Solving the inverse problem of time independent Fokker–Planck equation with a self supervised neural network method

Wei Liu[1], Connie Khor Li Kou[1], Kun Hee Park[2], Hwee Kuan Lee[1]

1) Bioinformatics Institute, Agency for Science, Technology and Research, Singapore
2) Centre for Quantum Technologies, National University of Singapore

Abstract

The **Fokker–Planck equation** (FPE) has been used in many important applications to study **stochastic processes** with the evolution of the probability density function (pdf). In order to facilitate data driven discoveries, we propose an approach of starting with the observed pdfs to recover the FPE terms using a **self-supervised machine learning** method. This approach, known as the inverse problem, has the advantage of requiring minimal assumptions on the FPE terms. Specifically, we propose an **FPE-based neural network** (FPE-NN) which directly incorporates the FPE terms as neural network weights. By training the network on observed pdfs, we recover the FPE terms. Our experimental results on various forms of FPE show that FPE-NN can accurately recover FPE terms and denoising the pdf plays an essential role.

Problem Definition

We focus on the one-dimensional and time-independent FPE:

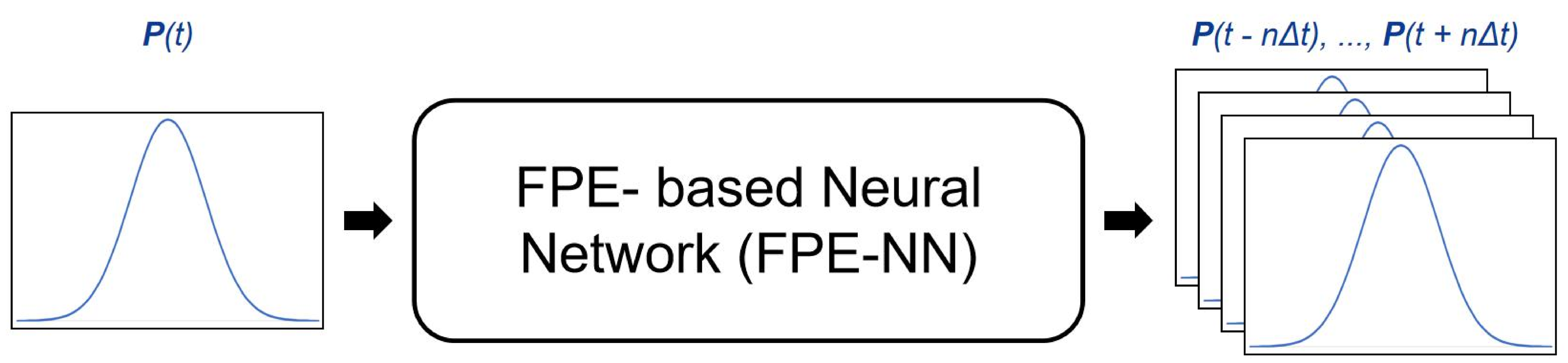
$$\frac{\partial}{\partial t}P(x,t) = \frac{\partial}{\partial x}(g(x)P(x,t)) + \frac{\partial^2}{\partial x^2}(h(x)P(x,t))$$

We study the scenario in which a pdf *P(x,t)* is measured but corrupted with noise. It is assumed to satisfy FPE but the exact form of g(x) and h(x) is unknown. In the following context, *P(x, t)*, *g(x)*, and *h(x)* are discretized over variable *x* and denoted as ***P(t)***, ***g***, and ***h***, respectively.

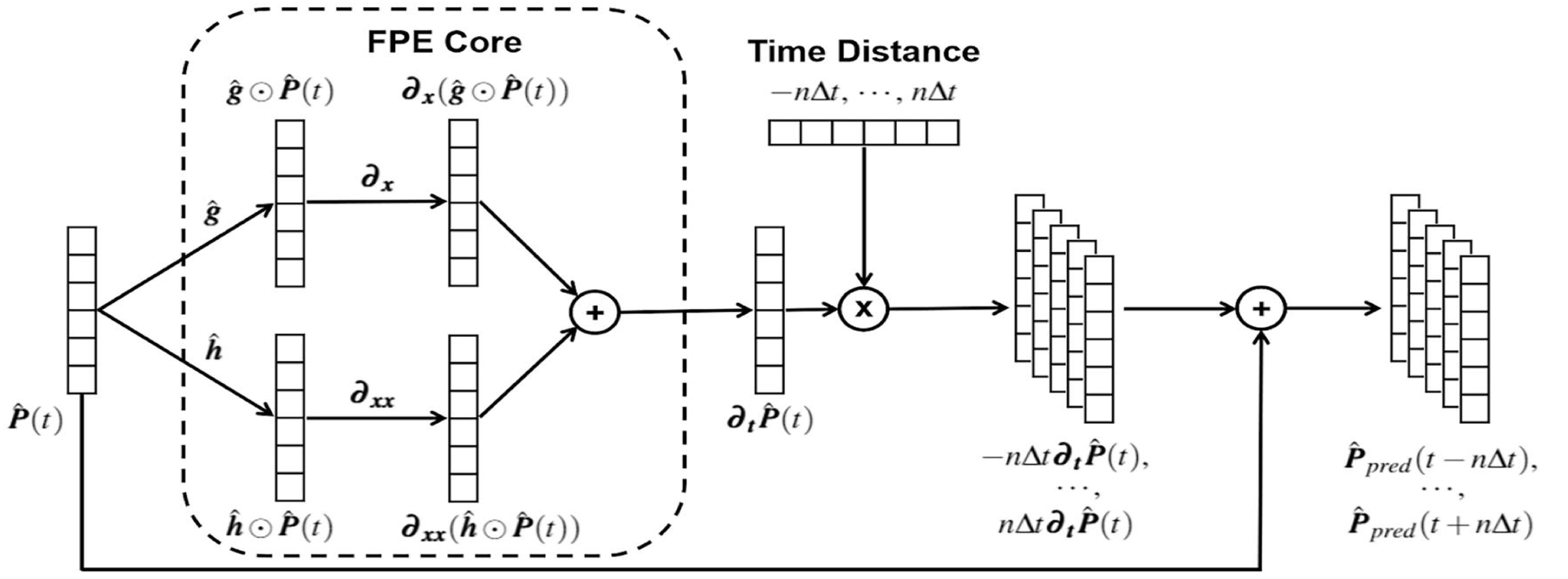
The objective is to find out ***g*** and ***h***. In order to achieve high accuracy, the measured pdf ***P(t)*** has to be denoised in the meanwhile.

Neural Network Architecture

The neural network (FPE-NN) takes ***P(t)*** at a time point as the input, and outputs ***P(t)*** at several neighboring time points.

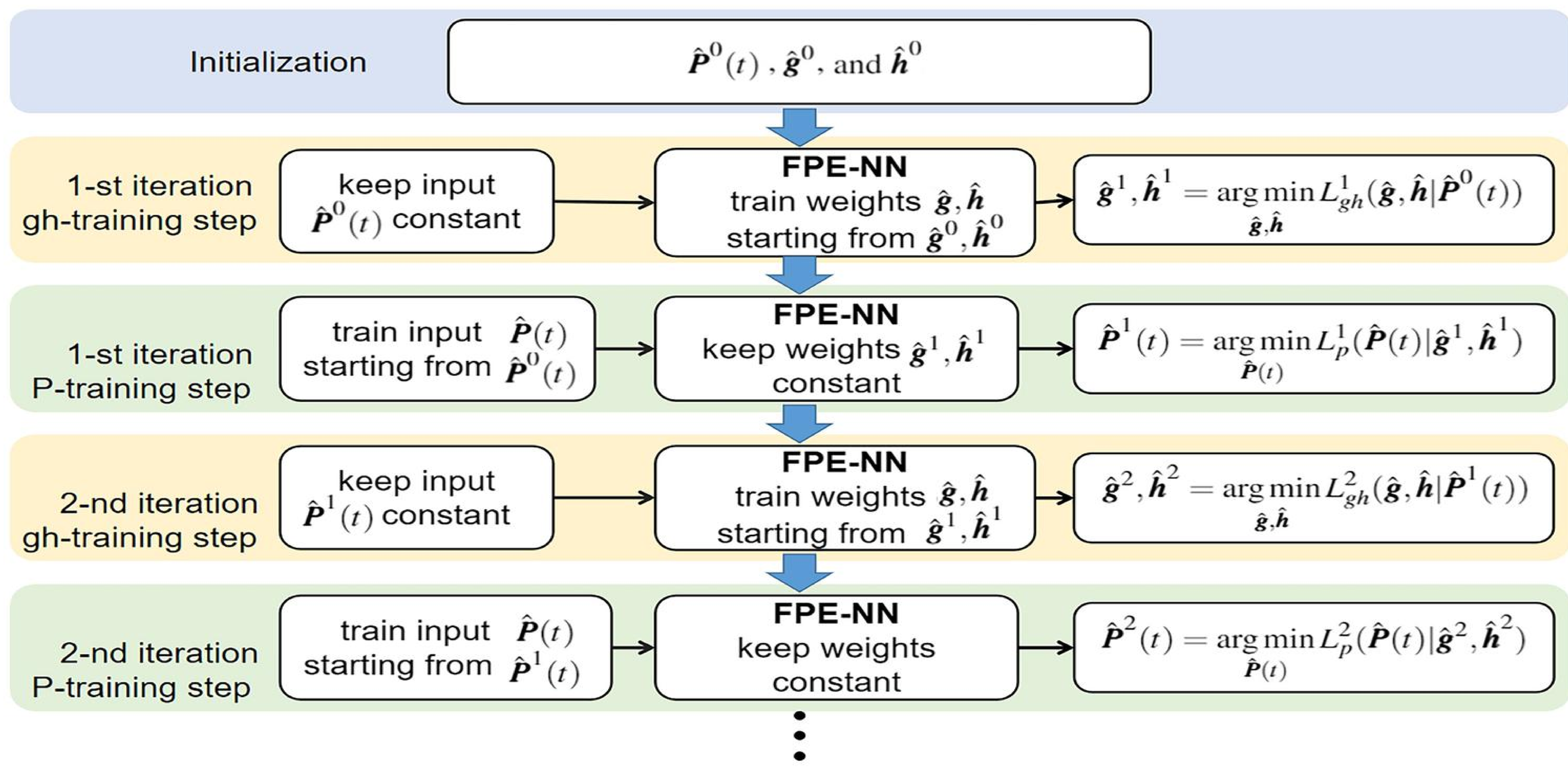


The FPE terms, ***g*** and ***h***, are directly incorporated in FPE-NN as the only trainable weights. The detailed architecture is shown below.

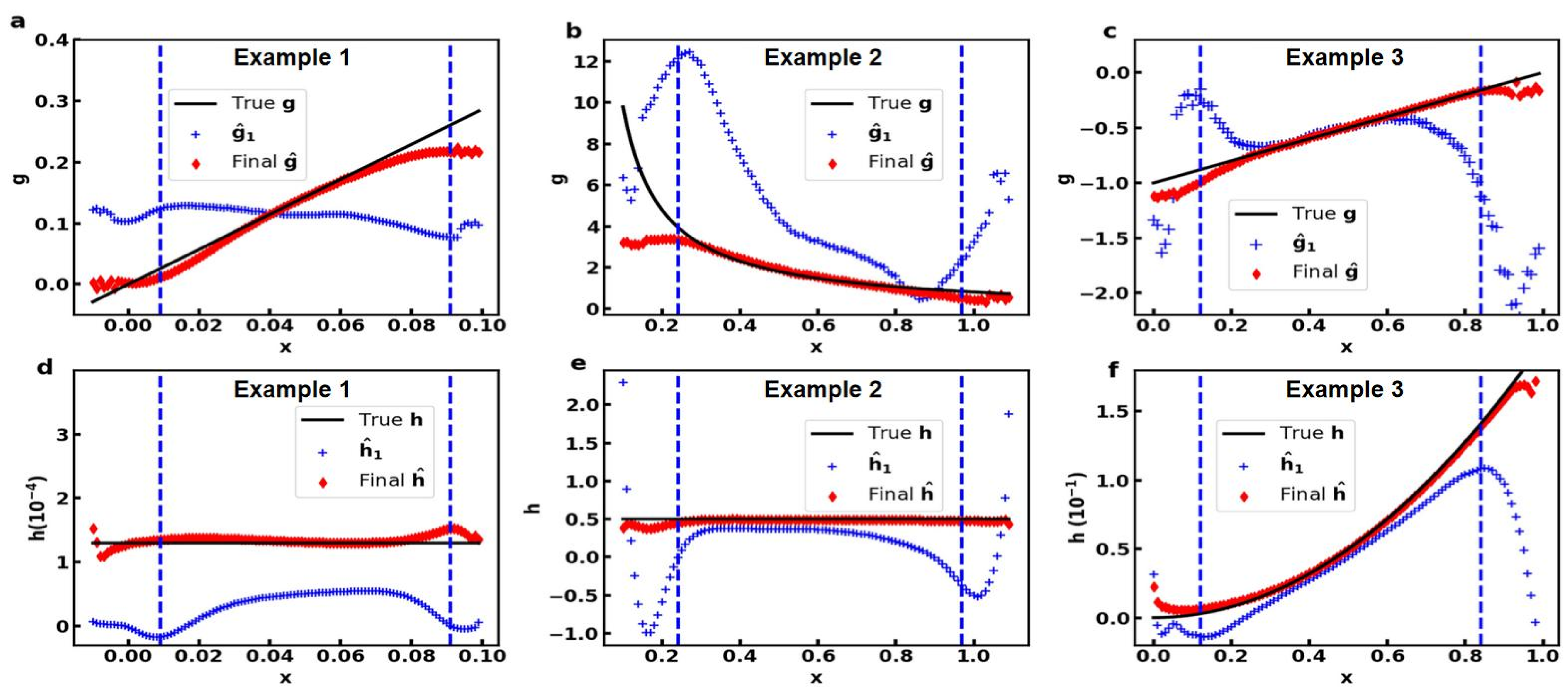


Training Process

FPE-NN is trained in an alternating way which consists of two steps in each iteration. In the gh-training step, the weights of FPE-NN (***g*** and ***h***) are trained. In the P-training step, the input ***P(t)*** is trained while the FPE-NN weights are fixed. By repeating the alternating training steps, the true values of ***P(t)***, ***g*** and ***h*** can be recovered together.



Results



Comparison of the true ***g*** and ***h*** (—) and the trained weights $\hat{g}$ and $\hat{h}$ (♦) in three examples. The boundary area which is less important is indicated by the vertical blue dash line (¦). $\hat{g}^1$ and $\hat{h}^1$(+) are the trained weights when ***P(t)*** is not denoised, as a control.

Email: liuwei@bii.a-star.edu.sg **Reference: Liu, W., et al. Sci Rep 11, 15540 (2021)** **Code: https://github.com/Gobliu/FPE-NN.git**