the FIM. One of our contributions is to prove that the relation between the B-FIM and the $f$-divergence follows an expression that is similar to (5).

### 2.2 Problem Statement

Consider a random variable $X$ whose pdf $p(x|\boldsymbol{\theta})$ is unknown. The parameters $\boldsymbol{\theta}$ can be either deterministic or random. We assume the parameters are tunable in the sense that they can be perturbed by a small deviation $\delta$. The problem is to estimate the FIM $\mathbf{F}(\boldsymbol{\theta})$ when $\boldsymbol{\theta}$ is deterministic or the B-FIM $\mathbf{B}$ when $\boldsymbol{\theta}$ is random using the data samples of $X$.

The closest related work to ours is [20], where Berisha and Hero exploited the relation in (5) to develop a non-plug-in Fisher information estimator based on the FR statistic. More specifically, the method in [20] approximates $D_f(p_{\boldsymbol{\theta}}||p_{\boldsymbol{\eta}})$ using the minimal spanning tree (MST) for $M$ different perturbations of $\delta$ where $M\geq d(d+1)/2$. Then, the FIM $\mathbf{F}(\boldsymbol{\theta})$ is obtained by solving (5) based on the $M$ computed divergence values. However, the complexity of the MST construction is $\mathcal{O}(N^2)$, where $N$ is the total number of data samples. Our proposed approach employs neural networks to approximate $D_f(p_{\boldsymbol{\theta}}||p_{\boldsymbol{\eta}})$ with a $\mathcal{O}(N)$ complexity. As will be shown later, the proposed approach gives a higher estimation accuracy compared to the method in [20] and is also scalable to high dimensions.

## 3 Proposed FINE

### 3.1 Deterministic Parameters

Here, we consider deterministic parameters $\boldsymbol{\theta}$ and we want to estimate the FIM $\mathbf{F}(\boldsymbol{\theta})$. FINE also exploits the relation between the Fisher information and the $f$-divergence in (5) but unlike the method in [20] which computes the $f$-divergence by using the FR statistic and the MST, the proposed FINE employs neural networks to compute the $f$-divergence. As will be shown later, the use of neural networks not only helps improve the estimation accuracy but also significantly reduces the computational complexity.

Our motivation for computing the $f$-divergence by neural networks comes from a recently developed mutual information neural estimation method in [21], which is referred to as MINE. Specifically, the mutual information between two random variables $Y$ and $Z$ is given as

$$I(Y,Z)=H(Y)-H(Y\mid Z) \tag{6}$$
$$=D_{\mathrm{KL}}(P_{YZ}\|P_YP_Z) \tag{7}$$
$$=\sup_{T:\Omega\to\mathbb{R}}\mathbb{E}_{P_{YZ}}[T]-\log\left(\mathbb{E}_{P_YP_Z}[e^T]\right) \tag{8}$$
$$\geq\sup_{T\in\mathcal{F}}\mathbb{E}_{P_{YZ}}[T]-\log\left(\mathbb{E}_{P_YP_Z}[e^T]\right) \tag{9}$$

where $\mathcal{F}$ is any class of functions $T:\Omega\to\mathbb{R}$ satisfying the integrability constraints of the Donsker-Varadhan representation [21], and $\Omega$ is a compact domain that the two distributions $P_{YZ}$ and $P_YP_Z$ belong to. Here, $I(\cdot)$ and $H(\cdot)$ denote the mutual information and the entropy, respectively. Note that the equality in (8) is the Donsker-Varadhan representation. Belghazi et al. utilize a well-known property of neural networks stated as the universal approximation theorem. The idea in [21] is to treat $T$ as a neural network and the mutual information $I(Y,Z)$ is estimated by using the observed data to train $T$ such that $\mathbb{E}_{P_{YZ}}[T]-\log\left(\mathbb{E}_{P_YP_Z}[e^T]\right)$ is maximized. After training, the trained objective function value reads an estimate of the mutual information. Note that the MINE method can be applied to estimate a general $f$-divergence [28]. In [21], $T$ is referred to as a statistic network because it is trained to estimate a statistic.

Our idea is to compute $D_f(p_{\boldsymbol{\theta}}\|p_{\boldsymbol{\eta}})$ in (5) by using the observed data to train a neural network $T$ such that $\mathbb{E}_{p_{\boldsymbol{\theta}}}[T]-\log\left(\mathbb{E}_{p_{\boldsymbol{\eta}}}[e^T]\right)$ is maximized. Since $\mathbf{F}(\boldsymbol{\theta})$ contains $d(d+1)/2$ different elements, we need to compute $D_f(p_{\boldsymbol{\theta}}\|p_{\boldsymbol{\eta}})$ for $M$ different perturbations of $\delta$ where $M\geq d(d+1)/2$. Then we can obtain an estimate of $\mathbf{F}(\boldsymbol{\theta})$ by solving (5).

Let $\boldsymbol{\eta}_i=\boldsymbol{\theta}+\delta_i$ with $i=1,\dots,M$ and let $\mathbf{X}\in\mathbb{R}^{N\times K}$ and $\mathbf{X}_i\in\mathbb{R}^{N\times K}$ denote the data sets observed from $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\eta}_i}$, respectively. Here, $N$ and $K$ are the number of data samples and the size of each sample, respectively. A neural network $T_i$ is used to estimate the $f$-divergence $d(\delta_i)=D_f(p_{\boldsymbol{\theta}}\|p_{\boldsymbol{\eta}_i})$ based on $\mathbf{X}$ and $\mathbf{X}_i$. Specifically, $T_i$ takes a vector of size $K$ as the input and returns a scalar as the output. Thus, $\mathbf{X}$ and $\mathbf{X}_i$ are the input data sets of $T_i$. Let $\mathbf{z}=T_i(\mathbf{X})\in\mathbb{R}^N$ and $\mathbf{z}_i=T_i(\mathbf{X}_i)\in\mathbb{R}^N$. Then $T_i$ is trained to maximize $\mathbf{1}^T\mathbf{z}/N-\log(\mathbf{1}^T\exp\{\mathbf{z}_i\}/N)$, which is used as an estimate of $d(\delta_i)$. For notational simplicity, we use $\exp\{\mathbf{z}_i\}$ to indicate that $\exp\{\cdot\}$ is applied to $\mathbf{z}_i$ element-wise. An illustration of the proposed FINE is given in Figure 1.

Once the $f$-divergences have been obtained, we need to solve (5) for $\mathbf{F}(\boldsymbol{\theta})$ under the constraint that $\mathbf{F}(\boldsymbol{\theta})$ is a symmetric positive semi-definite (PSD) matrix. We

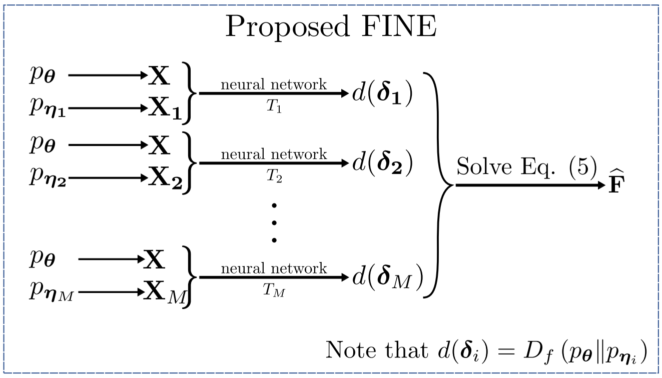

Proposed FINE
$p_{\boldsymbol{\theta}}$
$\mathbf{X}$
$p_{\boldsymbol{\eta}_1}$
$\mathbf{X}_1$
neural network
$T_1$
$d(\boldsymbol{\delta}_1)$
$p_{\boldsymbol{\theta}}$
$\mathbf{X}$
$p_{\boldsymbol{\eta}_2}$
$\mathbf{X}_2$
neural network
$T_2$
$d(\boldsymbol{\delta}_2)$
Solve Eq. (5)
$\widehat{\mathbf{F}}$
$p_{\boldsymbol{\theta}}$
$\mathbf{X}$
$p_{\boldsymbol{\eta}_M}$
$\mathbf{X}_M$
neural network
$T_M$
$d(\boldsymbol{\delta}_M)$
Note that $d(\boldsymbol{\delta}_i) = D_f\left(p_{\boldsymbol{\theta}} \| p_{\boldsymbol{\eta}_i}\right)$

Figure 1. Illustration of the proposed FINE method.

vectorize $\mathbf{F}(\boldsymbol{\theta})$ by including the distinct upper triangular values of $\mathbf{F}(\boldsymbol{\theta})$ and convert (5) to a linear function of this quantity. Let

$$\mathbf{f} = [F_{11}, \ldots, F_{dd}, F_{12}, \ldots, F_{1d}, F_{23}, \ldots, F_{2d}, \ldots, F_{(d-1)d}]^T,$$

where $F_{ij}$ is the element in the $i$-row and $j$-column of $\mathbf{F}(\boldsymbol{\theta})$, and let

$$\mathbf{u}_i = [\delta_{i1}^2, \ldots, \delta_{id}^2, 2\delta_{i1}\delta_{i2}, \ldots, 2\delta_{i1}\delta_{id}, \ldots, 2\delta_{i(d-1)}\delta_{id}]^T,$$

where $i = 1, \ldots, M$, $\delta_{ij}$ is the the $j$-element of $\delta_i$. Denote $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]^T$, then we have a linear system $2\mathbf{d} = \mathbf{U}\mathbf{f}$ where $\mathbf{d} = [d(\delta_1), \ldots, d(\delta_M)]^T$. Using the least square (LS) estimator, we can find an estimate of $\mathbf{f}$ as

$$\hat{\mathbf{f}}^{\mathrm{LS}} = 2\big(\mathbf{U}^T\mathbf{U}\big)^{-1}\mathbf{U}^T\mathbf{d}. \tag{10}$$

This LS estimator, however, does not ensure that the resulting estimate is positive semi-definite. So we employ a semi-definite program (SDP) as follows [20]:

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \|2\mathbf{d} - \mathbf{U}\mathbf{f}\|_2^2 \\ \text{subject to} \quad & f_k = \hat{f}_k^{\mathrm{LS}},\ k = 1, \ldots, d \\ & \mathrm{mat}(\mathbf{f}) = \mathbf{F}(\boldsymbol{\theta}) \succeq \mathbf{0} \end{aligned} \tag{11}$$

where $f_k$ and $\hat{f}_k^{\mathrm{LS}}$ are the $k$-th element of $\mathbf{f}$ and $\hat{\mathbf{f}}^{\mathrm{LS}}$, respectively. The $\mathrm{mat}(\cdot)$ operator converts the vectorized FIM $\mathbf{f}$ to a full matrix representation $\mathbf{F}(\boldsymbol{\theta})$. To ensure the symmetric PSD requirement, we only need to refine the off-diagonal elements of $\mathbf{F}(\boldsymbol{\theta})$, which explains the constrains $f_k = \hat{f}_k^{\mathrm{LS}}$, $k = 1, \ldots, d$.

### 3.2 Random Parameters

In many systems, the operating parameters $\boldsymbol{\theta}$ are not deterministic, but random. This leads to the study of the B-FIM $\mathbf{B}$. Let $\pi(\boldsymbol{\theta})$ be the distribution of $\boldsymbol{\theta}$, then

$$\begin{aligned} \mathbf{B} &= -\mathbb{E}_{X,\boldsymbol{\theta}}\left[\mathbf{H}_{\boldsymbol{\theta}}\big(\log p(x, \boldsymbol{\theta})\big)\right] \\ &= -\mathbb{E}_{\boldsymbol{\theta}}\left[\mathbf{H}_{\boldsymbol{\theta}}\left(\log p(\boldsymbol{\theta})\right)\right] - \mathbb{E}_{\boldsymbol{\theta}X}\left[\mathbf{H}_{\boldsymbol{\theta}}\left(\log p(x|\boldsymbol{\theta})\right)\right] \\ &= \mathbf{F}(\pi) + \mathbb{E}_{\boldsymbol{\theta}}\left[\mathbf{F}(\boldsymbol{\theta})\right]. \end{aligned} \tag{12}$$

As can be seen from (12) that the B-FIM $\mathbf{B}$ does not depend on a particular value of $\boldsymbol{\theta}$ and consists of two terms. The first term is the information of the prior distribution and the second term is the expected Fisher information. Some closely-related studies about the Bayesian Carmér-Rao bound (BCRB) were presented in [29] and [30, Chapter 7]. To the best of our knowledge, the relation between the $f$-divergence and the Bayesian Fisher information has not been stated formally in the literature. Here, we show that the relation between the B-FIM and the $f$-divergence follows an expression that is similar to the one in (5) by Theorem 1 below.

**Theorem 1.** *Consider a distribution $P_{X\theta}$ with the pdf $p(x, \theta + \delta)$ and another distribution $Q_{X\theta}$ with the pdf $p(x, \theta)$, where $\theta \in \Theta \subset \mathbb{R}$ is a random parameter, and $\delta \in \mathbb{R}$ is a small perturbation. For any convex function $f$ satisfying $f(1) = 0$ and $f''(1) = 1$, the $f$-divergence between $P_{X\theta}$ and $Q_{X\theta}$*

$$D_f(P_{X\theta}||Q_{X\theta}) = \iint f\left(\frac{p(x, \theta + \delta)}{p(x, \theta)}\right) p(x, \theta) dx d\theta \tag{13}$$

*can be approximated as follows:*

$$D_f(P_{X\theta}||Q_{X\theta}) \approx \frac{1}{2}\delta^2 B \tag{14}$$

*where $B$ is the Bayesian information and defined as*

$$B = \iint \left(\frac{\partial \log p(x, \theta)}{\partial \theta}\right)^2 p(x, \theta) dx d\theta. \tag{15}$$

*Proof:* Using the Taylor expansion about $\theta$, we have

$$p(x, \theta + \delta) = p(x, \theta) + p'(x, \theta)\delta + o(\delta), \tag{16}$$

and thus

$$D_f(P_{X\theta}||Q_{X\theta}) = \iint f\left(1 + \frac{p'(x, \theta)\delta}{p(x, \theta)}\right) p(x, \theta) dx d\theta. \tag{17}$$

Using the Taylor expansion of $f$ about 1, we have

$$f(1 + \Delta) = f(1) + f'(1)\Delta + \frac{1}{2}f''(1)\Delta^2 + o(\Delta^2) \approx \frac{1}{2}\Delta^2. \tag{18}$$

The approximation in (18) is obtained since $f(1) = 0$, $f'(1) = 0$, and $f''(1) = 1$. We can always have $f'(1) = 0$ because $D_{f_c}(P\|Q) = D_f(P\|Q)$ where $f_c(t) = f(t) - c(t-1)$, which means if $f(t)$ does not satisfy $f'(1) = 0$, we can replace $f(t)$ by $f_c(t)$ with $c = f'(1)$. Applying (18) to (17), we obtain

$$D_f(P_{X\theta}||Q_{X\theta}) \approx \frac{\delta^2}{2} \iint \left(\frac{p'(x, \theta)}{p(x, \theta)}\right)^2 p(x, \theta) dx d\theta = \frac{\delta^2}{2} B$$

■

Although Theorem 1 is stated for one dimensional parameters, for higher dimensional parameters $\boldsymbol{\theta}$, one can use the same reasoning and obtain

$$D_f(P_{X\boldsymbol{\theta}}||Q_{X\boldsymbol{\theta}}) \approx \frac{1}{2}\delta^T \mathbf{B} \delta. \tag{19}$$

Hence, the relation between the $f$-divergence and the Bayesian Fisher information follows an expression similar to the case of deterministic parameters in (5). Therefore, FINE can be directly applied to this Bayesian framework. However, it should be noted that the data samples in this Bayesian framework are generated from the joint distributions of $X$ and $\boldsymbol{\theta}$.

# 4 Accelerated FINE

## 4.1 Complexity Analysis

As described in the previous section, the proposed FINE approach has two stages: (i) $f$-divergence estimation by neural networks for various perturbations; and (ii) FIM estimation by solving (5) for deterministic $\boldsymbol{\theta}$ or B-FIM estimation by solving (19) for random $\boldsymbol{\theta}$.

The complexity of the first stage depends on the time for training the neural networks, which is $\mathcal{O}(JWN)$ [31] for each purterbation. Here, $J$, $W$, and $N$ are the number of epochs, the number of trainable parameters, and the number of training samples, respectively. For reliable estimation, $N$ often needs to be large, and so $JW$ is relatively small compared to $N$, making the complexity of FINE roughly $\mathcal{O}(N)$. Compared to the empirical estimator proposed by Berisha and Hero[20], their method needs to construct the MSTs of dense graphs whose complexity is $\mathcal{O}(N^2)$. We will show later that the run time of FINE is significantly lower than the run time of the estimator in [20].

In the second stage, FINE needs to implement an LS estimator in (10) and solve an SDP problem in (11). The complexity of the LS estimator in (10) is $\mathcal{O}\big(\max(d^4M, d^6)\big)$ because the size of the matrix $\mathbf{U}$ is $M \times (d(d+1)/2)$. Problem (11) is an SDP and so it can be solved in a polynomial time. Note that the complexity of the second stage of the method by Berisha and Hero in [20] is similar to that of the proposed FINE because they share the same second stage.

## 4.2 Accelerated FINE

The complexity analysis above shows that the proposed FINE has a lower complexity compared to the method in [20]. However, FINE can still be expensive to implement when the parameter dimension $d$ is large. This is because a large value of $d$ will also require large $N$ and $M$ for reliable estimation. Here, we propose an accelerated FINE which can be employed for scenarios where $d$ is large. The proposed accelerated FINE does not need to implement the LS estimator in (10) whose complexity is $\mathcal{O}(\max(d^4M, d^6))$, and only uses $M = d(d+1)/2$ perturbations.

The idea is to sequentially estimate the elements of the B-FIM $\mathbf{B}$. The strategy of the accelerated FINE is to perturb the elements of $\boldsymbol{\theta}$ one-by-one when estimating the diagonal elements of $\mathbf{B}$, and pair-by-pair when estimating the off-diagonal elements of $\mathbf{B}$. Specifically, for estimating a diagonal element $B_{ii}$, we will only perturb the $i$-th element of $\boldsymbol{\theta}$. This means the elements of the perturbation vector $\delta$ are all zeros, except the $i$-th one, i.e., $\delta_\ell \neq 0$ if $\ell = i$ and $\delta_\ell = 0$ if $\ell \neq i$. With this perturbation, we can obtain an estimate of $B_{ii}$ as follows:

$$\widehat{B}_{ii} \approx \frac{2d(\delta)}{\delta_i^2}. \tag{20}$$

Note that $d(\delta)$ is still obtained by a neural network.

After estimating all the diagonal elements $B_{ii}$ of $\mathbf{B}$, we can now estimate the off-diagonal elements of $\mathbf{B}$ as follows. For estimating an off-diagonal element $B_{ij}$ with $i \neq j$, we will only perturbs the $i$-th and the $j$-th elements of $\boldsymbol{\theta}$, which means $\delta_\ell \neq 0$ if $\ell = i$ or $\ell = j$, otherwise $\delta_\ell = 0$. With this pair perturbation, we can obtain an estimate of $B_{ij}$ as follows:

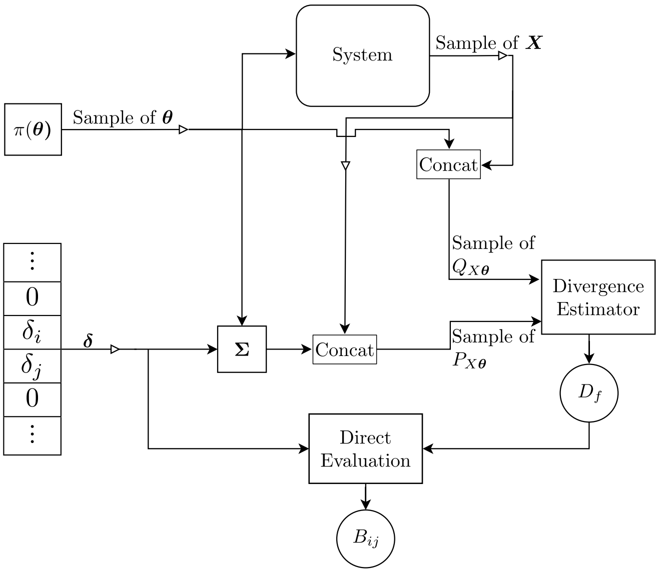

System
Sample of $\boldsymbol{X}$
$\pi(\boldsymbol{\theta})$
Sample of $\boldsymbol{\theta}$
Concat
Sample of $Q_{X\boldsymbol{\theta}}$
Divergence Estimator
⋮
0
$\delta_i$
$\delta_j$
0
⋮
$\delta$
$\Sigma$
Concat
Sample of $P_{X\boldsymbol{\theta}}$
$D_f$
Direct Evaluation
$B_{ij}$

Figure 2. Procedure of the accelerated FINE for B-FIM estimation.

$$\widehat{B}_{ij} = \frac{2d(\delta) - \delta_i^2\widehat{B}_{ii} - \delta_j^2\widehat{B}_{jj}}{2\delta_i\delta_j}. \tag{21}$$

It should be noted that $\mathbf{B}$ is a symmetric matrix, and thus, we only need to estimate the upper triangular part of $\mathbf{B}$, i.e., for off-diagonal elements, estimating $B_{ij}$ with $j > i$ is sufficient. An illustration for the procedure of the accelerated FINE is given in Figure 2 where the "Direct Evaluation" module utilizes (20) and (21).

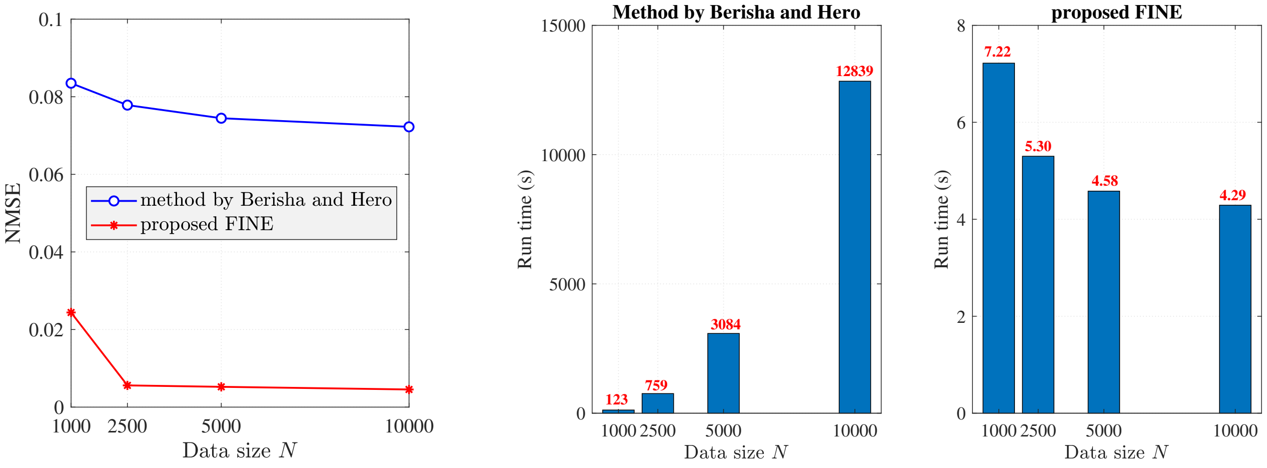
Finally, to make sure that $\widehat{\mathbf{B}}$ is PSD, we employ a SDP as follows:

$$\begin{aligned} \underset{\mathbf{B}}{\text{minimize}} \quad & ||\mathbf{B} - \widehat{\mathbf{B}}||_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{B} \succeq \mathbf{0} \\ & B_{ii} = \widehat{B_{ii}}, \quad i \in 1, \ldots, d. \end{aligned} \tag{22}$$

It is worth noting that, when the size of the data set ($N$) is large enough, it is often found that the estimated B-FIM $\widehat{\mathbf{B}}$ obtained from (20) and (21) already satisfies the PSD requirement. In this case, we do not need to solve the SDP in (22). It should also be noted that although the above procedure is for estimating the B-FIM $\mathbf{B}$, i.e., for the case random parameters, it can also be directly applied to the case of deterministic parameters.

# 5 Numerical Results

Here we numerically validate and evaluate the performance of FINE for both cases of deterministic and random parameters. We use neural networks with hyperparameters as illustrated in Figure 4 where $W_{in}$ denotes the width of input layer, $L$ and $W$ denote the number of hidden layers and the width of those hidden layers, respectively. The ReLU activation function is used and the output of the network is a scalar.



(a) NMSE

(b) Run time

Figure 3. Estimation accuracy and computational complexity comparison for the case of deterministic $\boldsymbol{\theta}$.

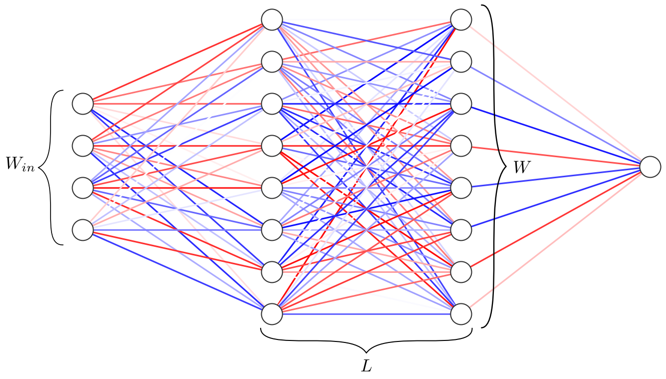


Figure 4. Example of a statistics network.

### 5.1 Deterministic Parameters

Here, we provide test results for the case of deterministic parameters. Consider a $K$-dimensional Gaussian distribution as $\mathcal{N}(\boldsymbol{\theta}, \mathbf{I}_K)$. The objective is to estimate the FIM $\mathbf{F}$ at $\boldsymbol{\theta} = \mathbf{0}$. It should be noted that $d = K$ in this scenario. The FIM has a simple closed form as $\mathbf{F}_{\text{true}} = \mathbf{I}$. Each element of $\boldsymbol{\delta}$ is drawn from $\mathcal{N}(\mathbf{0}, 0.05\mathbf{I}_K)$. We use the same data size of $N$ for all data sets $\mathbf{X}$, $\mathbf{X}_1, \ldots, \mathbf{X}_M$. We set $K = 4$ and $M = 5d(d+1)/2$. Here, $W_{in} = K = 4$ and we set $W = 5W_{in}$ and $L = 1$. The proposed FINE is compared with the method proposed by Berisha and Hero in [20]. Test results are given in Figure 3. The normalized mean squared error (NMSE) is defined as NMSE $= \|\hat{\mathbf{F}} - \mathbf{F}_{\text{true}}\|_{\text{F}}^2 / d^2$. It can be seen that the proposed FINE outperforms the method in [20] in terms of both accuracy and computational complexity. The run time of the method by Berisha and Hero scales quadratically with $N$ whereas the run time of the proposed FINE decreases because it is found that the neural networks converged faster with larger data sets.

### 5.2 Random Parameters

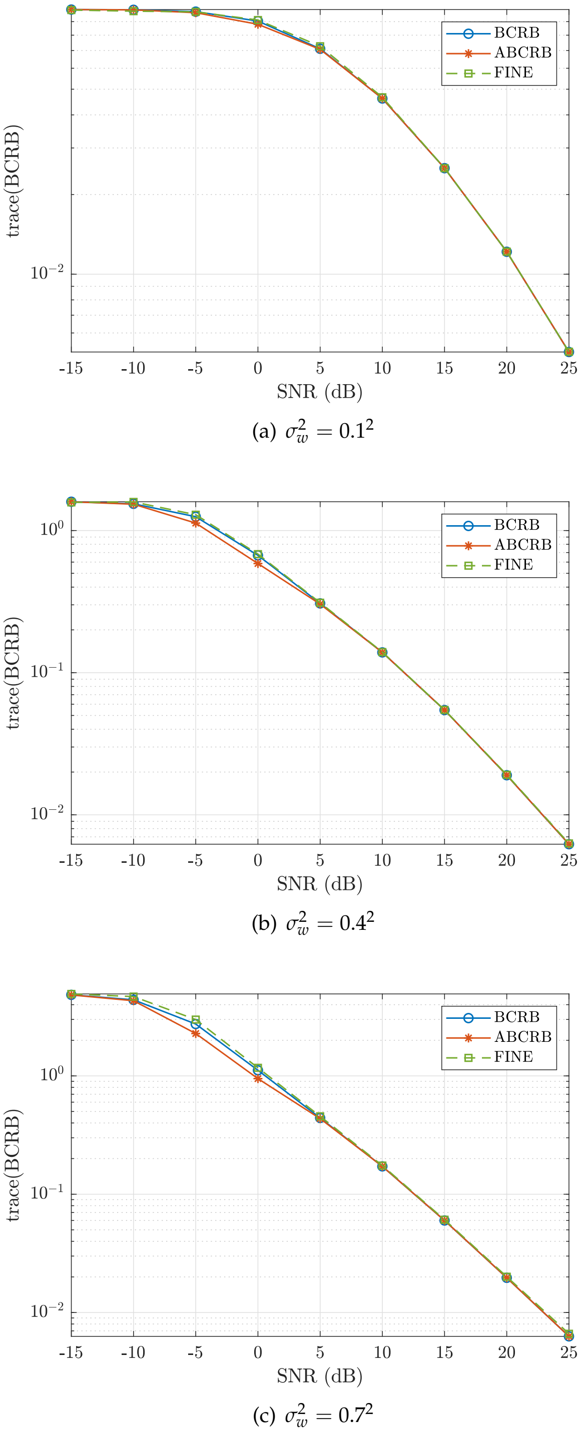
Here, we provide test results for the case of random parameters. Consider the transmission of $\tau$ BPSK symbols $\boldsymbol{a} = [a_1, \ldots, a_\tau]^T$ over an additive white Gaussian noise (AWGN) channel affected by carrier phase offsets $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_\tau]^T$, the received signal is given as

$$y_t = a_t e^{j\theta_t} + n_t,$$

where $n_t$ is the additive white Gaussian noise and distributed as $\mathcal{N}(0, \sigma_n^2)$ and $\boldsymbol{\theta}$ follows the Wiener phase-offset evolution, i.e., $\theta_t = \theta_{t-1} + w_t$, where $w_t \sim \mathcal{N}(0, \sigma_w^2)$. The receiver needs to estimate the carrier-phase offsets $\boldsymbol{\theta}$. In [32], a closed-form BCRB was derived for this scenario. In addition, the authors also proposed an asymptotic bound, which is referred to as ABCRB. We adopt the observation block length $\tau = 4$ and various values of $\sigma_w^2$. In this scenario, $d = \tau$ and $K = 2\tau$ since the received signal $y_t$ is in the complex domain. In general, the input size for the case of random parameters is $W_{in} = d + K$. Here, we use neural networks with $L = 1$ and $W = 5W_{in}$. Comparison results are shown in Figure 5. It is observed that with a low value of $\sigma_w^2$, both the estimated BRCB (by FINE) and ABCRB are close to the true BCRB. Figures 5(b) and 5(c) show that our proposed FINE produces remarkable improvements over the ABCRB when $\sigma_w^2$ is higher. Thus, the results in Figure 5 verify the efficiency of the proposed FINE in case of random parameters.

### 5.3 Scalability

In this section, we show the scalability of the proposed accelerated FINE by considering high parameter dimensions. As mentioned in Section 4.1, the high complexities of FINE and the method in [20] make them only suitable for small parameter dimensions. Here, we set the parameter dimension $d$ as well as the data sample size $N$ to large values for which FINE and the method in [20] are too expensive to implement. However, the proposed accelerated FINE can still handle the cases of large $d$ and $N$ efficiently. We try to estimate the FIM of the parameters of the model described in Section 5.1 with $d$ increasing up to 120. We use neural networks with the size of $L = 1$ and $W = 2W_{in}$. Numerical results presented in Figure 6 verify the scalability of the proposed accelerated FINE



(a) $\sigma_w^2 = 0.1^2$

(b) $\sigma_w^2 = 0.4^2$

(c) $\sigma_w^2 = 0.7^2$

Figure 5. Validation of FINE for the case of random $\boldsymbol{\theta}$.

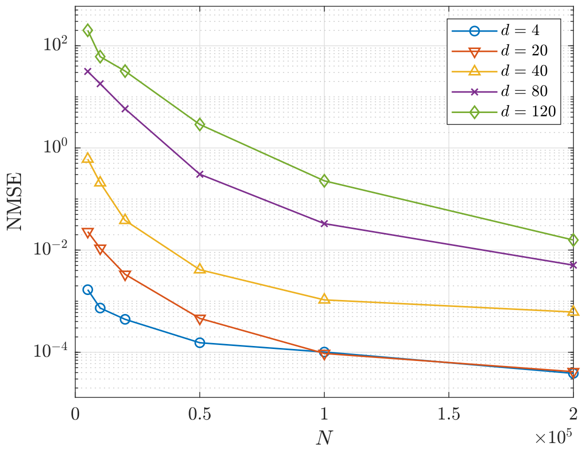


Figure 6. Accelerated FINE for high parameter dimensions.

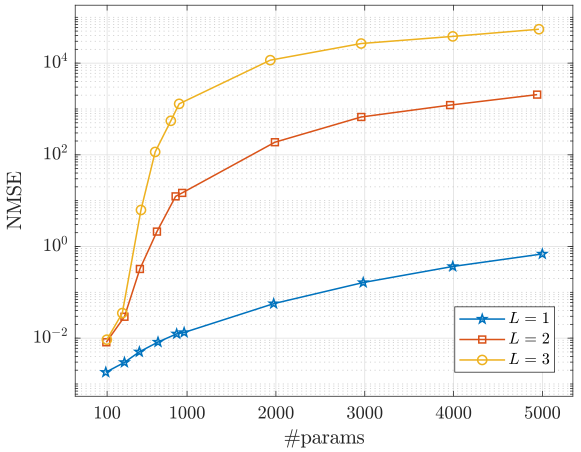


Figure 7. Effects of the neural network size with $d = 40$ and $N = 10000$.

as $d$ increases. It can also be seen that the accelerated FINE can give reliable estimation results for large $d$ as long as the data sample size $N$ is large enough.

### 5.4 Effect of the Neural Network Size

In this section, we examine the effect of the neural network size on the estimation accuracy and propose a method for choosing an appropriate network size that gives the best estimation performance.

We consider the same number of units in all hidden layers. The total number of trainable parameters (the network size) is changed by varying the number of hidden layers $L$ and the number of units $W$ in each hidden layer. The total number of parameters in terms of $W_{in}$, $L$, and $W$ is given as

$$\#params = (L-1)W^2 + (W_{in} + L + 1)W + 1.$$

Results in Figure 7 show that different network sizes give different estimation results and increasing the network size does not necessarily improve the performance, which is due to the overfitting phenomenon. This prompts the need for developing an algorithm that can choose a proper network size to give the best performance.

In Figure 8 and Figure 9, we examine the convergence of the network training process versus the network size and the training epoch. The normalized error is given as $|estimate - ground\_truth|/ground\_truth$. It is observed that a large network size can make the training process diverge (Figure 8) and a small network size can make the training converge but it does not converge to the ground truth (Figure 9). In the following, we propose a method for choosing an appropriate network size that can give the optimal solution. The proposed algorithm is based on an observation that there is some range of the network size that produces good results.

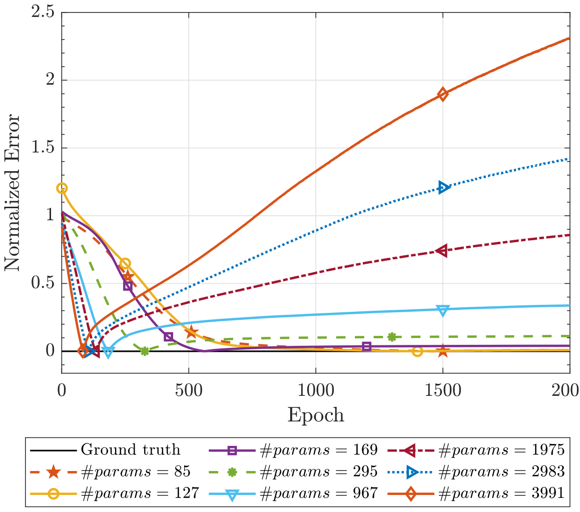


Figure 8. Estimation error versus *#params* and training epoch for the test scenario in Section 5.1 with $L = 1$, $W = 40$, and $N = 20000$.

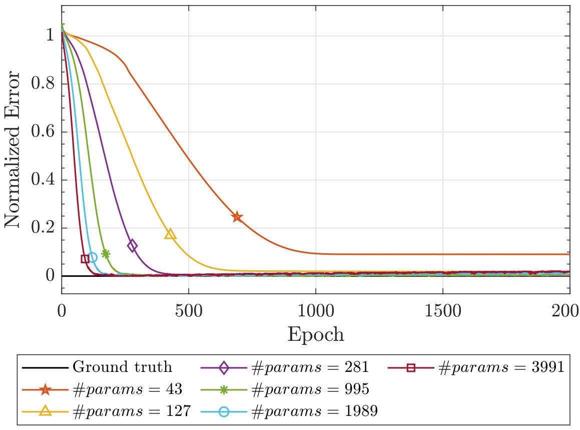


Figure 9. Estimation error versus *#params* and training epoch for the test scenario in Section 5.2 with $L = 1$, $W = 12$, and $N = 10000$.

A description of the proposed method is given in Algorithm 1. First, we generate samples from the original distribution and a perturbed distribution. Then we initialize an ascending array $\mathcal{W}$ of candidate values. For each $W_i$ in the array $\mathcal{W}$, we train a corresponding statistics network using with the data samples for a large number of epochs to obtain an estimate of the divergence, which is stored in an array. We define an auxiliary function to compute "mean gradient", which is defined as the mean of the absolute difference between an estimate and its two neighbouring estimates, except for the first and last estimates since there is only one adjacent estimate. The $W$ value that produces the smallest mean gradient is chosen as the optimal width of the hidden layer.

**Algorithm 1:** Choosing neural network size.

```
function MeanGradients($data$)
    $mean\_grads \leftarrow []$
    $n = data.length()$
    $grads.append(|data[0] - data[1]|)$
    for $i$ in $1 : 1 : (n-2)$ do
        $mean\_grad = (|data[i] - data[i-1]| +$
            $|data[i] - data[i+1]|)\,/2$
        $mean\_grads.append(mean\_grad)$
    end for
    $mean\_grads.append(|data[n-1] - data[n-2]|)$
    return $mean\_grads$
end function
Initialization:
$Q \leftarrow$ Generate samples from original distribution
$P \leftarrow$ Generate samples from perturbed distribution
$\mathcal{W} = \{W_1, W_2, \ldots, W_U\}$
$estimates \leftarrow []$
Train different statistics networks:
for $W$ in $\mathcal{W}$ do
    $estimate = Divergence\_approximate(P, Q, W)$
    $estimates.append(estimate)$
end for
Evaluation:
$mean\_grads = $ MeanGradients$(estimates)$
$i_{\mathtt{min}} = \operatorname{argmin}_i\ mean\_grads[i]$
return $\mathcal{W}[i_{\mathtt{min}}]$
```

## 6 Conclusion

This paper has proposed FINE – a Fisher information estimator based on neural networks. The proposed FINE was shown to be able to provide accurate FIM estimation with a low-computational complexity. We demonstrated that FINE is applicable for estimating the FIM of both deterministic and random parameters. We also introduced an accelerated FINE version whose computational complexity is much lower than that of the original FINE version and therefore applicable for high-dimensional parameters. Finally, we presented an algorithm for choosing a proper size of the neural networks used in the two FINE versions.

## Acknowledgment

This work was supported by the National Foundation for Science and Technology Development of Vietnam under Grant 01/2019/TN.

Tran Trong Duy was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.ThS.018.

## References

[1] A. Hannun, C. Guo, and L. van der Maaten, “Measuring data leakage in machine-learning models with Fisher information,” in *Proceedings of the Uncertainty in Artificial Intelligence*, vol. 161, 2021, pp. 760–770.

[2] A. Ly, M. Marsman, J. Verhagen, R. P. Grasman, and E.-J. Wagenmakers, “A tutorial on Fisher information,” *Journal of Mathematical Psychology*, vol. 80, pp. 40–55, Oct. 2017.

[3] S. A. Frank, “Natural selection maximizes Fisher information,” *Journal of Evolutionary Biology*, vol. 22, no. 2, pp. 231–244, Jan. 2009.

[4] N. Tran Duc, Y.-M. Frapart, and S. Li Thiao-Té, "Estimation of spectrum parameters for quantitative EPR in the derivative limit," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC)*, 2017, pp. 214–219.

[5] D.-N. Tran, S. Li-Thiao-Té, and Y.-M. Frapart, "Parameter estimation for quantitative epr spectroscopy," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–7, 2021.

[6] D. L. Donoho, "One-sided inference about functionals of a density," *The Annals of Statistics*, vol. 16, no. 4, pp. 1390–1420, Dec. 1988.

[7] P. J. Huber, "Fisher information and spline interpolation," *The Annals of Statistics*, vol. 2, no. 5, pp. 1029–1033, Sep. 1974.

[8] W. Cao, A. Dytso, M. Fauß, H. V. Poor, and G. Feng, "Nonparametric estimation of the Fisher information and its applications," *arXiv preprint arXiv:2005.03622*, 2020.

[9] W. Cao, A. Dytso, M. Fauß, H. V. Poor, and G. Feng, "On nonparametric estimation of the fisher information," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2216–2221.

[10] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, Sep. 1956.

[11] P. K. Bhattacharya, "Estimation of a probability density function and its derivatives," *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 29, no. 4, pp. 373–382, 1967.

[12] E. Nadaraya, "On Non-Parametric Estimates of Density Functions and Regression Curves," *Theory of Probability and Its Applications*, vol. 10, pp. 186–190, 1965.

[13] E. F. Schuster, "Estimation of a Probability Density Function and Its Derivatives," *The Annals of Mathematical Statistics*, vol. 40, no. 4, pp. 1187–1195, Aug. 1969.

[14] B. W. Silverman, "Weak and Strong Uniform Consistency of the Kernel Estimate of a Density and its Derivatives," *The Annals of Statistics*, vol. 6, no. 1, pp. 177–184, Jan. 1978.

[15] J. C. Spall, "Monte Carlo computation of the Fisher information matrix in nonstandard settings," *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, pp. 889–909, Jan. 2005.

[16] O. Har-Shemesh, R. Quax, B. Minano, A. G. Hoekstra, and P. M. Sloot, "Nonparametric estimation of Fisher information from real data," *Physical Review E*, vol. 93, no. 2, p. 023301, Feb. 2016.

[17] L. Gilbertson and R. A. Lutfi, "Correlations of decision weights and cognitive function for the masked discrimination of vowels by young and old adults," *Hearing Research*, vol. 317, pp. 9–14, 2014.

[18] S. Nelander, W. Wang, B. Nilsson, Q.-B. She, C. Pratilas, N. Rosen, P. Gennemark, and C. Sander, "Models from experiments: Combinatorial drug perturbations of cancer cells," *Molecular Systems Biology*, vol. 4, no. 1, p. 216, 2008.

[19] R. C. Jansen, "Studying complex biological systems using multifactorial perturbation," *Nature Reviews Genetics*, vol. 4, no. 2, pp. 145–151, 2003.

[20] V. Berisha and A. O. Hero, "Empirical non-parametric estimation of the Fisher information," *IEEE Signal Process. Letters*, vol. 22, no. 7, pp. 988–992, July 2015.

[21] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 531–540.

[22] J. H. Friedman and L. C. Rafsky, "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests," *The Annals of Statistics*, pp. 697–717, 1979.

[23] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical transactions of the Royal Society of London. Series A*, vol. 222, no. 594-604, pp. 309–368, 1922.

[24] H. L. van Trees, *Detection, Estimation and Modulation Theory Part I*. New York, NY, USA: John Wiley & Sons, 1968.

[25] R. D. Gill and B. Y. Levit, "Applications of the van Trees Inequality: A Bayesian Cramér-Rao Bound," *Bernoulli*, vol. 1, no. 1/2, pp. 59–79, 1995.

[26] A. Rényi, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 4, 1961, pp. 547–562.

[27] S.-i. Amari and A. Cichocki, "Information geometry of divergence functions," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 58, Mar. 2010.

[28] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Information Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.

[29] M. A. Kumar and K. V. Mishra, "Information geometric approach to bayesian lower error bounds," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 746–750.

[30] Y. Wu, "Lecture notes on information-theoretic methods for high-dimensional statistics," Yale University, USA, Jan. 2020. [Online]. Available: http://www.stat.yale.edu/∼yw562/teaching/it-stats.pdf

[31] E. Mizutani and S. E. Dreyfus, "On complexity analysis of supervised MLP-learning for algorithmic comparisons," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, 2001, pp. 347–352.

[32] S. Bay, C. Herzet, J.-M. Brossier, J.-P. Barbot, and B. Geller, "Analytic and asymptotic analysis of Bayesian Cramér–Rao bound for dynamical phase offset estimation," *IEEE Transactions on Signal Processing*, vol. 56, pp. 61–70, Feb. 2008.

**Tran Trong Duy** received the B.Eng. degree in electronics and telecommunications from the VNU University of Engineering and Technology (VNU-UET), Hanoi, Vietnam, in 2021. He has been a teaching assistant at VNU-UET since then. He is pursuing a joint Master's degree in communication and data engineering from VNU-UET and Paris-Saclay University. His main research interests include communication and machine learning.

**Nguyen Van Ly** (Member, IEEE) received the B.Eng. degree in electronics and telecommunications from the Vietnam National University (VNU) University of Engineering and Technology, Hanoi, Vietnam, in 2014, the M.Sc. degree in advanced wireless communications systems from Centrale Supélec, Paris-Saclay University, France, in 2016, and the joint Ph.D. degree in computational science from San Diego State University and the University of California at Irvine, Irvine, CA, USA, in 2022. He is currently with the University of California, Irvine and also an adjunct member of the Advanced Institute of Engineering and Technology, VNU. His research interests include wireless communications, signal processing, and machine learning. He received a Best Paper Award at the 2020 IEEE International Conference on Communications (ICC).

**Nguyen Linh Trung** obtained his B.Eng. and Ph.D. degrees, both in Electrical Engineering, from Queensland University of Technology, Brisbane, Australia, in 1998 and 2005. He joined VNU University of Engineering and Technology, Vietnam National University, Hanoi (VNU), in 2006, where he is currently an associate professor of electronic engineering.

His technical interests are theories, methods and applications of signal processing; in particular, detection and estimation, adaptive processing, subspace and tensor methods, blind separation, time-frequency analysis, graph processing, machine learning, and their applications in communications, networks and biomedicine.

He has served as technical editor-in-chief of the Journal of Research and Development on Information and Communication Technology published by the Ministry of Information and Communication of Vietnam, chair of the IEEE Signal Processing Vietnam Chapter, general chair of the 2019 International Symposium on Communication and Information Technology and the 2023 IEEE Statistical Signal Processing workshop.

**Karim Abed-Meraim** was born in 1967. He received the State Engineering Degree from Ecole Polytechnique, Palaiseau, France, in 1990, the State Engineering Degree from Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1992, the M.Sc. degree from Paris XI University, Orsay, France, in 1992 and the Ph.D. degree from the ENST in 1995 (in the field of Signal Processing and communications).

From 1995 to 1998, he took a position as a research staff at the Electrical Engineering Department of the University of Melbourne where he worked on several research project related to “Blind System Identification for Wireless Communications”, “Blind Source Separation”, and “Array Processing for Communications”, respectively.

From 1998 to 2012 he has been Assistant then Associate Professor at the Signal and Image Processing Department of Telecom-ParisTech. His research interests are in signal processing for communications, adaptive filtering and tracking, array processing and statistical performance analysis.

In September 2012 he joined the University d’Orléans (PRISME Lab.) as a full Professor.

He is the author of over 500 scientific publications including book chapters, patents, and international journal and conference papers.

Dr. Abed-Meraim is an IEEE Fellow, a past IEEE SAM-TC member, a member of the TAC on Signal Processing for Multi-sensor Systems and the TAC on Theoretical and Methodological Trends in Signal Processing (EURASIP), and a senior area editor for the IEEE Transactions on Signal Processing.