Reconstruction of Traffic Flow with Learning Methods

Nail Baloul, Amaury Hayat, Thibault Liard, Pierre Lissy

CERMICSxTED - Parent, Auvergne

April, 3rd 2025

Outline

1. Introduction
2. Existing Traffic Models
3. Optimization (Learning-Based) for Traffic Flow Reconstruction
4. Conclusion and Perspectives

Table of Contents

1. Introduction
2. Existing Traffic Models
3. Optimization (Learning-Based) for Traffic Flow Reconstruction
4. Conclusion and Perspectives

Traffic State Reconstruction Approaches

- Model-Based Method
  - Uses microscopic and macroscopic models
  - Provides theoretical guarantees
  - Struggles to capture real-world complexities

Traffic State Reconstruction Approaches

- Model-Based Method
  - Uses microscopic and macroscopic models
  - Provides theoretical guarantees
  - Struggles to capture real-world complexities
- Data-Driven Method
  - Learns patterns directly from measurement data
  - Derives system properties or predicts near-future states
  - Requires extensive data for effectiveness

Traffic State Reconstruction Approaches

- Model-Based Method
  - Uses microscopic and macroscopic models
  - Provides theoretical guarantees
  - Struggles to capture real-world complexities
- Data-Driven Method
  - Learns patterns directly from measurement data
  - Derives system properties or predicts near-future states
  - Requires extensive data for effectiveness
- Our Approach
  - Combines models and data to address sparsity and improve realism
  - **Integrates physical priors with data observations**
  - Achieves reliable traffic reconstruction with limited observations

Table of Contents

1 Introduction

2 Existing Traffic Models

3 Optimization (Learning-Based) for Traffic Flow Reconstruction

4 Conclusion and Perspectives

- Follow-the-Leader (FtL), **microscopic** first order model
  $\Rightarrow$ dynamics of each vehicle depend on vehicle immediately in front

$$\begin{cases} \dot{x}_N^N(t) = v_{\max}, & t > 0, \\ \dot{x}_i^N(t) = v\left(\frac{L}{N(x_{i+1}^N(t) - x_i^N(t))}\right), & t > 0,, \quad i = 0, \cdots, N-1 \\ x_i^N(0) = \bar{x}_i, & i = 0, \cdots, N \end{cases} \tag{1}$$

$\Rightarrow$ accurate traffic representation, encodes individual movements
$\Rightarrow$ computationally demanding, requires more data

- Follow-the-Leader (FtL), **microscopic** first order model
  $\Rightarrow$ dynamics of each vehicle depend on vehicle immediately in front

$$\begin{cases} \dot{x}_N^N(t) = v_{\max}, & t > 0, \\ \dot{x}_i^N(t) = v\left(\frac{L}{N(x_{i+1}^N(t) - x_i^N(t))}\right), & t > 0,, \quad i = 0, \cdots, N-1 \\ x_i^N(0) = \bar{x}_i, & i = 0, \cdots, N \end{cases} \tag{1}$$

  $\Rightarrow$ accurate traffic representation, encodes individual movements
  $\Rightarrow$ computationally demanding, requires more data

- Lighthill-William-Richards (LWR), **macroscopic** traffic flow model
  $\Rightarrow$ vehicles treated as a continuous medium similar to particles in fluid
  $\Rightarrow$ hyperbolic conservation law

$$\begin{cases} \rho_t(x,t) + (f(\rho(x,t)))_x = 0, & x \in \mathbb{R}, \quad t > 0, \\ \rho(x,0) = \bar{\rho}(x), & x \in \mathbb{R} \end{cases} \tag{2}$$

  $\Rightarrow$ faster implementation, less data-intensive
  $\Rightarrow$ overlooks traffic heterogeneity, oversimplifies traffic phenomena

- **Convergence analysis of FtL approximation scheme towards LWR model**[1]
- Link between FtL and LWR based on atomization of initial density $\bar{\rho}$

$$\bar{x}_{i+1} := \sup \left\{ x \in \mathbb{R} : \int_{\bar{x}_i}^{x} \bar{\rho}(y) dy = \frac{L}{N} \right\}, \quad i = 0, \cdots, N-1 \tag{3}$$

[1] **holden_continuum_2017**.
[2] **francesco_rigorous_2015**.

- **Convergence analysis of FtL approximation scheme towards LWR model**[1]
- Link between FtL and LWR based on atomization of initial density $\bar{\rho}$

$$\bar{x}_{i+1} \coloneqq \sup \left\{ x \in \mathbb{R} : \int_{\bar{x}_i}^{x} \bar{\rho}(y)dy = \frac{L}{N} \right\}, \quad i = 0, \cdots, N-1 \tag{3}$$

- Solution of PDE (1) can be recovered as many particle limit[2] of ODE system (14)

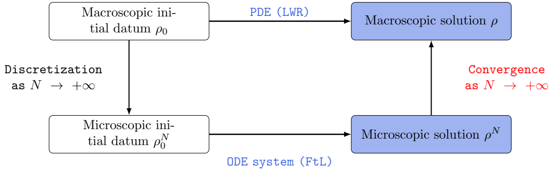

Figure: Coupled Resolution of a Microscopic ODE System and a Macroscopic PDE

[1] **holden_continuum_2017**.
[2] **francesco_rigorous_2015**.

Coupled Micro-Macro Reconstruction models

- Hybrid micro-macro models explored in traffic density reconstruction[3]

$$
\begin{cases}
\dot{x}_N(t) = v_{\max}, & t > 0, \\
\dot{x}_i(t) = v\left(\rho(t, x_i(t))\right), & t > 0,, \quad i = 0, \cdots, N-1 \\
\rho_t + \left(f(\rho)\right)_x = \gamma^2 \rho_{xx}, & x \in \mathbb{R}, \quad t > 0,
\end{cases}
\tag{4}
$$

[3] **barreau_physics-informed_2021**.
[4] **liu_learning-based_2020**.

- Hybrid micro-macro models explored in traffic density reconstruction[3]

$$\begin{cases} \dot{x}_N(t) = v_{\max}, & t > 0, \\ \dot{x}_i(t) = v\left(\rho(t, x_i(t))\right), & t > 0,, \quad i = 0, \cdots, N-1 \\ \rho_t + \left(f(\rho)\right)_x = \gamma^2 \rho_{xx}, & x \in \mathbb{R}, \quad t > 0, \end{cases} \tag{4}$$

- **Partial state reconstruction**[4] using measurements from probe vehicles (PVs)
  $\Rightarrow$ low penetration rate $N_{\text{probes}} \ll N_{\text{total}}$
  $\Rightarrow$ recover density ρ from **limited** trajectories

[3] **barreau_physics-informed_2021**.
[4] **liu_learning-based_2020**.

Coupled Micro-Macro Reconstruction models

- Hybrid micro-macro models explored in traffic density reconstruction[3]

$$\begin{cases} \dot{x}_N(t) = v_{\max}, & t > 0, \\ \dot{x}_i(t) = v\left(\rho(t, x_i(t))\right), & t > 0,, \quad i = 0, \cdots, N-1 \\ \rho_t + \left(f(\rho)\right)_x = \gamma^2 \rho_{xx}, & x \in \mathbb{R}, \quad t > 0, \end{cases} \tag{4}$$

- **Partial state reconstruction**[4] using measurements from probe vehicles (PVs)
  $\Rightarrow$ low penetration rate $N_{\text{probes}} \ll N_{\text{total}}$
  $\Rightarrow$ recover density ρ from **limited** trajectories
- Requires access to real-time positions, densities and instantaneous speeds of PVs

[3] **barreau_physics-informed_2021**.
[4] **liu_learning-based_2020**.

Table of Contents

1 Introduction

2 Existing Traffic Models

3 Optimization (Learning-Based) for Traffic Flow Reconstruction

4 Conclusion and Perspectives

Parametrized Microscopic Model

- **More limited data scenario** $\Rightarrow$ only **initial and final positions** of PVs available

Parametrized Microscopic Model

- **More limited data scenario** $\Rightarrow$ only **initial and final positions** of PVs available
- **Enhanced FtL scheme** (1)
  $\Rightarrow$ α accounts for unobserved vehicles between consecutive PVs
  $\Rightarrow$ adjusts dynamics and allows varying levels of response
  $\Rightarrow$ bridges discrete (vehicle-level) dynamics to continuous (density-level) dynamics

Parametrized Microscopic Model

- **More limited data scenario** $\Rightarrow$ only **initial and final positions** of PVs available
- **Enhanced FtL scheme** (1)
  - $\Rightarrow$ α accounts for unobserved vehicles between consecutive PVs
  - $\Rightarrow$ adjusts dynamics and allows varying levels of response
  - $\Rightarrow$ bridges discrete (vehicle-level) dynamics to continuous (density-level) dynamics
- Parametrized ODE system

$$\begin{cases} \dot{x}_n^N(t) = v_{\max}, & t \in (0, T] \\ \dot{x}_i^N(t) = v\left(\rho_i^N(t)\right), & t \in (0, T] \quad i = 0, \cdots, n-1 \\ x_i^N(0) = \bar{x}_i, & i = 0, \cdots, n \end{cases} \tag{5}$$

$\Rightarrow$ local discrete densities

$$\rho_i^N(t) := \frac{\alpha_i L}{N\left(x_{i+1}^N(t) - x_i^N(t)\right)}, \quad t \in (0, T], \quad i = 0, \cdots, n-1, \tag{6}$$

Parametrized Microscopic Model

- **More limited data scenario** $\Rightarrow$ only **initial and final positions** of PVs available
- **Enhanced FtL scheme** (1)
  $\Rightarrow$ α accounts for unobserved vehicles between consecutive PVs
  $\Rightarrow$ adjusts dynamics and allows varying levels of response
  $\Rightarrow$ bridges discrete (vehicle-level) dynamics to continuous (density-level) dynamics
- Parametrized ODE system

$$\begin{cases} \dot{x}_n^N(t) = v_{\max}, & t \in (0, T] \\ \dot{x}_i^N(t) = v\left(\rho_i^N(t)\right), & t \in (0, T] \quad i = 0, \cdots, n-1 \\ x_i^N(0) = \bar{x}_i, & i = 0, \cdots, n \end{cases} \tag{5}$$

$\Rightarrow$ local discrete densities

$$\rho_i^N(t) := \frac{\alpha_i L}{N\left(x_{i+1}^N(t) - x_i^N(t)\right)}, \quad t \in (0, T], \quad i = 0, \cdots, n-1, \tag{6}$$

- Assumptions on velocity

$$v \in C^1([0, +\infty)) \tag{7a}$$

$$v \text{ is decreasing on } [0, +\infty) \tag{7b}$$

$$v(0) = v_{\max} < \infty \tag{7c}$$

Well Posedness

- Local existence and uniqueness of solution to (5) (for fixed α) via Picard-Lindelöf

Well Posedness

- Local existence and uniqueness of solution to (5) (for fixed α) via Picard-Lindelöf
- Condition on intial positions $\bar{x}_0 < \bar{x}_1 < \cdots < \bar{x}_{n-1} < \bar{x}_n$
  $\Rightarrow$ global existence

Lemma (Discrete maximum principle)

For solution $x(t)$ of (5) with v satisfying (7a)-(7c), for all $i = 0, \cdots, n-1$,

$$\frac{\alpha_i L}{NM} \leq x_{i+1}(t) - x_i(t) \leq \bar{x}_n - \bar{x}_0 + (v_{\max} - v(M))\, t, \quad \forall t \in [0, T], \tag{8}$$

where $M := \max_i \left(\frac{\alpha_i L}{N(\bar{x}_{i+1} - \bar{x}_i)} \right)$

- Local existence and uniqueness of solution to (5) (for fixed α) via Picard-Lindelöf
- Condition on intial positions $\bar{x}_0 < \bar{x}_1 < \cdots < \bar{x}_{n-1} < \bar{x}_n$
  $\Rightarrow$ global existence

Lemma (Discrete maximum principle)

For solution $x(t)$ of (5) with v satisfying (7a)-(7c), for all $i = 0, \cdots, n-1$,

$$\frac{\alpha_i L}{NM} \leq x_{i+1}(t) - x_i(t) \leq \bar{x}_n - \bar{x}_0 + (v_{\max} - v(M))\, t, \quad \forall t \in [0, T], \tag{8}$$

where $M := \max_i \left(\frac{\alpha_i L}{N(\bar{x}_{i+1} - \bar{x}_i)} \right)$

- Piecewise constant Eulerian discrete density

$$\rho^N(t,x) := \sum_{i=0}^{N-1} \rho_i^N(t) \chi_{[x_i(t), x_{i+1}(t))}(x), \quad x \in \mathbb{R}, \quad t \in [0, T], \tag{9}$$

$\Rightarrow$ discrete approximation of solution to LWR model (14)

- **Approximate density reconstruction** $\Rightarrow$ find optimal interaction parameter α

$$
\begin{aligned}
& \underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\left\|x(T)-\bar{y}\right\|^2 \\
\text{s.t.} \quad & \dot{x}(t) = V\left(W_\alpha x(t) + b_\alpha(t)\right) \\
& x(0) = \bar{x} \\
& \sum_{i=0}^{n-1} \alpha_i = N, \quad \alpha \in \mathbb{R}_+^n
\end{aligned} \tag{10}
$$

- **Approximate density reconstruction** $\Rightarrow$ find optimal interaction parameter α

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\left\|x(T)-\bar{y}\right\|^2 \\ \text{s.t.} \quad & \dot{x}(t)=V\left(W_\alpha x(t)+b_\alpha(t)\right) \\ & x(0)=\bar{x} \\ & \sum_{i=0}^{n-1}\alpha_i=N, \quad \alpha\in\mathbb{R}_+^n \end{aligned} \tag{10}$$

- Existence of solutions guaranteed by assumptions on v (continuity) and constraints on α (compactness)

- **Approximate density reconstruction** $\Rightarrow$ find optimal interaction parameter α

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\left\|x(T)-\bar{y}\right\|^2 \\ \text{s.t.} \quad & \dot{x}(t) = V\left(W_\alpha x(t) + b_\alpha(t)\right) \\ & x(0) = \bar{x} \\ & \sum_{i=0}^{n-1} \alpha_i = N, \quad \alpha \in \mathbb{R}_+^n \end{aligned} \tag{10}$$

- Existence of solutions guaranteed by assumptions on v (continuity) and constraints on α (compactness)
- No uniqueness (a priori) since nonlinear dynamics can lead to multiple minima

Treats problem as a optimization task (not machine learning)
⇒ **Method: SLSQP (Sequential Least Squares Quadratic Programming)**

- **Key Considerations**
  - Gradient computations typically via finite difference methods
  - Nonlinear equality and inequality constraints via QP subproblems
  - Alternative methods are adjoint methods or derivative-free optimizers

[5] **nocedal_sequential_2006**.

Treats problem as a optimization task (not machine learning)

⇒ **Method: SLSQP (Sequential Least Squares Quadratic Programming)**

- Iterative method for constrained nonlinear optimization[5]
- **Key Considerations**
  - Gradient computations typically via finite difference methods
  - Nonlinear equality and inequality constraints via QP subproblems
  - Alternative methods are adjoint methods or derivative-free optimizers

[5] **nocedal_sequential_2006**.

Treats problem as a optimization task (not machine learning)

$\Rightarrow$ **Method: SLSQP (Sequential Least Squares Quadratic Programming)**

- Iterative method for constrained nonlinear optimization[5]
- **Key Considerations**
  - Gradient computations typically via finite difference methods
  - Nonlinear equality and inequality constraints via QP subproblems
  - Alternative methods are adjoint methods or derivative-free optimizers
- **Advantages**
  - Smooth constraint handling $\Rightarrow$ differentiability of ODE solver
  - Bound constraints on parameters $\Rightarrow$ physically meaningful solutions
  - Gradient-based optimization $\Rightarrow$ efficient convergence

[5] **nocedal_sequential_2006**.

Treats problem as a optimization task (not machine learning)
$\Rightarrow$ **Method: SLSQP (Sequential Least Squares Quadratic Programming)**

- Iterative method for constrained nonlinear optimization[5]
- **Key Considerations**
  - Gradient computations typically via finite difference methods
  - Nonlinear equality and inequality constraints via QP subproblems
  - Alternative methods are adjoint methods or derivative-free optimizers
- **Advantages**
  - Smooth constraint handling $\Rightarrow$ differentiability of ODE solver
  - Bound constraints on parameters $\Rightarrow$ physically meaningful solutions
  - Gradient-based optimization $\Rightarrow$ efficient convergence
- **Limitations**
  - Static $\Rightarrow$ cannot adapt dynamically to changing conditions without re-solving
  - No learning capacity $\Rightarrow$ lacks ability to predict future states
  - Computationally expensive for high-dimensional systems
    $\Rightarrow$ $\mathcal{O}(n^2)$ memory complexity (from Hessian approximations) and $\mathcal{O}(n^3)$ time complexity (from solving QP subproblems).

[5] **nocedal_sequential_2006**.

Treats problem as a machine learning task (not pure optimization)
$\Rightarrow$ **Method: "Physics-Aware" Neural Networks**

- **Key Considerations**
  - Physics principles via custom layers (parameter-dependent weights and biases)
  - Velocity acts as a physics-grounded activation function
  - Nonlinear constraints via additional constraint layers or penalty terms
  - Mimics Euler discretization step to achieve desired final state
  - Alternative methods are physics-informed NN

Treats problem as a machine learning task (not pure optimization)
⇒ **Method: "Physics-Aware" Neural Networks**

- Learning-based method for constrained nonlinear optimization
- **Key Considerations**
  - Physics principles via custom layers (parameter-dependent weights and biases)
  - Velocity acts as a physics-grounded activation function
  - Nonlinear constraints via additional constraint layers or penalty terms
  - Mimics Euler discretization step to achieve desired final state
  - Alternative methods are physics-informed NN

Treats problem as a machine learning task (not pure optimization)
⇒ **Method: "Physics-Aware" Neural Networks**

- Learning-based method for constrained nonlinear optimization
- **Key Considerations**
  - Physics principles via custom layers (parameter-dependent weights and biases)
  - Velocity acts as a physics-grounded activation function
  - Nonlinear constraints via additional constraint layers or penalty terms
  - Mimics Euler discretization step to achieve desired final state
  - Alternative methods are physics-informed NN
- **Advantages**
  - Dynamic Adaptation ⇒ learns from changing conditions
  - Physics-Grounded Learning ⇒ efficient generalization
  - Avoids explicit PDE constraints in loss function ⇒ contrast with PINN

Treats problem as a machine learning task (not pure optimization)
$\Rightarrow$ **Method: "Physics-Aware" Neural Networks**

- Learning-based method for constrained nonlinear optimization
- **Key Considerations**
  - Physics principles via custom layers (parameter-dependent weights and biases)
  - Velocity acts as a physics-grounded activation function
  - Nonlinear constraints via additional constraint layers or penalty terms
  - Mimics Euler discretization step to achieve desired final state
  - Alternative methods are physics-informed NN
- **Advantages**
  - Dynamic Adaptation $\Rightarrow$ learns from changing conditions
  - Physics-Grounded Learning $\Rightarrow$ efficient generalization
  - Avoids explicit PDE constraints in loss function $\Rightarrow$ contrast with PINN
- **Limitations**
  - Training Requirements $\Rightarrow$ needs large datasets or physics-based regularization
  - Interpretability Challenges $\Rightarrow$ difficult to verify constraint satisfaction
  - Computational Cost $\Rightarrow$ expensive training phase but fast inference
    $\Rightarrow$ $\mathcal{O}(n)$ memory complexity (from NN structure) and $\mathcal{O}(n^2)$ time complexity (from forward/backward passes).

Neural Network for Constrained Optimization

Fixed Step Learning Architecture

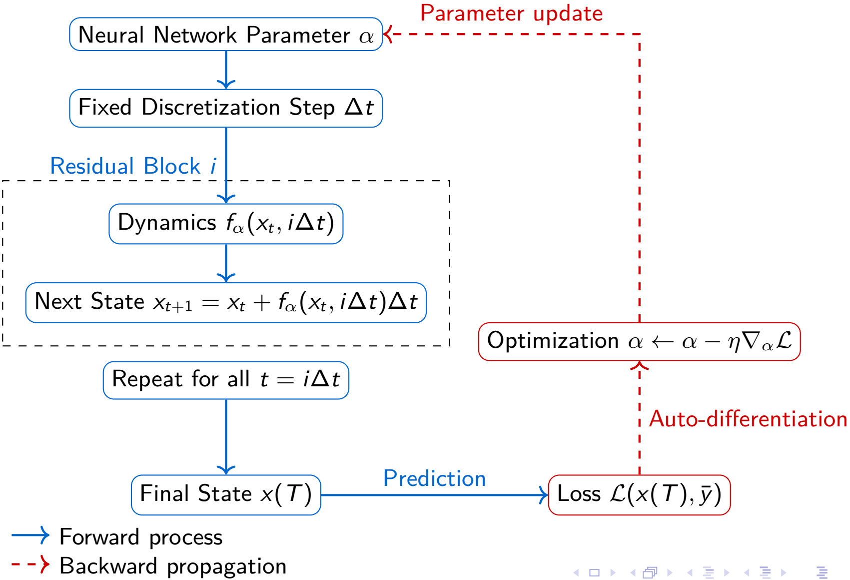

Table of Contents

1. Introduction
2. Existing Traffic Models
3. Optimization (Learning-Based) for Traffic Flow Reconstruction
4. Conclusion and Perspectives

Comparison of Approaches

- **FTL/LWR Models**:
  - Traditional, physics-based approaches for traffic modeling
  - Theoretical Focus ⇒ analytical insights but lack adaptability
- **Physics-Informed/Aware Methods**:
  - Combine data-driven learning with physics principles for optimization
  - Practical Focus ⇒ adaptive and scalable for real-time applications

Challenges in our approach

- Reconstructed discrete density depends on α, which may not be unique
  ⇒ complicates convergence analysis
  ⇒ convergence to a specific solution (e.g., LWR) challenging
- Additional constraints (physical or mathematical)
  ⇒ reduce ambiguity and guide optimization
- Explore solutions corresponding to multiple minima (even if not known LWR or FtL)
  ⇒ gain insights into system behavior

References I

Physics-Informed vs Physics-Aware Neural Networks

Physics-Informed Neural Networks (PINNs)

Key Characteristics

- Built around PDE
- Direct enforcement of physical laws through loss function
- Simultaneous optimization of data fitting and physics constraints

Best Applications

- Problems with known governing equations
- Limited training data scenarios

Physics-Aware Neural Networks

Key Characteristics

- Indirect incorporation of physical knowledge
- Uses physics-inspired architecture and regularization
- More flexible implementation

Best Applications

- Complex systems with uncertain physics
- Scenarios requiring computational efficiency

Convergence of the model

Convergence to LWR

Set

$$\mathcal{A}_N := \left\{ \alpha \in \mathbb{R}^n : \quad \alpha_i^N \in [1, \bar{z}_i], \quad i = 0, \ldots, n-1 \right\} \tag{11}$$

with

$$\bar{z}_i := \min \left\{ \frac{N(\bar{x}_{i+1} - \bar{x}_i)}{L}, \frac{N(\bar{y}_{i+1} - \bar{y}_i)}{L} \right\}. \tag{12}$$

Under some assumptions, the approximate density

$$\rho^N(t,x) = \sum_{i=0}^{n-1} \frac{\bar{\alpha}_i^N L}{N(x_{i+1}(t) - x_i(t))} \chi_{[x_i(t), x_{i+1}(t))}(x), \quad x \in \mathbf{R}, \quad t \in [0,T], \tag{13}$$

where $\bar{\alpha}_i^N$ is a solution to (10) converges to the **unique entropy** solution ρ of LWR

$$\begin{aligned} &\frac{\partial \rho}{\partial t}(t,x) + \frac{\partial f(\rho)}{\partial x}(t,x) = 0, \quad x \in \mathbf{R}, \quad t \in [0,T], \\ &\rho(0,x) = \bar{\rho}(x), \quad x \in \mathbf{R}. \end{aligned} \tag{14}$$

Typically, we impose a condition of the type

$$\max_{i=0,\ldots,n-1} \alpha_i^N = o(N). \tag{15}$$