

# Learning-based State Reconstruction for a Scalar Hyperbolic PDE under noisy Lagrangian Sensing

Matthieu Barreau

BARREAU@KTH.SE

John Liu

JOHNLIU@KTH.SE

Karl Henrik Johansson

KALLEJ@KTH.SE

*Division of Decision and Control Systems, KTH Royal Institute of Technology Stockholm, Sweden*

## Abstract

The state reconstruction problem of a heterogeneous dynamic system under sporadic measurements is considered. This system consists of a conversation flow together with a multi-agent network modeling particles within the flow. We propose a partial-state reconstruction algorithm using physics-informed learning based on local measurements obtained from these agents. Traffic density reconstruction is used as an example to illustrate the results and it is shown that the approach provides an efficient noise rejection.

**Keywords:** state reconstruction, hyperbolic PDE, Lagrangian sensing, noise rejection, physics-informed deep learning

## 1. Introduction

Traffic jams are a source of green gas and time-consumption (Ferrara et al., 2018). One solution to lighten the effect of congestion is to control the flow of vehicles. There exist a large variety of control strategies using variable speed limits (Delle Monache et al., 2017), ramp control (Zhang et al., 2019) or autonomous vehicles (Piacentini et al., 2018; Wu et al., 2017). These laws require knowledge of road density everywhere. There comes the need for an efficient state reconstruction algorithm using measurements from fixed sensors but also capable of dealing with GPS measurements (Amin and al., 2008) and Probe Vehicles (PVs) sensing (Herrera and Bayen, 2010).

This is challenging from a theoretical point of view since such a system is modeled by a cascaded partial differential equation with a multi-agents network (C-PDE-MA) such as the one derived by meBorsche et al. (2012). There exist several kinds of algorithms to estimate the density and one can refer to the survey by Seo et al. (2017) for an overview. For fixed sensors, one can investigate the use of backstepping observers (Smyshlyaev and Krstic, 2005). However, these cannot deal effectively with nonlinearities and one has to approximate the system dynamic depending on its regime (Yu et al., 2019). Using PVs, there exist some open-loop state reconstruction algorithms proposed for instance by Delle Monache et al. (2019); Barreau et al. (2020); Čičić et al. (2020). They are copying the system and matching the boundary conditions, consequently, they are not robust to model mismatch or noise. Another recent field of research is to use machine learning tools to estimate the density (Matei et al., 2019; Huang and Agarwal, 2020; Liu et al., 2021), no matter the origin of the measurement. They are then more flexible and they lead to a good reconstruction providing substantial computer resources.

This paper goes in that latter direction and extends the preliminary observations made in Liu et al. (2021). One contribution of this article is to define a partial-state reconstruction (P-SR) ex-

pressed as the solution to an optimization problem. We show that a neural network is capable of solving the noisy state reconstruction problem asymptotically and we explain how two neural networks with a dense architecture are well-suited to that problem. We also propose a training procedure that is robust with a moderate computational burden.

The outline of the paper is as follows. In Section 2, we describe the model C-PDE-MA and give some fundamental properties about the existence and regularity of solutions. We end this section by giving a general definition of state reconstruction. Similarly to what is proposed by [Ostrometzky et al. \(2019\)](#), Section 4 is dedicated to the neural network solution with a particular focus on the numerical implementation. Finally, we consider the traffic state reconstruction problem, and simulations show a good reconstruction accuracy, even in the noisy case. The conclusion enlarges the scope of the article by providing some perspectives.

**Notation:** We define  $L^1_{loc}(\mathcal{S}_1, \mathcal{S}_2)$ ,  $L^\infty(\mathcal{S}_1, \mathcal{S}_2)$  and  $C^k(\mathcal{S}_1, \mathcal{S}_2)$  as the spaces of locally integrable functions, bounded functions and continuous functions of class  $k$  from  $\mathcal{S}_1$  to  $\mathcal{S}_2$ , respectively. A function in  $C^k_p$  is piecewise- $C^k$ . Let  $f$  be a function defined in a neighborhood of a point  $x$ , we define  $f(x^-)$  as  $\lim_{s \rightarrow x, s < x} f(s)$ . For  $g$  a function defined in a neighborhood of  $x$  with  $g(x) \neq 0$ , we write  $f = o_x(g)$  if  $\lim_{s \rightarrow x} \frac{f}{g}(s) = 0$ .

## 2. Problem Statement

This section introduces some fundamental notions about the PDE model, the ODE model, and their coupling. It is followed by the problem considered in this paper. Theoretical questions regarding the existence and regularity of solutions to these models are also briefly discussed.

### 2.1. PDE model

To define our problem, we will use the notation introduced by [Bastin and Coron \(2016\)](#). We are interested in the following scalar hyperbolic and quasi-linear PDE for  $(t, x) \in \mathbb{R}^+ \times \mathbb{R}$ :

$$\frac{\partial \rho}{\partial t}(t, x) + F(\rho(t, x)) \frac{\partial \rho}{\partial x}(t, x) = 0 \quad (1)$$

with initial condition  $\rho(0, \cdot) = \rho_0 \in L^\infty(\mathbb{R}, [0, 1])$  is piece-wise  $C^\infty$ . In physical systems,  $\rho : \mathbb{R}^+ \times \mathbb{R} \rightarrow [0, 1]$  usually refers to the normalized density. Equation (1) is then the macroscopic expression of the conservation of mass.

We assume that

1.  $F : \mathbb{R} \rightarrow \mathbb{R}$  is of class  $C^1(\mathbb{R}, \mathbb{R})$ ,
2. The flux function  $f(\rho) = \int_0^\rho F(s) ds$  is concave,
3.  $f(\rho) = V_f \rho + o_0(\rho)$ , where  $V_f > 0$  is the free flow speed.

**Proposition 1** *There exist weak solutions to this system with the following regularity:*

$$\exists T > 0, \quad \rho \in C^0(\mathbb{R}^+, L^1_{loc}(\mathbb{R}, [0, 1])) \text{ and piece-wise } C^\infty. \quad (2)$$

The regularity result comes from a direct application of Theorems 6.2.2 and 11.3.10 by [Dafermos \(2010\)](#). The fact that  $\rho \in [0, 1]$  is a consequence of the generalized characteristic methodology

(Protter and Weinberger, 2012, Chap. 4). Considering the Lax-E entropy condition  $F(\rho^+) \leq F(\rho^-)$  guarantees that the solution is unique (Dafermos, 2010, Theorem 14.10.2). Note that  $f$  is concave, hence, the entropy condition turns out to be:

$$\rho(t, x^-) \leq \rho(t, x^+). \quad (3)$$

## 2.2. Agents model

Consider a network of  $N > 1$  agents, located at  $y_i \in \mathbb{R}$ ,  $i \in \{1, \dots, N\}$  such that  $y_1(0) < \dots < y_N(0)$ . These agents are particles within a flow of local density  $\rho$  defined in (1). In other words,  $\rho(t, \cdot)$  can be seen as a probability distribution and the agents are a subset of a given realization.

We assume first that the speed  $V_i$  of agent  $i$  depends only on the density at its location and that  $V_i$  is a decreasing function of the density. Since the density might be discontinuous, we consider that the speed is the infimum in any neighborhood around its position. Because of the entropy condition (3), the velocity function writes as:

$$V_i(t) = \min \{V(\rho(t, y_i(t)^-), V(\rho(t, y_i(t)^+))\} = V(\rho(t, y_i(t)^+).$$

The agent's dynamic is then modeled by the following ordinary differential equation (ODE):

$$\dot{y}_i(t) = V(\rho(t, y_i(t)^+)) \quad \text{for } i \in \{1, \dots, N\}. \quad (4)$$

Under the entropy condition (3), one can show that the agents stay ordered in space, meaning that  $y_1(t) < \dots < y_N(t)$  for any  $t \in \mathbb{R}^+$ . Since the number of particles between agents 1 and  $N$  is  $M(t) = \int_{y_1(t)}^{y_N(t)} \rho(t, x) dx$  and a particle cannot overtake another one, the quantity  $M$  remains constant and

$$\dot{M} = V(\rho_N)\rho_N - f(\rho_N) - V(\rho_1)\rho_1 + f(\rho_1) = 0,$$

where  $\rho_i(t) = \rho(t, y_i(t))$ . The previous expression must hold for any  $\rho_1, \rho_N \in [0, 1]$  yielding:

$$V(\rho) = \begin{cases} f(\rho)\rho^{-1}, & \text{if } \rho \in (0, 1], \\ V_f, & \text{otherwise.} \end{cases} \quad (5)$$

Because  $f$  is concave and  $f(0) = 0$ , we get that  $V$  is indeed a decreasing function of the density.

**Proposition 2** *There exists a unique absolutely continuous solution  $y_i$  of (4) defined for  $t \geq 0$ .*

**Proof** From the definition of  $f$ , we get that  $V \in C^1([0, 1], \mathbb{R})$ . The proposition results from the application of Carathéodory theorem (Coddington and Levinson, 1955, Theorem 1.1, Chap 2). ■

## 2.3. C-PDE-MA model

In this paper, the network of agents is capable of sensing the PDE state at their locations. The C-PDE-MA model is illustrated in Figure 1. It is represented by a coupling between the PDE model (1) and the ODE model (4) for  $t \in \mathbb{R}^+$  and  $x \in \mathbb{R}$ :

$$\text{PDE} \begin{cases} \frac{\partial \rho}{\partial t}(t, x) + F(\rho(t, x)) \frac{\partial \rho}{\partial x}(t, x) = 0, \\ \rho(0, x) = \rho_0(x), \\ \tilde{\rho}(t) = C_{\tilde{y}(t)}(\rho(t, \cdot)) + n_\rho(t), \end{cases} \quad \text{MA} \begin{cases} \dot{\tilde{y}}(t) = V(\rho(t, \tilde{y}(t)^+)), \\ \tilde{y}(0) = \tilde{y}_0, \\ \tilde{w}(t) = \tilde{y}(t) + n_y(t), \end{cases} \quad (6)$$

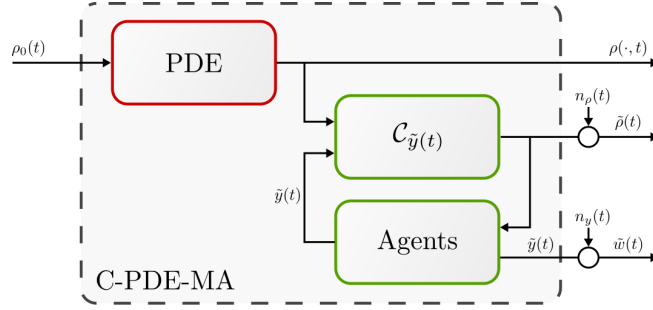


Figure 1: Block diagram of system (6).

where  $n_{\rho}, n_{\tilde{y}} \in L^{\infty}(\mathbb{R}^+, \mathbb{R}^N)$  are measurement noises,  $V$  is the element-wise function defined in (5),  $\tilde{y}(t) = [y_1(t) \cdots y_N(t)]^{\top}$  and  $\tilde{w}(t) = [w_1(t) \cdots w_N(t)]^{\top}$ . The operator  $C_{\tilde{y}} : L^1_{loc}(\mathbb{R}, [0, 1]) \rightarrow [0, 1]^N$  represents the way the agents are sensing, which ends up being:

$$C_{\tilde{y}(t)}(\rho(t, \cdot)) = [\rho(t, y_1(t)^+) \cdots \rho(t, y_N(t)^+)]^{\top}.$$

#### 2.4. Problem formulation

We are interested in this paper in the reconstruction of the PDE state  $\rho$  as expressed by the block diagram in Figure 1. Before stating the problem, some definitions are given below.

**Definition 1** Let  $\Omega \subseteq \mathbb{R}^+$ ,  $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$  be a semi-normed vector space and  $\mathcal{H}_c$  a subset of  $H^1(\Omega, \mathcal{H})$ . The *partial-reconstructed set*  $\mathcal{R}$  in  $\mathcal{H}_c$  of  $\rho \in H^1(\Omega, \mathcal{H})$  is defined as the set:

$$\mathcal{R} = \operatorname{argmin}_{\hat{\rho} \in \mathcal{H}_c} \int_{\Omega} \|\rho(t, \cdot) - \hat{\rho}(t, \cdot)\|_{\mathcal{H}}^2 dt. \quad (7)$$

We say that a signal  $\hat{\rho}$  is a *partial-state reconstruction (P-SR)* if  $\hat{\rho} \in \mathcal{R}$ .

**Remark 2** Note that this definition encapsulates the following cases:

- If  $\Omega = \infty$  and  $\rho \in \mathcal{H}_c$  then  $\hat{\rho}(\infty, \cdot) = \rho(\infty, \cdot)$  almost everywhere and this is the definition of an asymptotic state reconstruction.
- If  $\mathcal{H}_c$  is a vector space and  $\rho \notin \mathcal{H}_c$ , then the P-SR  $\hat{\rho}$  is the orthogonal projection of  $\rho$  on  $\mathcal{H}_c$ .

The state reconstruction problem implies to choose three entities: the space  $\mathcal{H}$  referring to the spatial domain of  $\rho(t, \cdot)$ ; a subset  $\mathcal{H}_c$  where we seek  $\hat{\rho}$  and the semi-norm  $\|\cdot\|_{\mathcal{H}}$  on  $\mathcal{H}$ . We will construct these three entities in the sequel.

We are working on a time window  $\Omega = [0, T]$  (we gather measurements during this time period). For the space interval, we are interested in the reconstruction between the first and the last agents. As noted in (2), we get  $\mathcal{H} = L^1_{loc}(\mathbb{R}, [0, 1])$ . In the case without noise and if we assume that the model (6) is known, we get that  $\rho \in \mathcal{H}_1$  where:

$$\mathcal{H}_1 = \{\bar{\rho} \in H^1(\Omega, \mathcal{H}) \mid C_{\tilde{y}(t)}\bar{\rho}(t, \cdot) = C_{\tilde{y}(t)}\rho(t, \cdot) \text{ for } t \in \Omega \text{ and (1) holds for } (t, x) \in \mathcal{D}_{\Omega}\}$$

with  $\mathcal{D}_{\Omega} = \{(t, x) \in \Omega \times \mathbb{R} \mid x \in [y_1(t), y_N(t)]\}$ . Note that  $\mathcal{H}_1$  is generally not a vector space. As noted by [Klibanov \(2006\)](#); [Gosse and Zuazua \(2017\)](#), even if  $T \rightarrow \infty$ ,  $\mathcal{H}_1$  is usually not a singleton

since there is no information on the initial condition. However, there always exists at least one weak solution to (1) so  $\mathcal{H}_1$  is not empty.

Adding bias on the measurements implies that  $\rho$  does not belong to  $\mathcal{H}_1$ . Consequently, it is better to use Definition 1 with  $\mathcal{H}_2 = \{\bar{\rho} \in H^1(\Omega, \mathcal{H}) \mid (1) \text{ holds for } (t, x) \in \mathcal{D}_\Omega\}$  and for  $\bar{\rho} \in H^1(\Omega, \mathcal{H})$ :

$$\|\bar{\rho}(t, \cdot)\|_{\mathcal{H}}^2 = (C_{\bar{y}(t)} \bar{\rho}(t, \cdot))^T (C_{\bar{y}(t)} \bar{\rho}(t, \cdot)). \quad (8)$$

It is difficult to solve the optimization problem (7) since the solution to (1) is usually not smooth (Bastin and Coron, 2016). One way to overcome this issue is to consider a slightly different partial differential equation:

$$\frac{\partial \rho}{\partial t}(t, x) + F(\rho(t, x)) \frac{\partial \rho}{\partial x}(t, x) = \gamma^2 \frac{\partial^2 \rho}{\partial x^2}(t, x), \quad (9)$$

where  $\gamma \in \mathbb{R}$ . This equation has many interesting properties and in particular, the solutions are smoother than in (1). Using (Amann, 1993, Theorem 14.6) with piece-wise  $C^\infty$  boundary conditions, then

$$\hat{\rho} \in C_p^1 \left( (0, T], C_p^2(\mathbb{R}, \mathbb{R}) \right). \quad (10)$$

Moreover, the vanishing viscosity method applied to this problem (Coclite and Garavello, 2013, Lemma 4.2) shows that the solution to (9) converges in a weak sense to the entropic solution to (1) as  $\gamma$  goes to zero. The set  $\mathcal{H}_c(\gamma)$  becomes then:

$$\mathcal{H}_c(\gamma) = \{\bar{\rho} \in H^1(\Omega, H^2(\mathbb{R}, \mathbb{R})) \mid (9) \text{ holds for } (t, x) \in \mathcal{D}_\Omega \text{ and } \gamma \in \mathbb{R}\}. \quad (11)$$

Using the previous definitions, we can define the partial-reconstructed set  $\mathcal{R}$ .

**Objective:** Find a P-SR of  $\rho$  from (6) in  $\mathcal{H}_c(\gamma)$  for the semi-norm in (8) with the smallest  $\gamma^2$ .

Since the set  $\mathcal{H}_c(\gamma)$  contains an equality constraint, it might be difficult to solve the optimization problem (7) with (8)-(11). Using the Lagrange multiplier  $\lambda_{\mathcal{F}} > 0$ , we propose the following relaxation:

$$\mathcal{R}_c(\gamma) = \underset{\bar{\rho} \in H^1(\Omega, H^2(\mathbb{R}, \mathbb{R}))}{\operatorname{argmin}} \left\{ \int_{\Omega} \|\rho(t, \cdot) - \bar{\rho}(t, \cdot)\|_{\mathcal{H}}^2 dt + \lambda_{\mathcal{F}} \iint_{\mathcal{D}_\Omega} \mathcal{F}_\gamma(\bar{\rho}, t, x)^2 dx dt \right\} \quad (12)$$

where  $\mathcal{F}_\gamma(\bar{\rho}, t, x) = \frac{\partial \bar{\rho}}{\partial t}(t, x) + F(\bar{\rho}(t, x)) \frac{\partial \bar{\rho}}{\partial x}(t, x) - \gamma^2 \frac{\partial^2 \bar{\rho}}{\partial x^2}(t, x)$ . It follows from the previous discussion that  $\mathcal{R} \subseteq \mathcal{R}_c(\gamma)$  and, in a weak-sense  $\mathcal{R} = \mathcal{R}_c(\gamma)$ . An element of  $\mathcal{R}_c(\gamma)$  is a P-SR of  $\rho$  solution to (6) in  $\mathcal{H}_c(\gamma)$  for the semi-norm (8). The objective rewrites then as follows.

**Rephrased objective:** Find a signal  $\hat{\rho} \in \mathcal{R}_c(\gamma)$  with the smallest  $\gamma^2$ .

To the best of the author knowledge, the P-SR solution to an hyperbolic PDE with noisy measurements coming from a network of agents has never been considered in the literature. We propose to solve this using machine learning as explained in the following section.

### 3. Learning-based State Reconstruction

This section is divided into two subsections. In the first one, we derive a P-SR problem associated to (6) without noise. In the second one, the same is discussed with addition of noisy measurements.

### 3.1. Neural network solution to the P-SR problem without noise

A neural network of  $N_n$  neurons designed for a regression problem is the mapping  $\Theta_i(\vec{X}) = \psi(b_i + \theta_i^\top \vec{X})$ , where  $b_i$  and  $\theta_i$  are the bias and the weight tensors respectively.  $\psi$  is the element-wise activation function. A densely connected deep-neural network is a neural network with  $N_L$  layers, meaning that  $\Theta = \Theta_1 \circ \Theta_2 \circ \dots \circ \Theta_{N_L}$ . This neural network is trained on the data-set  $(\vec{t}, \vec{x}, \rho(\vec{t}, \vec{x}))$ , meaning that the weights and biases are found as the arguments that locally minimize a loss function  $\mathcal{L}$ . For regression problems, the loss function is usually chosen to be  $\frac{1}{N_{\text{data}}} \sum_{k=1}^{N_{\text{data}}} (\rho(t_k, x_k) - \Theta(t_k, z_k))^2$  where  $N_{\text{data}} \in \mathbb{N}$  is the number of measurements. This is the Mean-Square Error (MSE) between the observed and estimated outputs. Note that this loss function is a discretization of the  $L^2$ -norm of the error on the whole space  $\mathcal{D}_\Omega$ .

We cannot measure points all over the domain but only on the image of the operator  $C_{\tilde{y}}$ . For  $\{t_k\}_{k=1, \dots, N_{\text{data}}}$  uniformly distributed in  $[0, T]$ ,  $\{(t_j^{\mathcal{F}}, x_j^{\mathcal{F}})\}_{j=1, \dots, N_{\mathcal{F}}}$  uniformly distributed in  $\mathcal{D}_\Omega$  and  $\lambda_1, \lambda_2, \lambda_\gamma > 0$ , an equivalent discretization of the cost function in (12) is:

$$\mathcal{L}_{\tilde{t}}(\tilde{y}, \tilde{\rho}, \Theta) = \frac{\lambda_1}{N_{\text{data}}} \sum_{k=1}^{N_{\text{data}}} \|\tilde{\rho}(t_k) - C_{\tilde{y}(t_k)} \Theta(t_k, \cdot)\|^2 + \frac{\lambda_2}{N_{\mathcal{F}}} \sum_{j=1}^{N_{\mathcal{F}}} \left( \mathcal{F}(\Theta, t_j^{\mathcal{F}}, x_j^{\mathcal{F}}) \right)^2 + \lambda_\gamma \gamma^2. \quad (13)$$

$\mathcal{L}$  is divided into three parts:

1. the first term is the traditional MSE over the measurements.
2. the second term (often called the physics MSE) does not rely on measurements but only depends on the weights and biases of the neural network, it is derived using automatic differentiation (Baydin et al., 2017) based on the dynamic of the system. As stressed in Raissi et al. (2019), it can be seen as a regularization agent, preventing over-fitting by disregarding non physical solutions and reducing the impact of the noise by enforcing the model. The choice of the weight  $\lambda_2$  depends then on the confidence we have in the model and the sparsity of the dataset.
3. the minimization of  $\gamma^2$  such that the solutions of (9) are close to the solutions of (1).

The second term of the previous objective function was introduced by Lagaris et al. (1998); Psychogios and Ungar (1992) where the authors used a neural network to provide a solution to a differential equation. It has been investigated more recently in Raissi et al. (2019) under the name "Physics-informed deep learning" where it has been extended to robust parameter identification.

Using a backpropagation algorithm, we train the neural network by locally solving the following optimization problem:

$$\mathcal{R}_{N_n} = \underset{b_1, \dots, b_{N_L}, \theta_1, \dots, \theta_{N_L}}{\operatorname{argmin}} \mathcal{L}_{\tilde{t}}(\tilde{y}, \tilde{\rho}, \Theta). \quad (14)$$

Using a neural network to solve this problem is reasonable since, as shown in Sirignano and Spiliopoulos (2018), it has a convergence property recalled below.

**Theorem 1** *In the case  $\lambda_\gamma = 0$ , the following holds:*

$$\lim_{N_n, N_{\text{data}}, N_{\mathcal{F}} \rightarrow \infty} \mathcal{L}_{\tilde{t}}(\tilde{y}, \tilde{\rho}, \Theta) = 0 \quad \text{and} \quad \lim_{N_n, N_{\text{data}}, N_{\mathcal{F}} \rightarrow \infty} \left( \inf_{\rho_1 \in \mathcal{R}_{N_n}, \rho_2 \in \mathcal{R}_c(\gamma)} \iint_{\mathcal{D}_\Omega} (\rho_1 - \rho_2)^2 \right) = 0.$$

The proof is given in the extended version of that paper and is similar to the one of Theorems 7.1 by Sirignano and Spiliopoulos (2018). The previous theorem means that  $\Theta$  in (14) is an approximation of a P-SR and if the number of neurons tends to infinity, it tends to be a P-SR in  $\mathcal{H}_c(\gamma)$ .

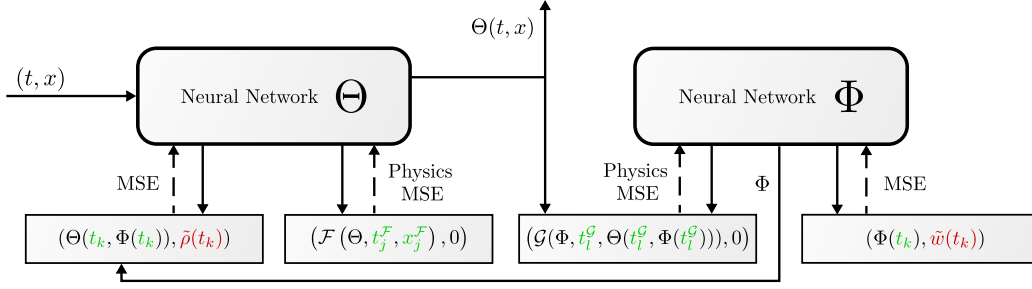


Figure 2: Block diagram of the neural network for P-SR with noise. The dashed arrows refer to values used for training the neural network. Green variables are a priori choices while the red ones are the measurements.

### 3.2. Neural network solution to the P-SR problem with noise

First, assume that  $n_\rho$  is a realization of a Gaussian distribution with mean  $\mu$  and standard deviation  $\sigma$ . The current optimization problem (14) implies the minimization of a MSE, consequently, the variance of the noise on the agents' trajectories is minimized. The second part of the cost function enforces the signal to be a solution to the noiseless model keeping a low variance as well. Such an implementation performs well and rejects the unbiased noise  $n_\rho$  (Raissi et al., 2019).

In the biased case, if  $T$  is large enough, we get  $\mathcal{L}_T(\tilde{y}, \tilde{\rho}, \rho + \varepsilon\mu) = N(\sigma^2 + (1 - \varepsilon)\mu^2) + o_0(\varepsilon)$ . Consequently,  $\varepsilon$  small and  $\mu \neq 0$  yield  $\mathcal{L}_T(\tilde{y}, \tilde{\rho}, \rho + \varepsilon\mu) < \mathcal{L}_T(\tilde{y}, \tilde{\rho}, \rho)$  and  $\rho$  is not a local minimizer. One solution to this issue is to consider instead the loss function  $\mathcal{L}_T(\tilde{y}, \tilde{\rho} - \tilde{n}_\rho, \Theta)$  where  $\tilde{n}_\rho \in \mathbb{R}^N$ .

To deal with the noise  $n_y$ , the neural network architecture has to be modified. In that case, one need to propose a similar architecture with some physics cost on  $\tilde{w}$ , leading to the diagram in Figure 2. There are then two neural networks that are trained simultaneously.  $\Theta$  is the same as in the previous part and the second one,  $\Phi$ , estimates  $\tilde{y}$ . Naturally, the loss function is:

$$\tilde{\mathcal{L}}_T(\tilde{w}, \tilde{\rho}) = \mathcal{L}_T(\Phi, \tilde{\rho} - \tilde{n}_\rho, \Theta) + \sum_{k=1}^{N_{\text{data}}} \frac{\lambda_3 \|\Phi(t_k) - \tilde{w}(t_k)\|^2}{N_{\text{data}} \times N} + \sum_{l=1}^{N_{\mathcal{G}}} \frac{\lambda_4 \left\| \mathcal{G}(\Phi, t_l^{\mathcal{G}}, \Theta(t_l^{\mathcal{G}}), \Phi(t_l^{\mathcal{G}})) \right\|^2}{N_{\mathcal{G}} \times N}$$

where  $\mathcal{L}_T$  is defined in (13),  $\tilde{n}_\rho \in \mathbb{R}^N$ ,  $\lambda_3, \lambda_4 > 0$  and  $\mathcal{G}(\tilde{y}, t, \tilde{\rho}) = \dot{\tilde{y}}(t) - V(\tilde{\rho}(t))$ .

**Remark 3** *The physics cost related to the trajectories of the agents is adding some artificial boundary data. Consequently, the measurements can be sparser for a similar trajectory estimation.*

### 3.3. Numerical implementation

#### 3.3.1. NETWORK ARCHITECTURE

It is well-known that some solutions of (1) are discontinuous while solutions of (9) are usually not. Nevertheless, they may have fast variations. The neural network  $\Theta$  must handle both shock and rarefaction waves while  $\Phi$  represents an absolutely continuous function.

$\Phi$  takes one input (the time  $t$ ) and has  $N$  outputs (the positions of the agents). Since the solution is absolutely continuous (it is piece-wise  $C^1$ ), the best is to combine some ReLu and tanh activation functions. To keep a simple structure, one can then consider two dense neural networks with different activation functions such as ReLu and tanh. They are summed and weighed at the end to give the final output. There are 3 hidden layers, each with  $2N$  neurons for each sub-network.

For  $\Theta$ , the architecture is more complex. Since we are considering a viscous PDE, the solutions are continuous and using the tanh as an activation function seems a good choice for several reasons:

1. A dense neural network with tanh activation functions can deal with almost-shock waves since  $\Theta(t, x) = \tanh(kT_s(t, x))$  where  $T_s$  is the trajectory of a shock wave. If  $k$  is chosen very large, then there is a very steep slope along the curve  $T_s(t, x) = 0$ .
2. It can deal with rarefaction waves as well since in this case the solution is  $F^{-1}(x/t)$  (Evans, 1998, Section 11.2) which is continuous since  $F$  is  $C^1$  and  $F' < 0$ .

Consequently, the number of nodes can be related to the number of waves in the solution. On the contrary, the number of layers deals with the inference between these waves. In conclusion, the number of nodes is related to the length-space while the number of layers reflects the time-window. Choromanska et al. (2015) showed that having a neural network with many variables usually leads to better convergence. As said previously, the neural network should not overfit the data because of the regularization agent so we can consider a large network with  $8T/100$  layers made up of  $20L/7000$  neurons each.

### 3.3.2. TRAINING PROCEDURE

First of all, since the space and time intervals are not of the same order of magnitude, it is better to standardize the dataset before training. Indeed, that leads to a faster convergence of the optimization procedure to a more optimal solution (Goodfellow et al., 2016).

Similarly to what was done in Raissi et al. (2019); Lagaris et al. (1998); Sirignano and Spiliopoulos (2018), the best solver for this problem seems to be the BFGS algorithm (Nocedal and Wright, 2006, Chap 6). Nevertheless, it is quite slow so one can consider speeding it up with a pre-training using Adam (Kingma and Ba, 2015). Another issue comes with the choice of the different  $\lambda$ . It appears that the effect of this choice decreases when we split the training into several steps, leading to the following procedure:

1. Let  $\lambda_1 = \lambda_2 = \lambda_\gamma = 0$  and  $\lambda_3 = 1, \lambda_4 = 0.5$ . We then estimate precisely the trajectories of the agents and we modify the value of the density along these curves. We train using the BFGS algorithm with low requirements such that the convergence is rapidly obtained.
2. Then, we train the value of the density without modifying the trajectories. For that, we fix the  $\Phi$  neural network and we use first Adam and then BFGS with  $\lambda_1 = 1, \lambda_2 = 0.1, \lambda_\gamma = \lambda_3 = 0$  and  $\lambda_4 = 0.25$ . This step takes more time and fixes the values of the density at its boundaries while starting to enforce the physics of the PDE inside the domain.
3. Finally, we optimize using BFGS only and  $\lambda_1 = 1, \lambda_2 = 1, \lambda_\gamma = 0.1, \lambda_3 = 1$  and  $\lambda_4 = 0.5$ . We focus on this step on the noise reduction and we minimize the coupled system proposed in Figure 2. Since the trajectories are close to their optimal values, this step changes mostly the values of the density inside the domain, trying to keep a low value of  $\gamma$ .

This procedure seems, in general, to be a good starting point for such a problem since it follows the iterative procedure described in Bertsekas (2014). The weights values have to be designed in each case and the one used here leads to the lowest computed generalization error in our case.



## 4. Traffic flow theory and simulation results

### 4.1. Traffic flow theory

The simplest model for traffic flow on a highway is given by the continuity equation, which is nothing more than the conservation of mass (Evans, 1998, Chapter 11). This model has been derived by Lighthill and Whitham (1955); Richards (1956) and it results in the so-called LWR model:

$$\frac{\partial \rho}{\partial t}(t, x) + \frac{\partial \rho}{\partial x}(\rho V)(t, x) = 0, \quad (t, x) \in \mathbb{R}^+ \times \mathbb{R} \quad (15)$$

where  $\rho : \mathbb{R}^+ \times \mathbb{R} \rightarrow [0, 1]$  is the normalized density of cars. It can be represented as the probability distribution of a car at a given position  $x$  at time  $t$ .  $V : [0, 1] \rightarrow \mathbb{R}^+$  stands for the equilibrium velocity of a car depending on the density around. The LWR model assumes, similarly to Greenshields et al. (1935), that  $V = V_f(1 - \rho)$ , where  $V_f > 0$  is the free flow speed. With this choice, (15) transforms into the following scalar hyperbolic PDE for  $(t, x) \in \mathbb{R}^+ \times \mathbb{R}$ :

$$\frac{\partial \rho}{\partial t}(t, x) + V_f(1 - 2\rho(t, x)) \frac{\partial \rho}{\partial x}(t, x) = 0. \quad (16)$$

In the context of traffic flow, we are considering that there are some vehicles – called Probe Vehicles (PV) – capable of sensing the density around them using the distances to their neighbors, cameras (Chellappa et al., 2004; Kang and Kum, 2020) or radar sensors (Sun et al., 2006). The state reconstruction problem without noise is solved in finite-time by Delle Monache et al. (2019); Barreau et al. (2020) for instance. The results in our case are shown in the following subsection. A full study on this example is conducted in Liu et al. (2021).

### 4.2. Simulation results

We are studying here the P-SR problem of (16) using probe vehicles. The real density is obtained from a Godunov scheme (LeVeque, 1992) with random initial and boundary conditions. From the GitHub repository (Barreau and Liu, 2020), one can see that shocks and rarefaction waves are making the reconstruction difficult. The reconstruction is evaluated in two cases with a neural network coded using Tensorflow 1.15 (Abadi and al., 2015). The first scenario considers the system without noise. The second case contemplates  $n_\rho$  as an unbiased Gaussian noise with a standard deviation of  $\sigma_\rho = 0.2$  and  $n_y$  as a Brownian motion generated from a normal distribution with zero mean and a variance  $\sigma_y = 2$ . 20 P-SR are done using a naive BFGS Raissi et al. (2019); Sirignano and Spiliopoulos (2018) and 20 others using the training procedure described earlier. The computations are done on a Intel Core i5-8365 CPU @ 1.60GHz and 16 GB of RAM computer. The computation time and the normalized error are displayed in Figure 3.

One can see that the reconstruction has a small generalization error in general. The error is coming from discrepancies for small-time  $t$  because of the effect of the unknown initial condition. But, after a while, the reconstruction error is close to zero between the PVs except on the shock trajectories where the neural network is unable of reproducing the steep variation. The modified training has a larger computation time (10 times larger in average) but a smaller and more consistent generalization error (4 times smaller in average with a standard deviation is 3 times smaller). That means the modified training procedure is indeed more accurate and robust to the neural network initialization at the price of a much larger computation time.

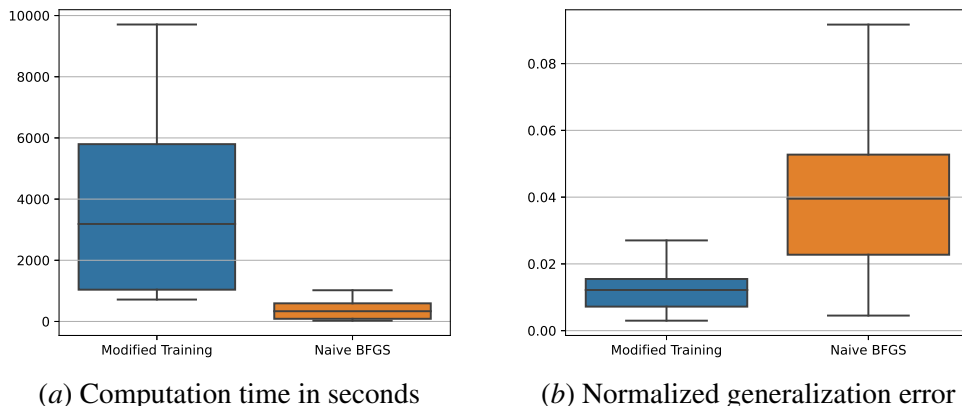


Figure 3: Computation time and error depending on the training algorithm.

## 5. Conclusion & Perspectives

In this paper, we proposed the partial-state reconstruction problem of a hyperbolic equation using only Lagrangian measurements and we applied it to the special example of traffic flow estimation. To that extend, we used physics-informed deep learning to counterbalance the small number of measurements and increase the robustness. The simulations showed that noise was indeed rejected.

This is preliminary work and we propose many perspectives. First of all, the neural network can be improved by considering a more specific architecture (Hochreiter and Schmidhuber, 1997). One can also look into the direction of a decentralized reconstruction algorithm, that would effectively reduce the computational burden while increasing the sensitivity to noise. The roles of the  $\lambda$  have not been explored in the paper, however, it appears that the speed of convergence highly depends on these values. A theoretical study on the role of these multipliers should be done to get a better understanding of the optimization process. Finally, one can also consider second-order models which are more adapted to traffic flow (Aw and Rascle, 1995; Zhang, 2002).

## Acknowledgments

The research leading to these results is partially funded by the KAUST Office of Sponsored Research under Award No. OSR-2019-CRG8-4033, the Swedish Foundation for Strategic Research, the Swedish Research Council and Knut and Alice Wallenberg Foundation.

## References

- M. Abadi and al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- H. Amann. Nonhomogeneous linear and quasilinear elliptic and parabolic boundary value problems. In *Function spaces, differential operators and nonlinear analysis*, pages 9–126. Springer, 1993.
- S. Amin and al. Mobile century using GPS mobile phones as traffic sensors: A field experiment. In *15th World Congress on Intelligent Transportation Systems*, 2008.

- A. Aw and M Rascle. Resurrection of "second order" models of traffic flow. *SIAM J. Appl. Math.*, 60(3):916–938, 1995.
- M. Barreau and J. Liu. Learning-based traffic reconstruction using probe vehicles. <https://github.com/mBarreau/TrafficReconstructionDL>, 2020.
- M. Barreau, A. Selivanov, and K. H. Johansson. Dynamic traffic reconstruction using probe vehicles. In *Conference on Decision and Control (CDC)*, Jeju Island, Republic of Korea, 2020.
- G. Bastin and J.-M. Coron. *Stability and boundary stabilization of 1-d hyperbolic systems*, volume 88. Springer, 2016.
- A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.
- Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- R. Borsche, R. M. Colombo, and M. Garavello. Mixed systems: Odes–balance laws. *Journal of Differential equations*, 252(3):2311–2338, 2012.
- R. Chellappa, G. Qian, and Q. Zheng. Vehicle detection and tracking using acoustic and video sensors. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–793. IEEE, 2004.
- A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multi-layer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.
- G. M. Coclite and M. Garavello. Vanishing viscosity for mixed systems with moving boundaries. *Journal of Functional Analysis*, 264(7):1664–1710, 2013.
- E. A. Coddington and N. Levinson. *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955.
- C. M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics; 3rd ed.* Grundlehren der mathematischen Wissenschaften. Springer, Dordrecht, 2010. doi: 10.1007/978-3-642-04048-1.
- M. L. Delle Monache, B. Piccoli, and F. Rossi. Traffic regulation via controlled speed limit. *SIAM Journal on Control and Optimization*, 55(5):2936–2958, 2017.
- M. L. Delle Monache, T. Liard, B. Piccoli, R. Stern, and D. Work. Traffic reconstruction using autonomous vehicles. *SIAM Journal on Applied Mathematics*, 79(5):1748–1767, 2019.
- L. C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998. ISBN 9780821807729.
- A. Ferrara, S. Sacone, and S. Siri. *Freeway Traffic Modelling and Control*. Springer Nature, 2018.
- I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

- L. Gosse and E. Zuazua. Filtered gradient algorithms for inverse design problems of one-dimensional burgers equation. In *Innovative algorithms and analysis*, pages 197–227. Springer, 2017.
- B. D. Greenshields, J. R. Bibbins, W. S. Channing, and H. H. Miller. A study of traffic capacity. In *Highway research board proceedings*, volume 14. National Research Council (USA), Highway Research Board, 1935.
- J. C. Herrera and A. M. Bayen. Incorporation of lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological*, 44(4):460–481, 2010.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- J. Huang and S. Agarwal. Physics informed deep learning for traffic state estimation. In *2020 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020.
- D. Kang and D. Kum. Camera and radar sensor fusion for robust vehicle localization via vehicle part localization. *IEEE Access*, 8:75223–75236, 2020.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*, 2015.
- M. V. Klibanov. Estimates of initial conditions of parabolic equations and inequalities via lateral Cauchy data. *Inverse problems*, 22(2):495, 2006.
- I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.
- R. J. LeVeque. *Numerical methods for conservation laws*, volume 3. Springer, 1992.
- M. J. Lighthill and G. B. Whitham. On kinematic waves II. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955.
- J. Liu, M. Barreau, M. Čičić, and K. H. Johansson. Learning-based traffic state reconstruction using probe vehicles. In *16th IFAC Symposium on Control in Transportation Systems (CTS)*, 2021.
- I. Matei, C. Mavridis, J. S. Baras, and M. Zhenirovskyy. Inferring particle interaction physical models and their dynamical properties. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4615–4621. IEEE, 2019.
- J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- J. Ostrometzky, K. Berestizshevsky, A. Bernstein, and G. Zussman. Physics-informed deep neural network method for limited observability state estimation. *arXiv preprint arXiv:1910.06401*, 2019.
- G. Piacentini, P. Goatin, and A. Ferrara. Traffic control via moving bottleneck of coordinated vehicles. *IFAC-PapersOnLine*, 51(9):13 – 18, 2018. ISSN 2405-8963. 15th IFAC Symposium on Control in Transportation Systems CTS 2018.

- M. H. Protter and H. F. Weinberger. *Maximum principles in differential equations*. Springer Science & Business Media, 2012.
- D. C. Psychogios and L. H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.
- M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- P. I. Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.
- T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura. Traffic state estimation on highway: A comprehensive survey. *Annual Reviews in Control*, 43:128–151, 2017.
- J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.
- A. Smyshlyaev and M. Krstic. Backstepping observers for a class of parabolic PDEs. *Systems & Control Letters*, 54(7):613–625, 2005.
- Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.
- M. Čičić, M. Barreau, and K. H. Johansson. Numerical investigation of traffic state reconstruction and control using connected automated vehicles. In *23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020.
- C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*, page 10, 2017.
- H. Yu, A. M. Bayen, and M. Krstic. Boundary observer for congested freeway traffic state estimation via Aw-Rascle-Zhang model. *IFAC-PapersOnLine*, 52(2):183 – 188, 2019. 3rd IFAC Workshop on Control of Systems Governed by Partial Differential Equations.
- H.M. Zhang. A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B*, 36:275–290, 2002.
- L. Zhang, C. Prieur, and J. Qiao. Pi boundary control of linear hyperbolic balance laws with stabilization of arz traffic flow models. *Systems & Control Letters*, 123:85–91, 2019.