Physics-Informed Learning and Control for Mixed Autonomy Systems: Application to Traffic Smoothing

ASU Interdisciplinary Transportation Seminar

PRESENTER
Zhe Fu
Ph.D. candidate at UC Berkeley
BAIR | ITS | CITRIS

Intelligent Mobility Systems – Mixed Autonomy

Complexity across autonomy level, human behavior, and human-machine interaction

We live around mixed-autonomy systems, where automated and human agents coexist

Automated & Human-driven Vehicles

Autonomous & Piloted Drones

Unpiloted & Piloted Air Taxis

Mixed autonomy systems operation is inherently **complex** on multiple layers

Autonomy Layer

- Limited scale
- Partially autonomous
- Different sensing & actuation

Human Behavior Layer

- Heterogeneous
- Unpredictable
- Varies responses to automation

Interaction & Scale Layer

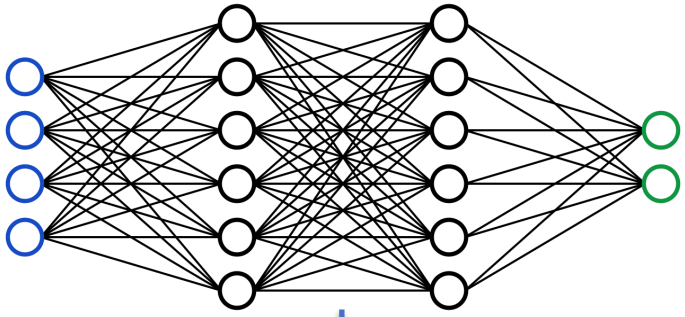
- Multi-agent
- Heterogeneities and behavioral couplings

Addressing these challenges requires an approach that is both **theoretically grounded** and **systemically integrated**.

Theoretically Grounded – Model-Informed Learning

Embedding physical and structural models into learning to build interpretability and data efficiency

Pure Data-Driven Learning



Limitations:

- High data & compute cost
- Limited interpretability & physical consistency
- Poor generalization to new/noisy environments

Domain Knowledge

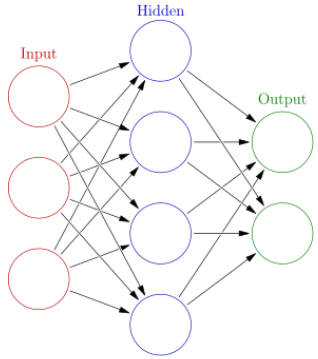
- Governing laws
- Conservation equations
- Structural priors
- Analytical models

Model-Informed Learning

$$\partial_t \rho + \partial_x f(\rho) = 0$$

$$f: \rho \mapsto \rho v(\rho) = v_{max}\rho\left(1 - \frac{\rho}{\rho_{max}}\right)$$

+



Goal: data-efficient, interpretable, and generalizable

Systemically Integrated – Co-Design and Embedded Intelligence

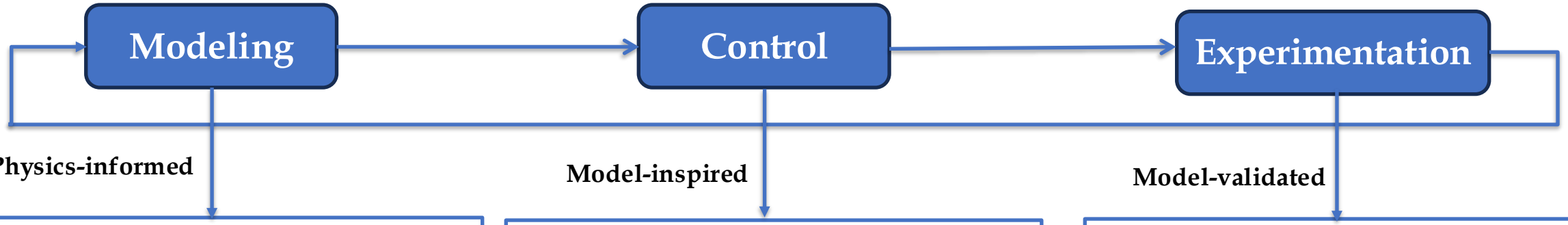
Modeling, Control and Experimentation as a united, closed loop and evolve together

Co-Design

Evolve in the unified pipeline within the physical systems

My Work: Co-Design for Safe and Efficient Mixed Autonomy

My work grounded with the philosophy of model-informed co-design of embedded intelligence



Modeling: Capture systems dynamics

Numerical Analysis and Machine Learning

Energy Model: [CAKFBLSB] **under review**

Neural Finite Volume Method:
[F*†BB*C*D*DL*LN*P] SIAM Journal on Scientific Computing (under review)
[F*L*C*N*DB] ICLR
[F*C*L*N*LDB] NeuS

Control: Design algorithms steer systems

Control Theory and Deep Learning

Kernel-based + Imitation Learning Control:
[F†AKB] TR-C;
[KFB] **IEEE ITSC;**
[F†KWLDB] IEEE IV
Learning-based State Estimation:
[MFNFLB] **CDC**
Neural Control Barrier Function:
[CF†B] **ACC**
Unsupervised Anomaly Detection:
[LF†B] **IEEE IV;**
[LF†B] **TRB**
Smart Charging Optimization:
[WFAL] **Energies**

Experimentation: Evaluation and validation

Experimentation and Data Analysis

CIRCLES (100 AVs Field Test):
[WFLNAUHKCBRWPSSBD] **IEEE CSM;**
[…F…] IEEE CSM;
[…F…] **Nature (under review);**

Bay Area Stop-and-go Trajectories:
https://nicolemzh.github.io/highways.dataset/downloads

Smart Charging Pilot w/ BMW:
[SFAL] **TR-D**

*** Equal first author; † Corresponding author**

Highlight: System Efficiency through a Few Automated Agents

Use a few partially automated cars to mitigate stop-and-go waves for the whole system

Problem: Stop-and-go waves

Stop-and-go waves emerge without bottlenecks – purely from human reaction and delay

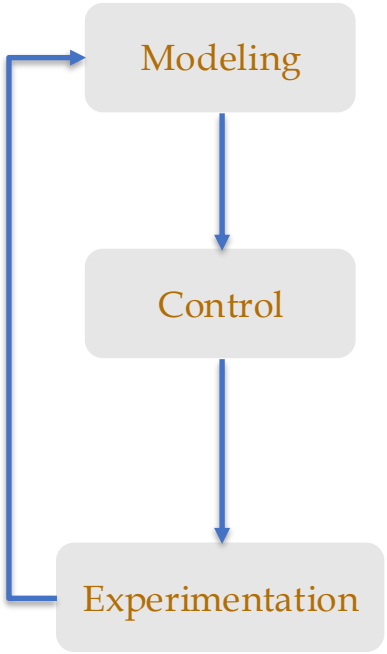
Research Question

Goal: Dissipate stop-and-go waves

Methods: Few partially automated vehicles (AVs)

Constraints: Maintain overall throughput

Our Solution: Few AVs as "Robot Managers" to "Shepherd" Human



1. Traffic State Estimation

Sensing downstream conditions

2. Target Speed Design

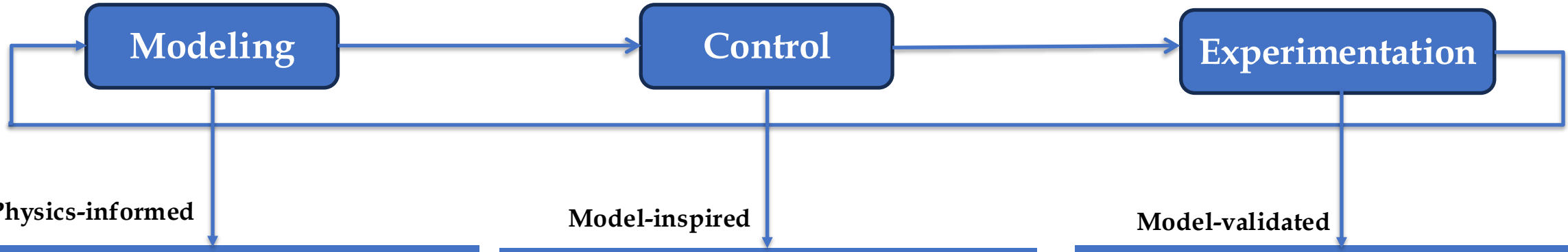
Planning a smooth speed profile

3. AV as Lead Control

The AV executes the plan, "shepherding" human drivers behind it.

This Project Spanned the Full Modeling – Control – Experimentation Loop

Physics-informed modeling; Model-inspired control; Field experiment validation



Neural Finite Volume Methods (NFVM)	Kernel-based + Imitation Learning Control	CIRCLES (100 AVs Field Test)
[F*†BB*C*D*DL*LN*P] **SIAM Journal on Numerical Analysis (under review)** [F*L*C*N*DB] **ICLR** [F*C*L*N*LDB] **NeuS**	[F†AKB] **TR-C;** [KFB] **IEEE ITSC;** [F†KWLDB] **IEEE IV**	[WFLNAUHKCBRWPSSBD] **IEEE CSM;** […F…] **IEEE CSM;** […F…] **Nature (under review);**

Source of Hardness	**Discontinuities in traffic dynamics; Limited data**
Solution Method	A new **physics-informed** neural solver -- NFVM

Source of Hardness	**Partial observability; limited actuation of AVs**
Solution Method	**Kernel-based** planner; Imitation Learning control

Source of Hardness	**Communication delay; sim-to-real gap**
Solution Method	**Model-validated** & feedback loop deployment

*** Equal first author; † Corresponding author**

A Physics-Informed Neural Solver for Hyperbolic PDEs

Accurate, Robust, and Computationally Efficient tool for Traffic Estimation & Prediction

Modeling

Physics-informed

Neural Finite Volume Methods (NFVM)

[F*†BB*C*D*DL*LN*P] **SIAM Journal on Scientific Computing (under review)**
[F*L*C*N*DB] **ICLR**
[F*C*L*N*LDB] **NeuS**

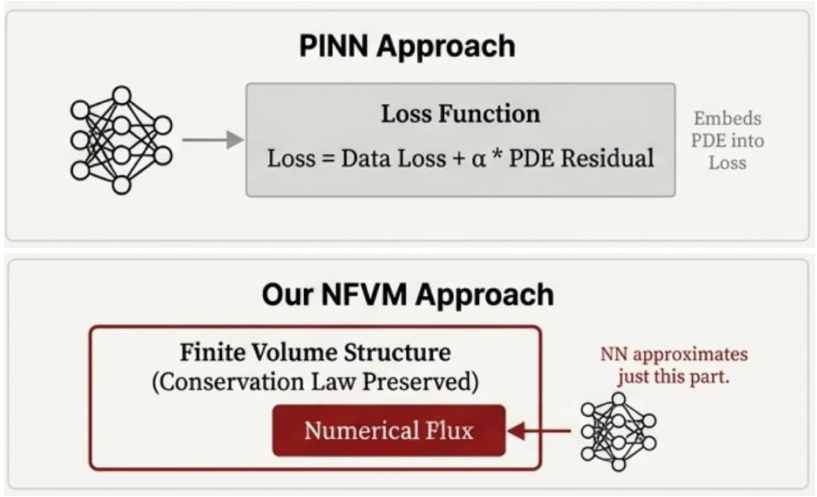
Source of Hardness	**Discontinuities in traffic dynamics; Limited data**
Solution Method	A new **physics-informed** neural solver -- NFVM

*** Equal first author; † Corresponding author**

A Physics-Informed Neural Solver for Hyperbolic PDEs

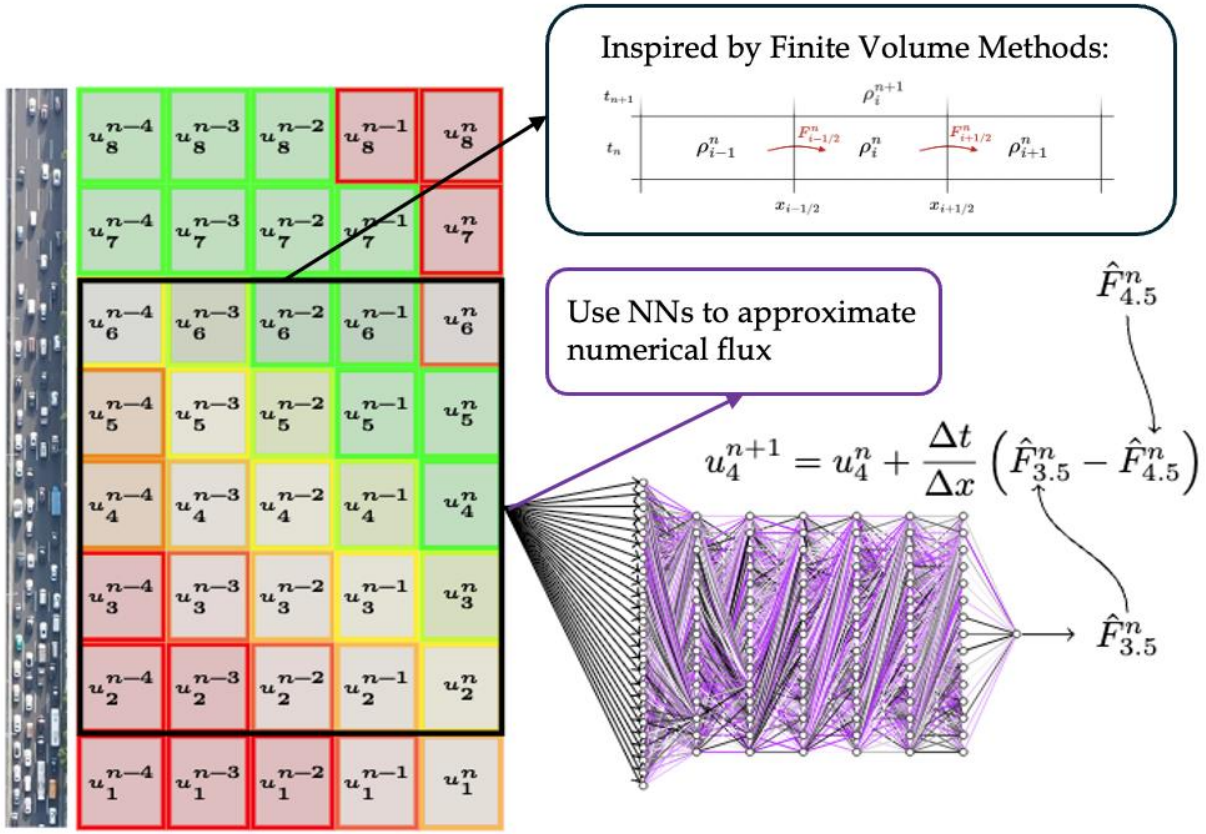
Accurate, Robust, and Computationally Efficient tool for Traffic Estimation & Prediction

Previous Physics-informed Machine Learning approach:
add PDE into loss



Key Contributions of Our NFVM:

- **Accuracy:** Outperforms all known FV numerical schemes across multiple hyperbolic PDE models;
- **Robustness:** Outperforms numerical baselines on multiple noisy traffic field data;
- **Computationally Efficient:** **Matches** the computational cost of first-order FV schemes;



An accurate and robust tool
for Traffic State Estimation & Prediction,
supports simulation and control design.

Traffic Flow Dynamics

1950s — **Lighthill-Whitham-Richards (LWR) Model**

- Conservation Equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0$$

- Fundamental Diagram (for flow q as a function of density ρ):

$$q = f(\rho) = \rho \cdot v(\rho)$$

1970s — **Payne-Whitham (PW) Model**

- Conservation of Mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} = 0$$

- Momentum Equation (for velocity v):

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{V(\rho) - v}{\tau} - \frac{1}{\rho}\frac{\partial P(\rho)}{\partial x}$$

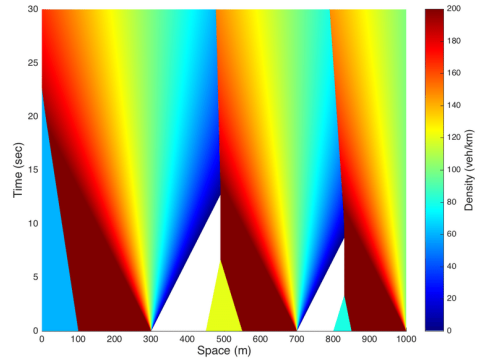
2000s — **Aw-Rascle-Zhang (ARZ) Model**

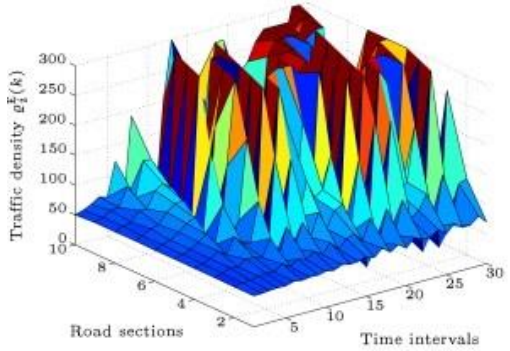
- Conservation of Mass:

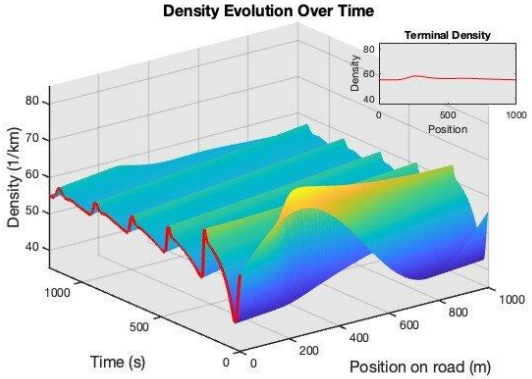
$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} = 0$$

- Relative Momentum Equation (for relative velocity w):

$$\frac{\partial w}{\partial t} + v\frac{\partial w}{\partial x} = 0$$



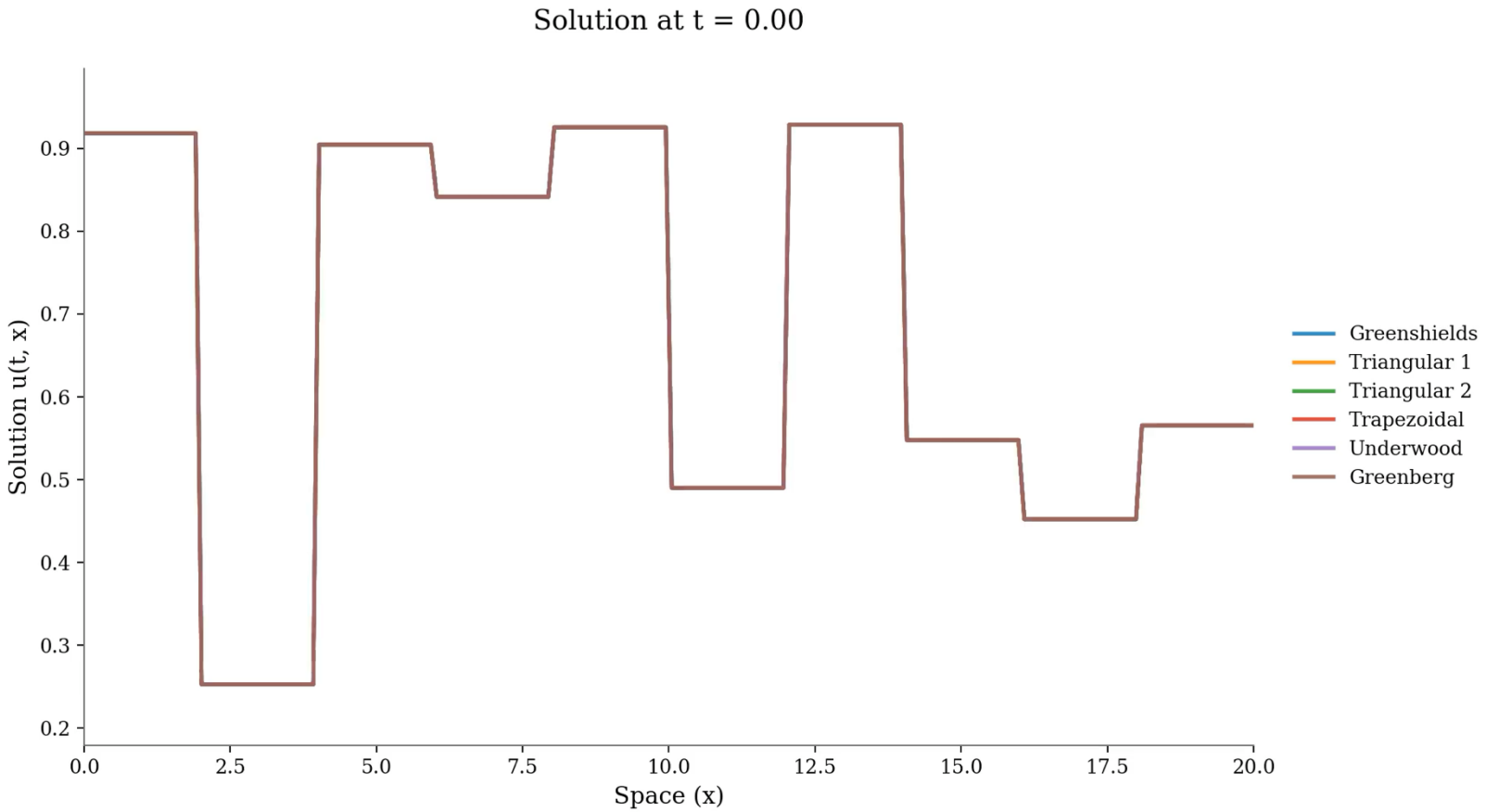




→ *All **Hyperbolic PDEs***

Why Are Hyperbolic PDEs Hard to Solve?

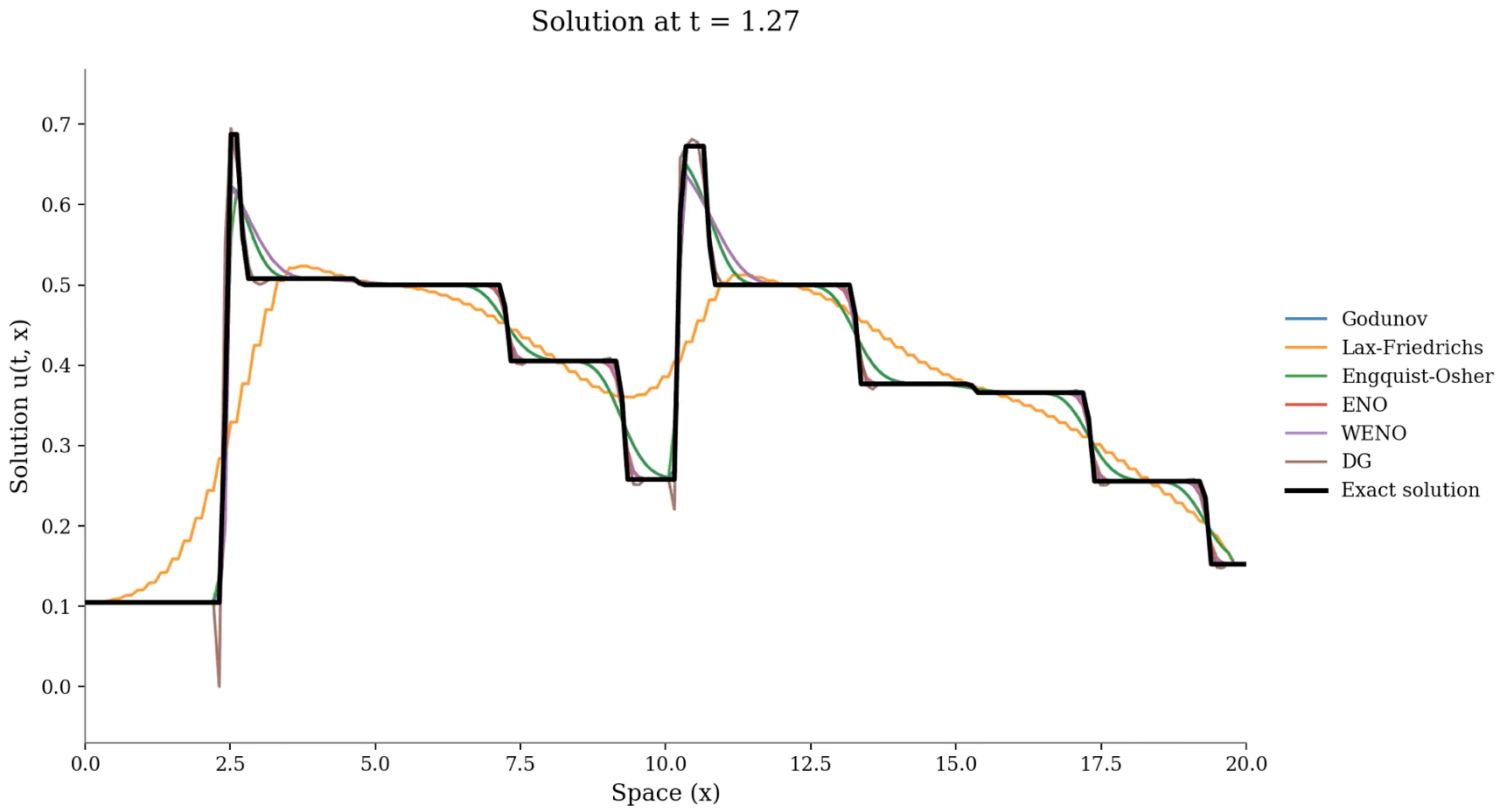
Irregular solutions and ambiguity in weak formulations make them challenging



- Irregular solutions (shocks & discontinuities)
- Infinitely many weak solutions – only the entropy solution is physically valid

Traditional Solvers: Can be Accurate but Expensive

Finite Volume Methods converge to entropy solutions – at high computational cost



Traditional solvers like Finite Volume Methods tend to over smooth shocks;

They achieve high accuracy requiring extremely fine meshes.

Can we design NN-based solvers that match or even surpass FVMs in accuracy, while keeping the cost low?

Structural-Based NN Solvers: NFVM

NFVM bridges the gap between deep learning and physically consistent PDE solutions

Limitations of Existing NN Solvers:

- Typical NN methods struggle with preserving:
  - Regularity
  - Entropy conditions
  - Conservation laws

Neural Finite Volume Method (NFVM):

- Preserves essential PDE properties:
  - Stability and conservation laws inherently satisfied
  - Entropy conditions to ensure physical relevance
- Efficient → Matches the computational cost of finite volume schemes
- Flexible architecture → Supports supervised and unsupervised learning
- Generalizable → Extensible to higher-order space/time, field data

Finite Volume Method for Hyperbolic Conservation Law

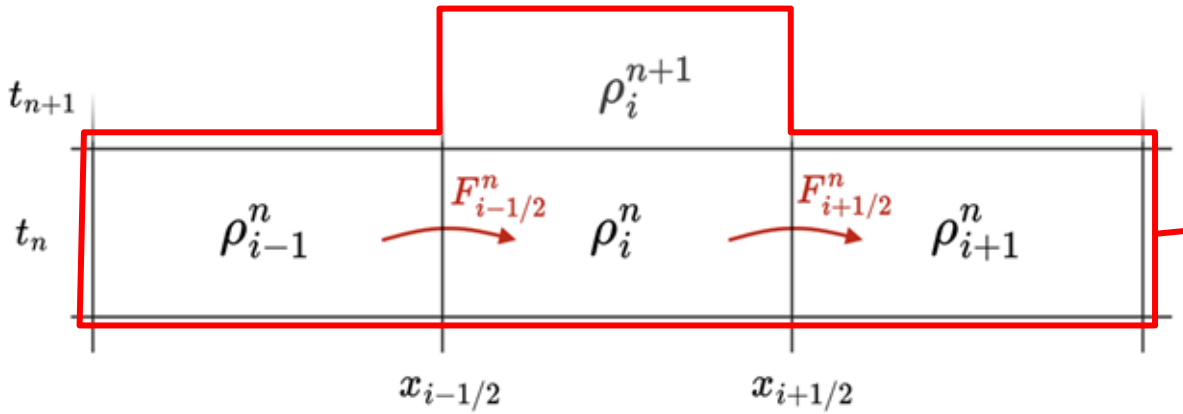
The structured foundation of our Neural Finite Volume Method (NFVM)

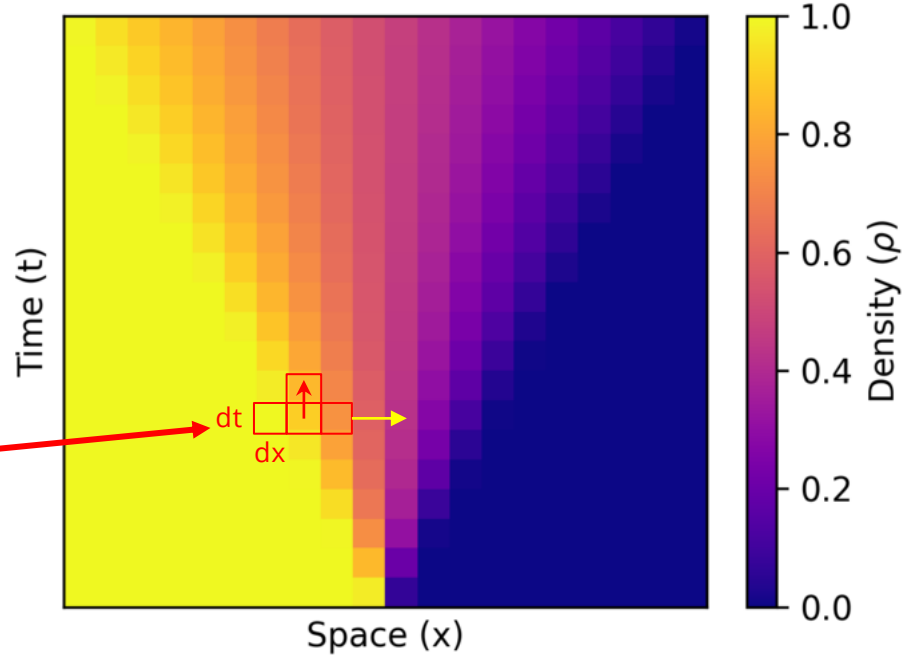
- Discretize space and time into cells

$$\rho_i^n = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \rho(x, t_n)dx \qquad F_{i+\frac{1}{2}}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(\rho(x_{i+1/2}, t))dt$$

- Update densities using numerical flux **F**

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x}(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n)$$



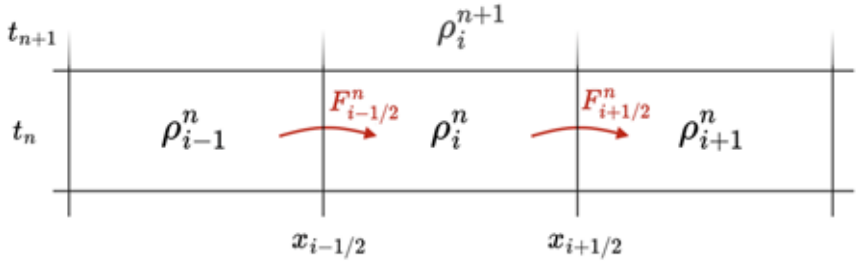


From Numerical Flux to Neural Flux

Replace hand-crafted flux approximations with learned neural networks

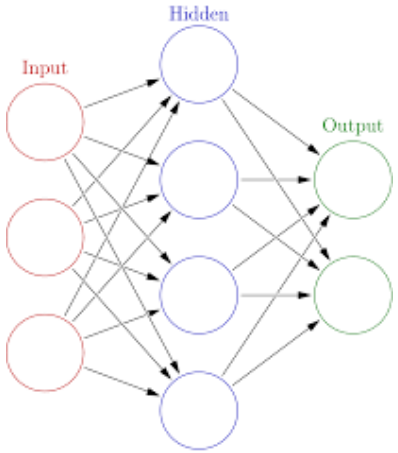
- Hand-crafted numerical flux (illustrated with Godunov)

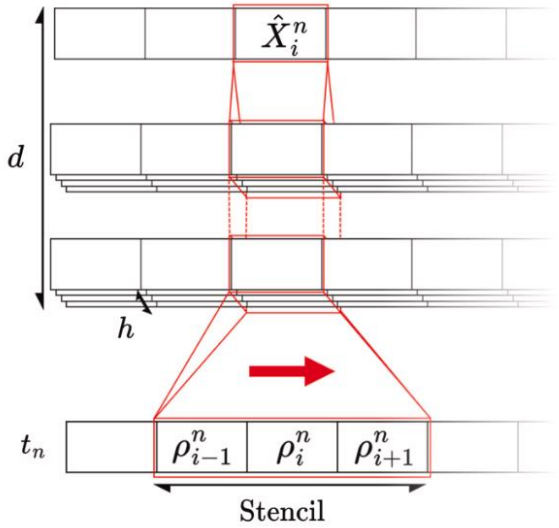
$$\mathcal{F}_G(\rho_1,\rho_2)=\begin{cases} f(\rho_2) & if\ \rho_c<\rho_2<\rho_1 \\ f(\rho_c) & if\ \rho_2<\rho_c<\rho_1 \\ f(\rho_1) & if\ \rho_2<\rho_1<\rho_c \\ \min\big(f(\rho_1),f(\rho_2)\big) & if\ \rho_1\le\rho_2 \end{cases}$$



- **Neural Flux used in NFVM**

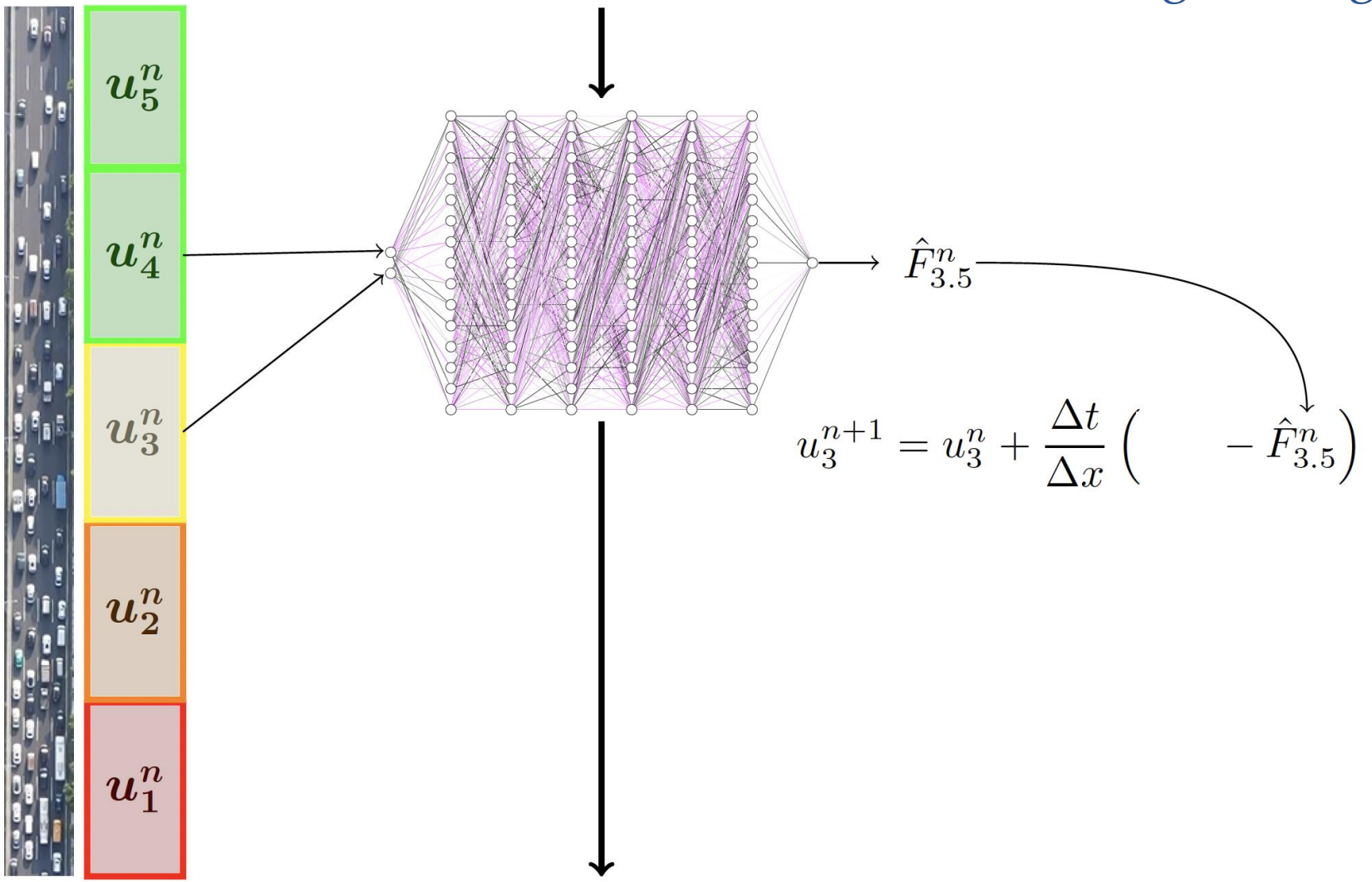
$$\mathcal{F}_{NN}(\rho_1,\rho_2)$$





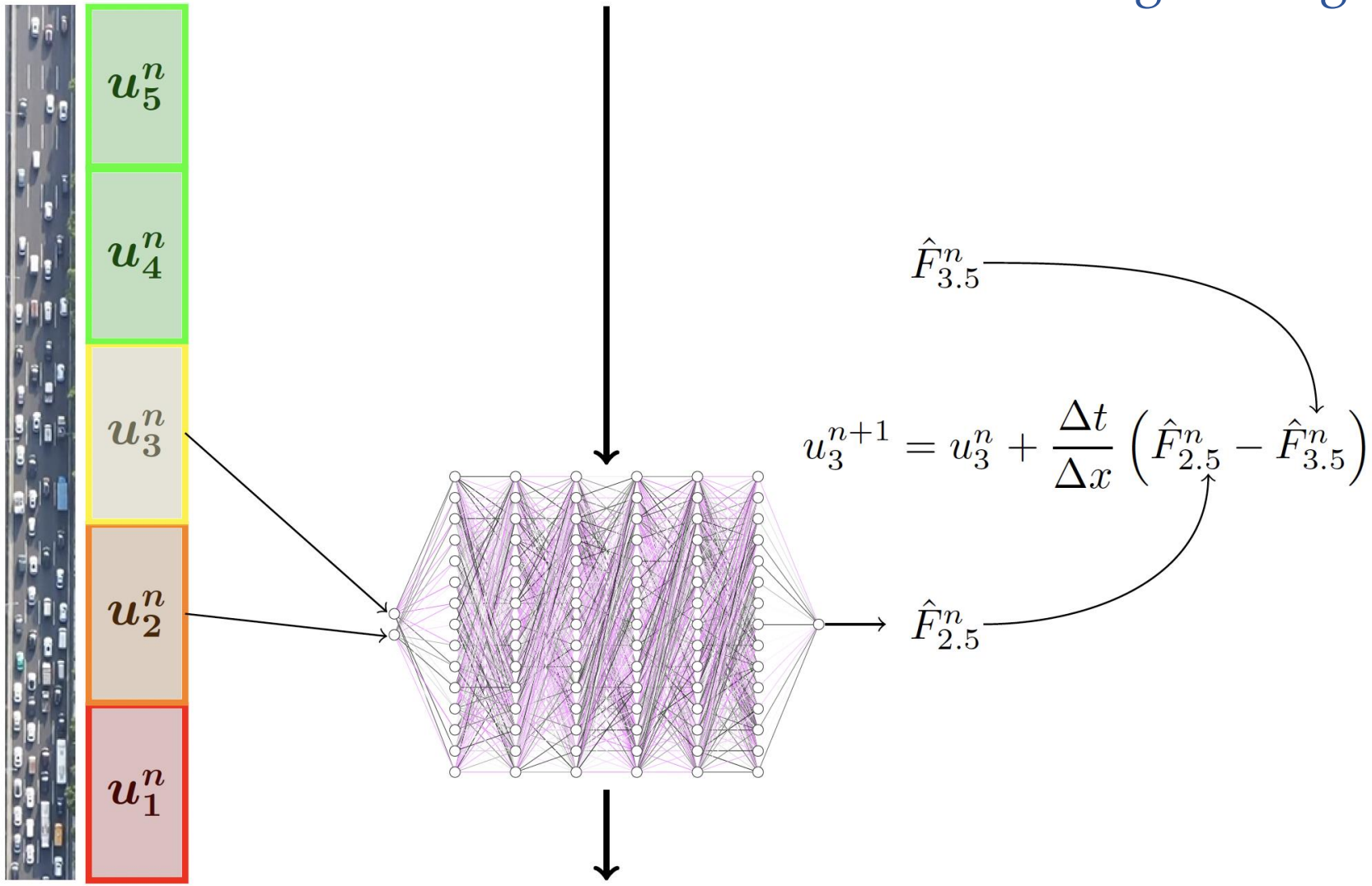
Space-wise Neural Approximation of Flux

A local neural network learns the numerical flux from neighboring cell values



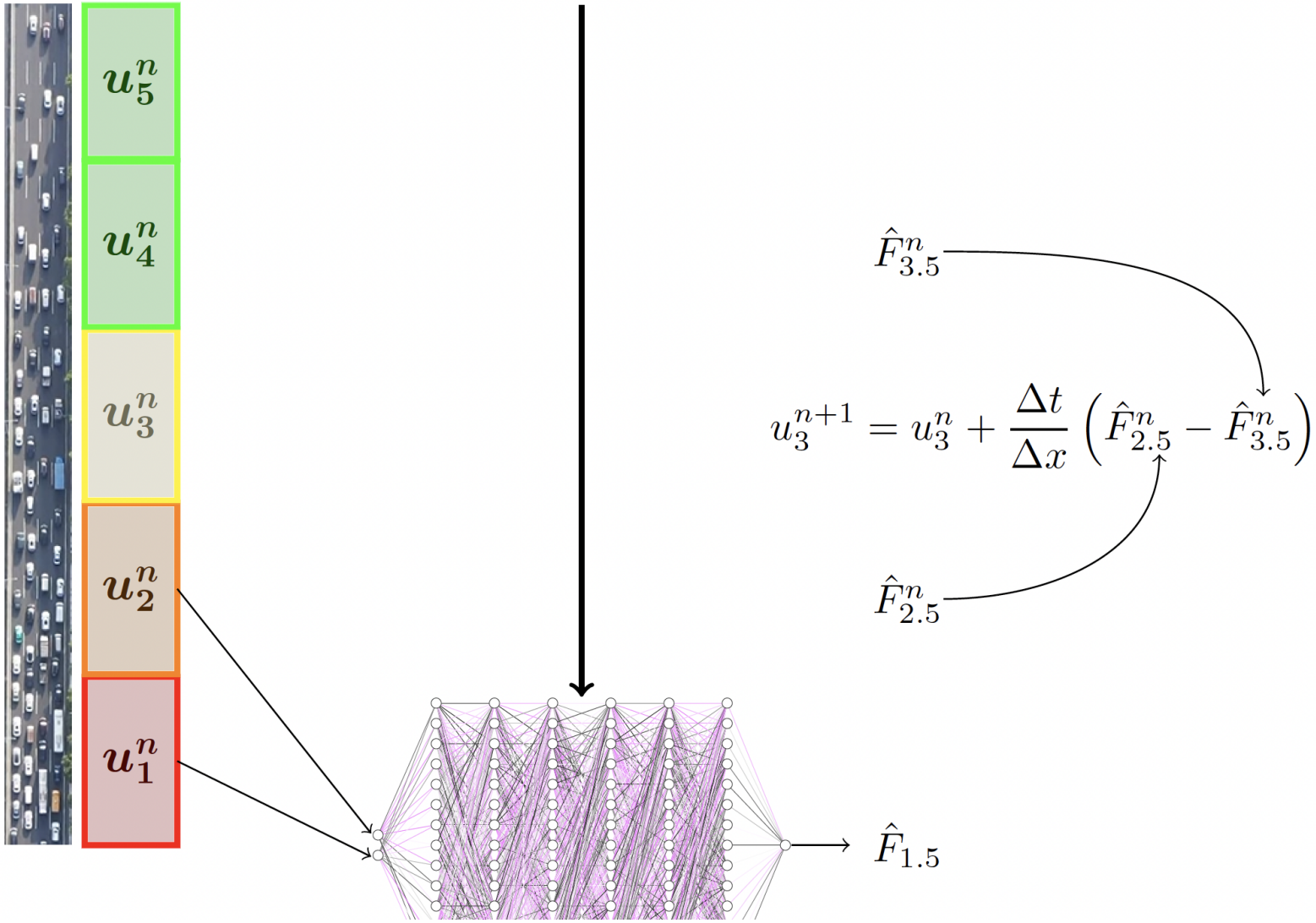
Space-wise Neural Approximation of Flux

A local neural network learns the numerical flux from neighboring cell values



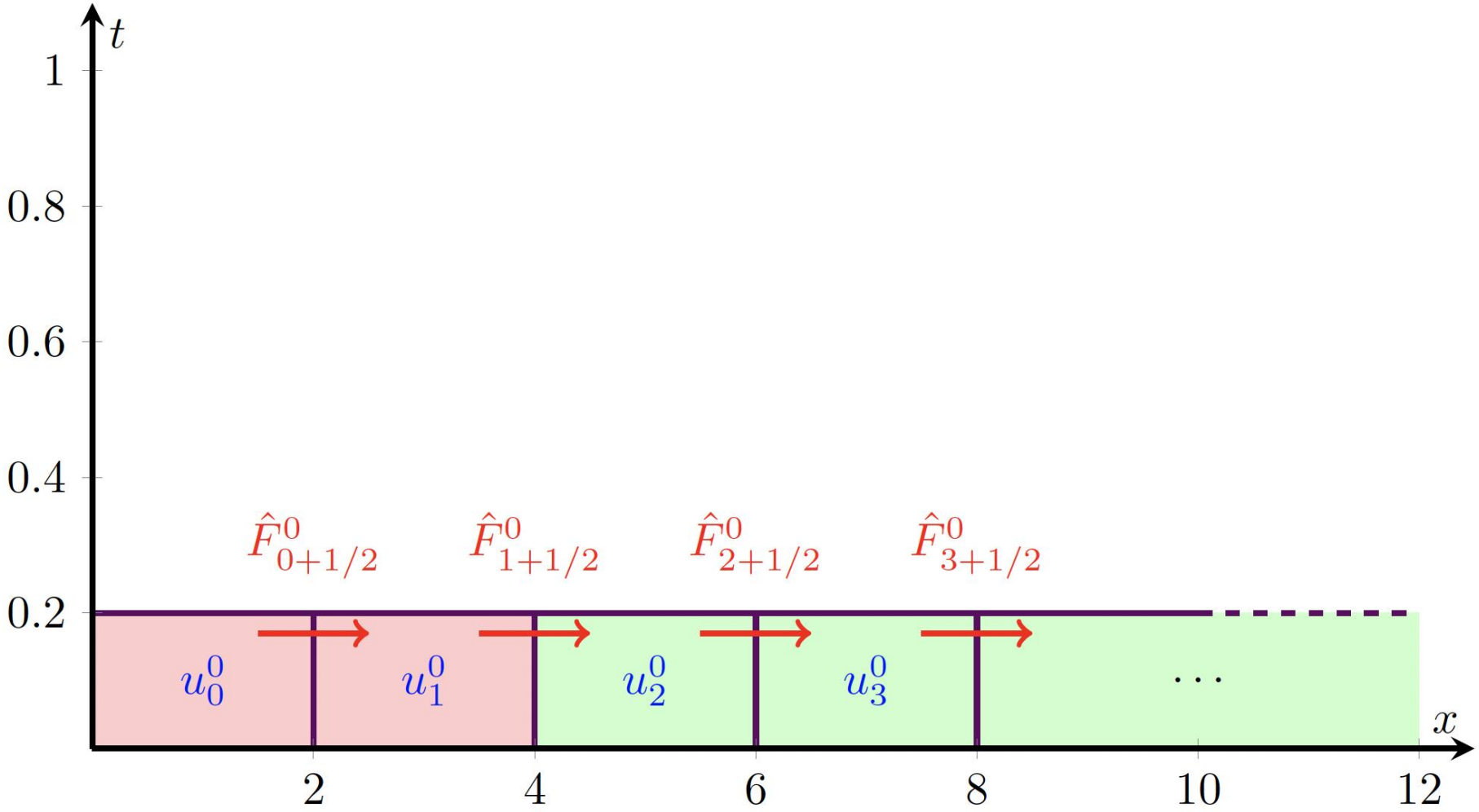
Space-wise Neural Approximation of Flux

A local neural network learns the numerical flux from neighboring cell values



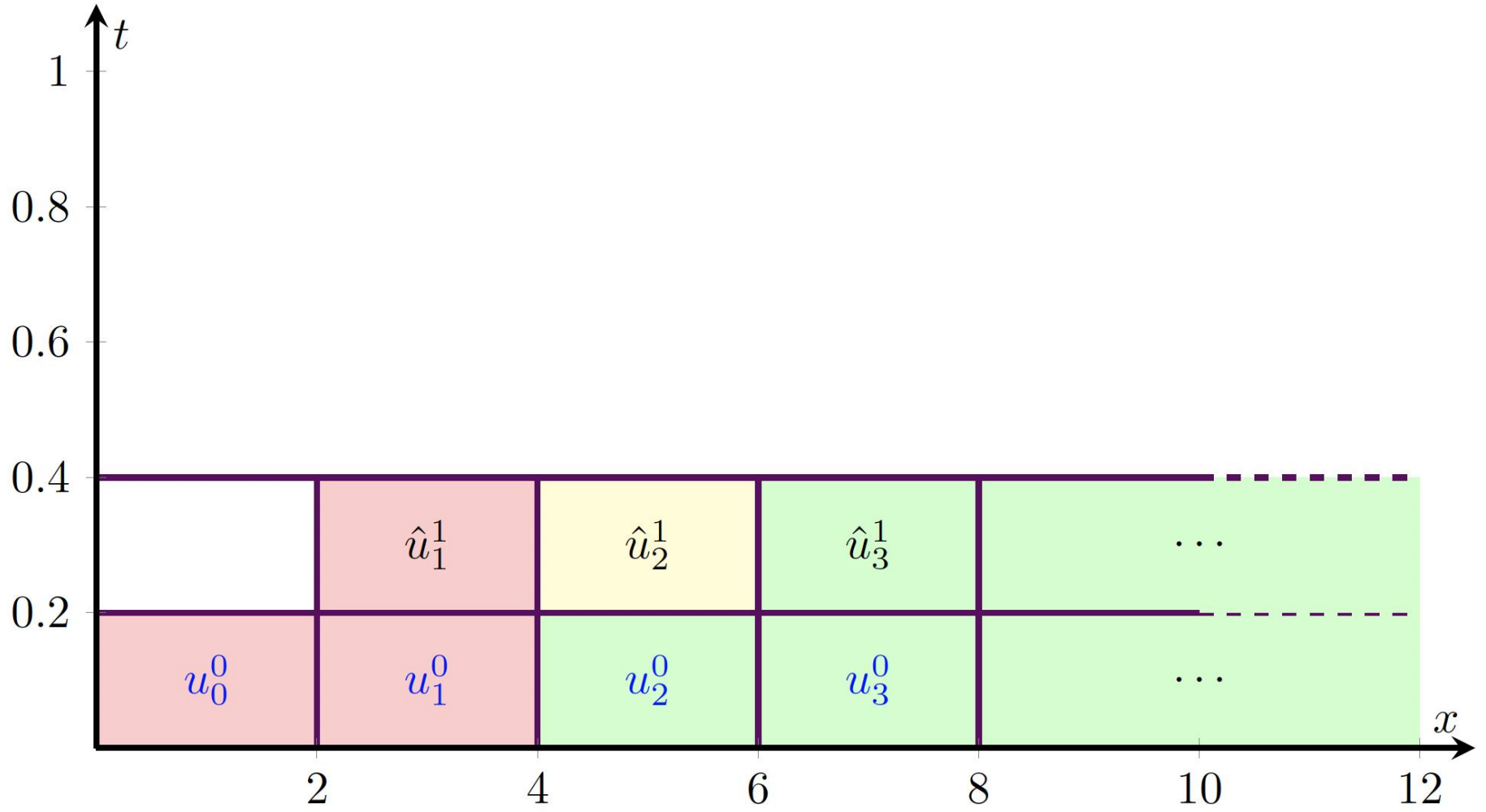
Time-wise Update Follows Finite Volume Method

Densities are advanced in time using learned fluxes – preserving conservation structure



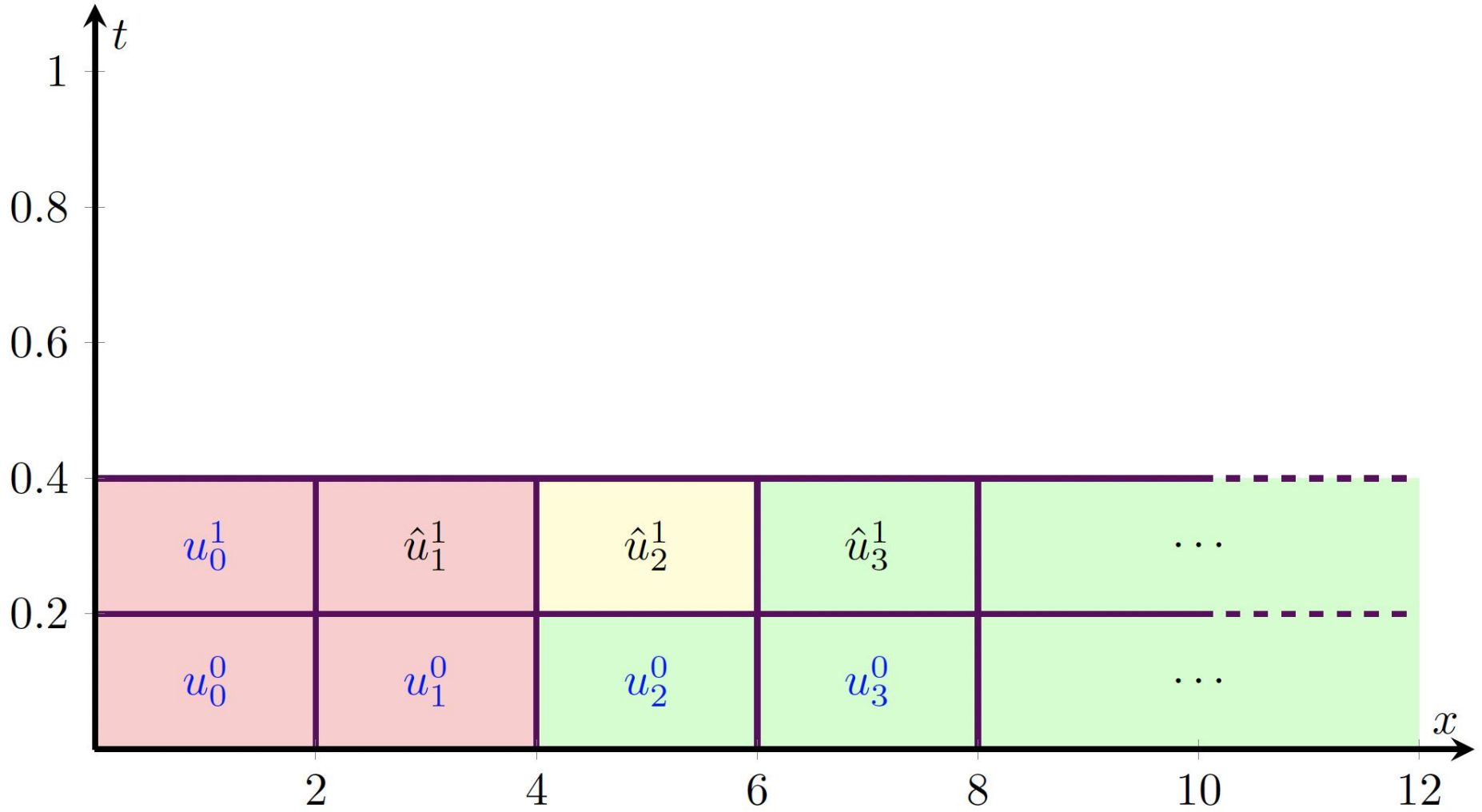
Time-wise Update Follows Finite Volume Method

Densities are advanced in time using learned fluxes – preserving conservation structure



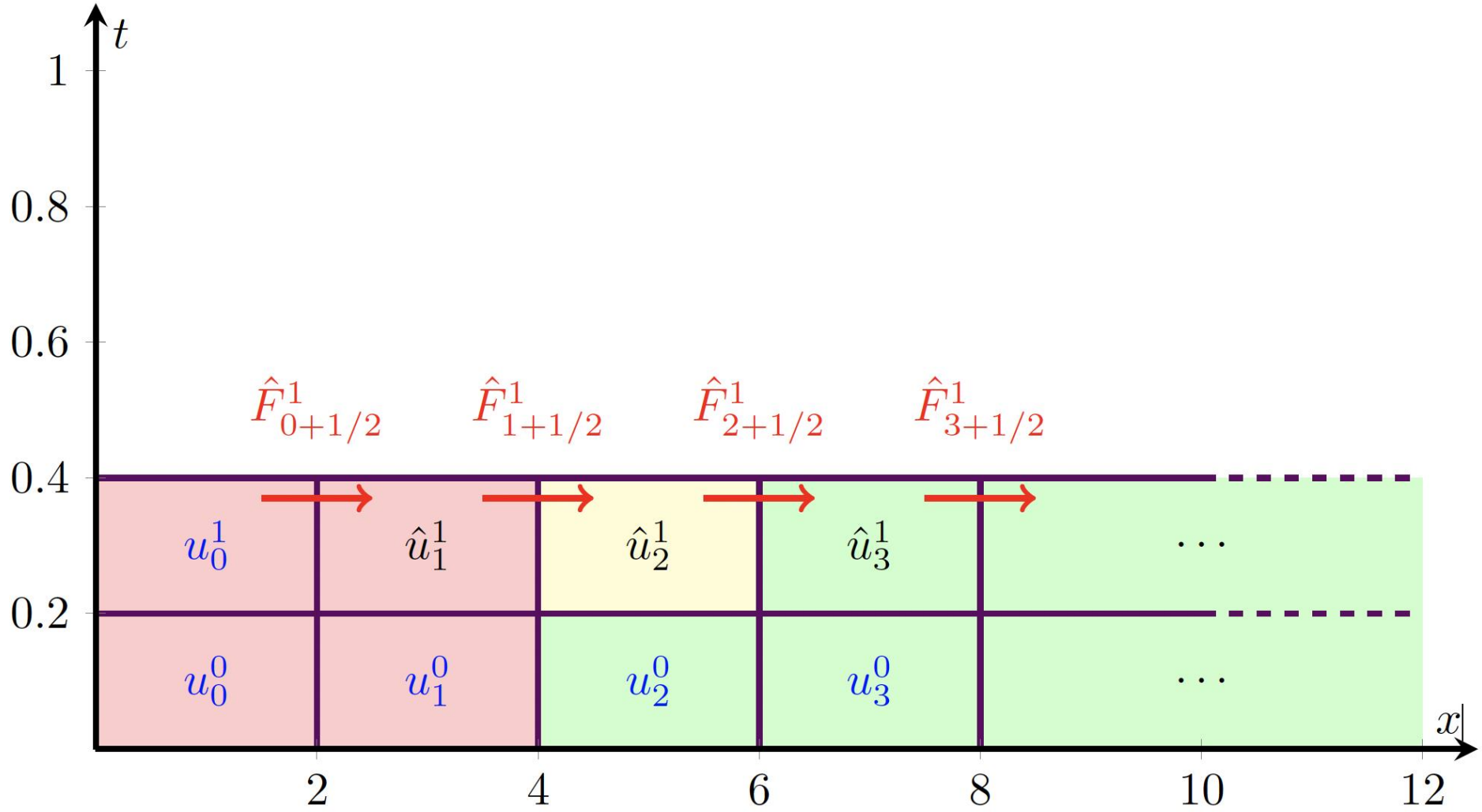
Time-wise Update Follows Finite Volume Method

Densities are advanced in time using learned fluxes – preserving conservation structure



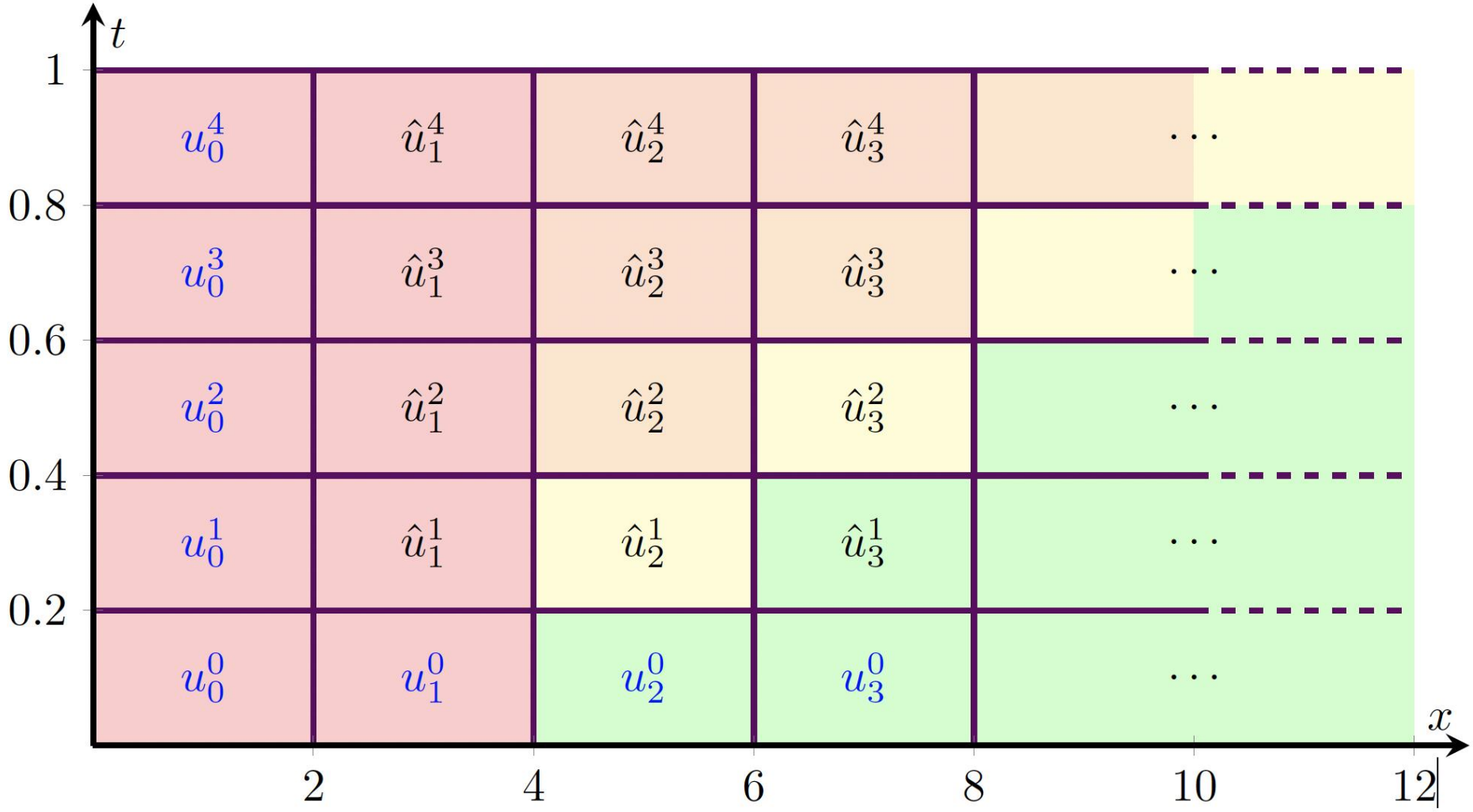
Time-wise Update Follows Finite Volume Method

Densities are advanced in time using learned fluxes – preserving conservation structure



Time-wise Update Follows Finite Volume Method

Densities are advanced in time using learned fluxes – preserving conservation structure



Our NN Architecture: A Compact 1D CNN

Small, efficient, and designed to fit into the FVM update loop

Our method: use a NN to better approximate the numerical flux

Architecture: 1D CNN

- Kernel size: # of stencils used (2 for 1st order) for the first layer
- Layers: 6
- Hidden layer size: 15
- Learned parameters: ~1200

The prediction updates $\hat{u}^{n+1}$ given $\hat{u}^n$ is as follows: $\forall j, m \in [\![1, 15]\!] \times [\![2, 5]\!]$,

$$\begin{cases} \boldsymbol{y}_{(\mathbf{1},\boldsymbol{j})} = \tanh(b_{(1,j)} + \boldsymbol{\omega}_{(\mathbf{1},\boldsymbol{j},\mathbf{1})} \star \hat{\boldsymbol{u}}^{\boldsymbol{n}}) \\ \boldsymbol{y}_{(\boldsymbol{m},\boldsymbol{j})} = \tanh(b_{m,j} + \sum_{k=1}^{15} \omega_{(m,j,k)} \boldsymbol{y}_{(\boldsymbol{m}-\mathbf{1},\boldsymbol{k})}) \\ \hat{\boldsymbol{F}}^{\boldsymbol{n}}_{\cdot+\frac{\mathbf{1}}{\mathbf{2}}} = b_{(6,1)} + \sum_{k=1}^{15} \omega_{(6,1)} \boldsymbol{y}_{(\mathbf{5},\boldsymbol{k})} \\ \hat{\boldsymbol{u}}^{\boldsymbol{n}+\mathbf{1}} = \hat{\boldsymbol{u}}^{\boldsymbol{n}} + \frac{\Delta t}{\Delta x}(\hat{\boldsymbol{F}}^{\boldsymbol{n}}_{\cdot-\frac{\mathbf{1}}{\mathbf{2}}} - \hat{\boldsymbol{F}}^{\boldsymbol{n}}_{\cdot+\frac{\mathbf{1}}{\mathbf{2}}}) \end{cases}$$

where $\boldsymbol{\omega}_{(\mathbf{1},\boldsymbol{j},\mathbf{1})} \in \mathbb{R}^2$, $\omega_{(6,1,k)} \in \mathbb{R}$, and $b_{(1,j)}, b_{(m,j)}, b_{(6,1)} \in \mathbb{R}$ are the trained neural network parameters.

Supervised NFVM: Learning from Riemann Problems

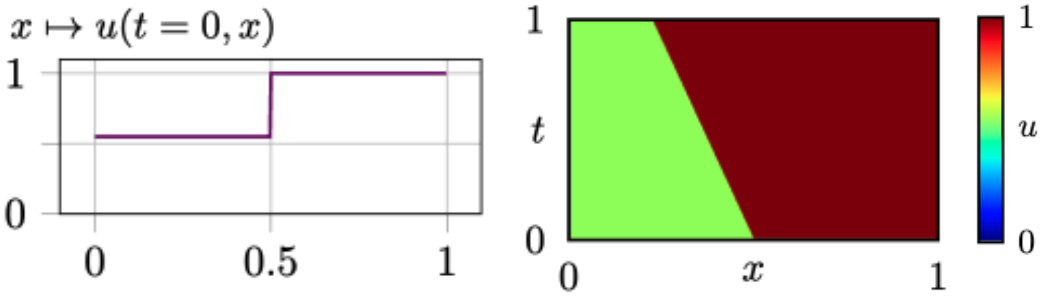
Trained on exact solutions of simple discontinuous setups to capture entropy behavior

Loss Function:

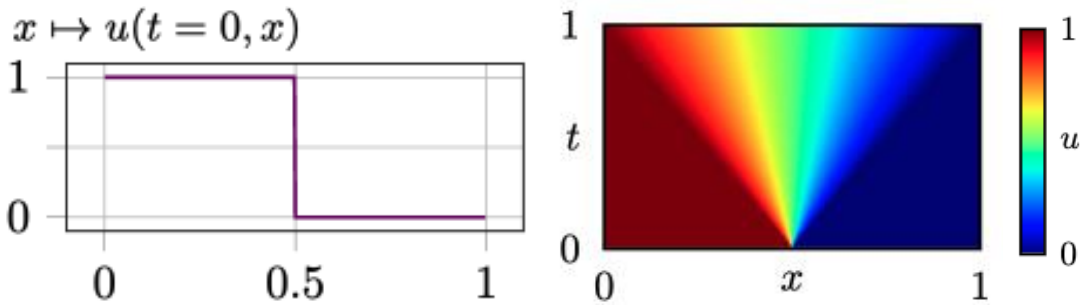
$$\mathcal{L} = \mathbb{E}_{u\sim\mathcal{D}}\|u - \hat{u}\|_2^2$$

Initial-value problem with two constant states separated by discontinuity.
(Simple cases, have explicit solutions, either shockwave or rarefaction wave)

- $\mathcal{D}$ is a dataset of solutions to Riemann problems
- u is a solution from $\mathcal{D}$.
- $\hat{u}$ is a solution computed using the NN on the same initial conditions as u.



Shockwave



Rarefaction wave

Unsupervised NFVM (UNFVM): Physics-Driven Learning

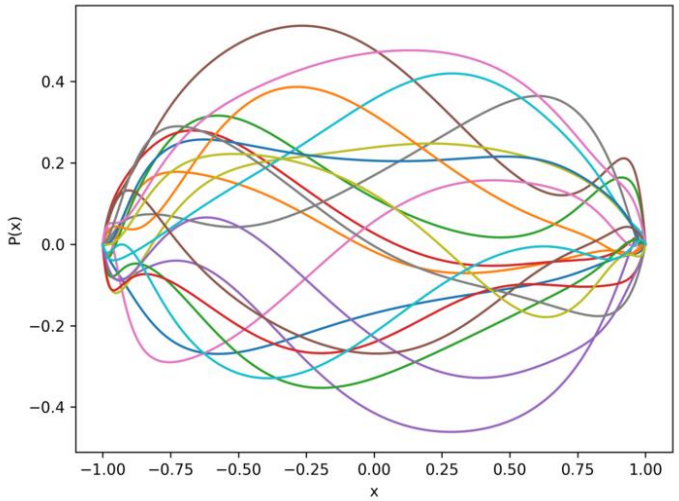
Learns from weak formulation loss over Riemann initial conditions

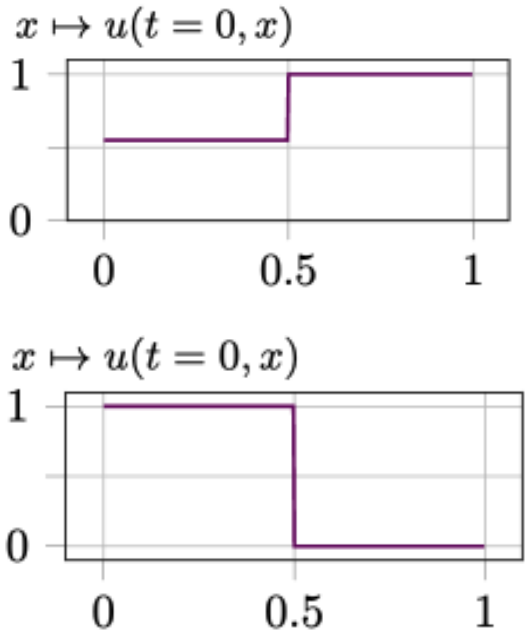
Loss Function:

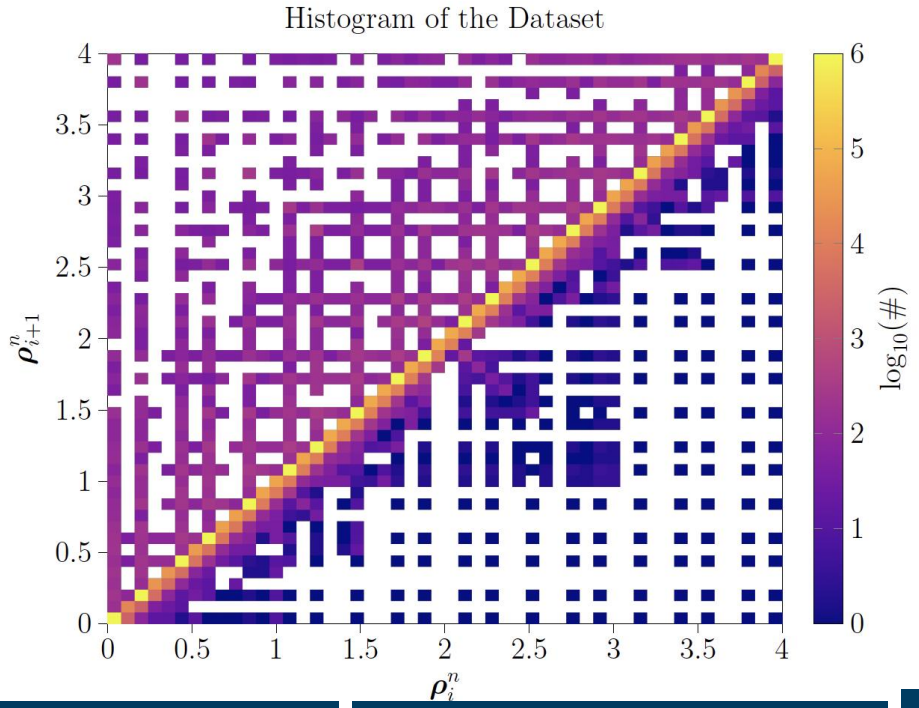
$$\mathcal{L} = \mathbb{E}_{\varphi\in\Phi,u_0\sim\mathcal{R}} \int_T \left(\int_X \partial_t \hat{u}(t,\cdot)\varphi(\cdot) + \int_X f(\hat{u}(t,\cdot))\varphi'(\cdot) \right)^2 dt$$

where Φ is a set of suitable test functions, $\mathcal{R}$ is a distribution of random Riemann initial conditions and X and T are the space and time domains.

Could be a set of compactly supported polynomials







Benchmark Problems: Scalar Hyperbolic PDEs

Evaluate NFVM on LWR traffic flow model and inviscid Burger's equation

Lighthill-Whitham-Richards (LWR):

In the context of traffic, the LWR model is the reduced form of Eq. (P) to:

$$\partial_t \rho + \partial_x f(\rho) = 0,$$

where:

- $\rho \in L^1([0,T] \times \mathbb{R})$ is the **conserved quantity** and represents **traffic density**,
- $f: \rho \longmapsto \rho v(\rho)$ is known as **traffic flux**.

Inviscid Burgers' Equation:

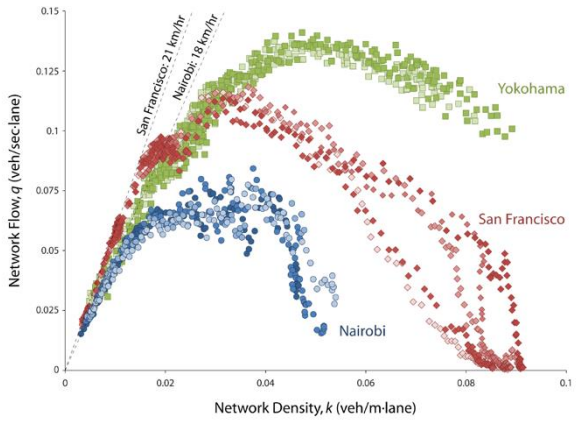
The Burgers' equation is a PDE used in various domains such as fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flow. It is a conservation law:

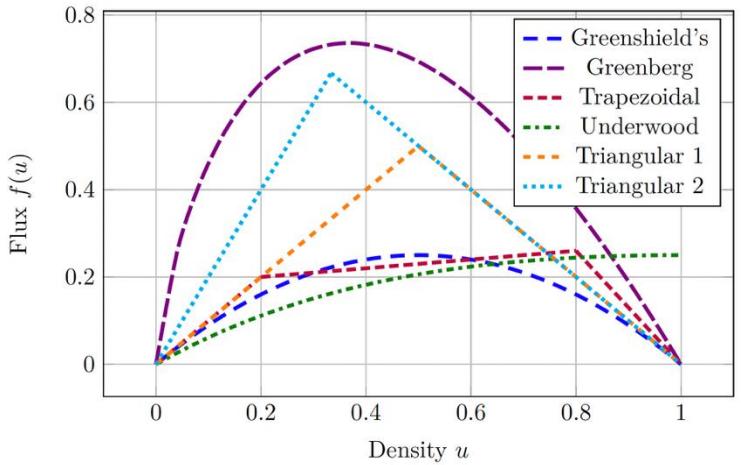
$$\partial_t u + \frac{1}{2}\partial_x u^2 = 0.$$

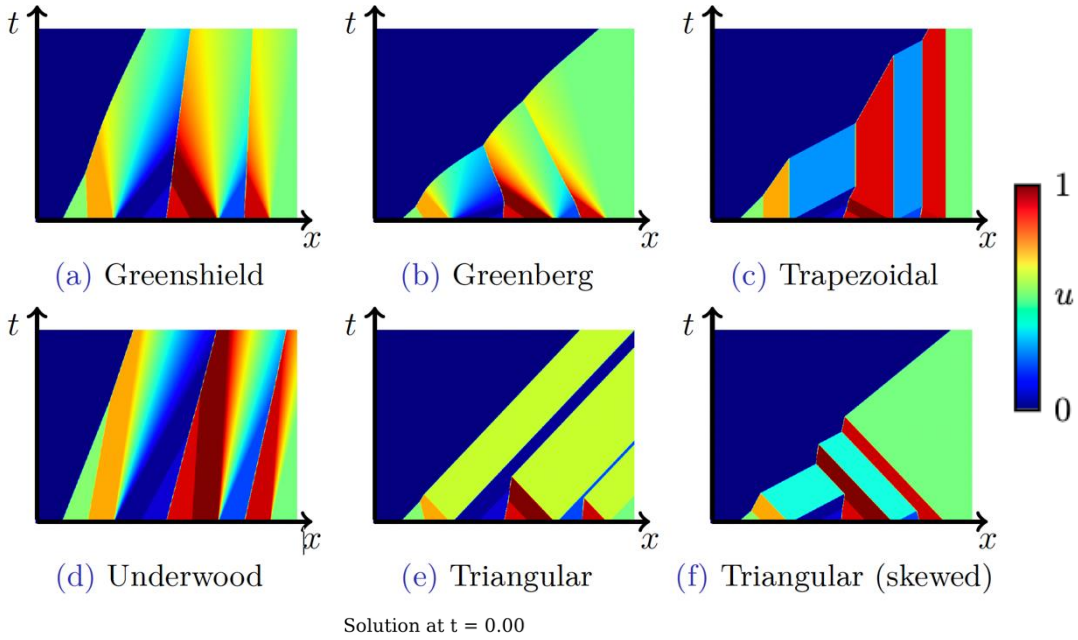
LWR Variants: Different Fluxes, Different Dynamics

Changing the flux function reshapes the evolution of traffic flow

$$\partial_t \rho + \partial_x f(\rho) = 0,$$





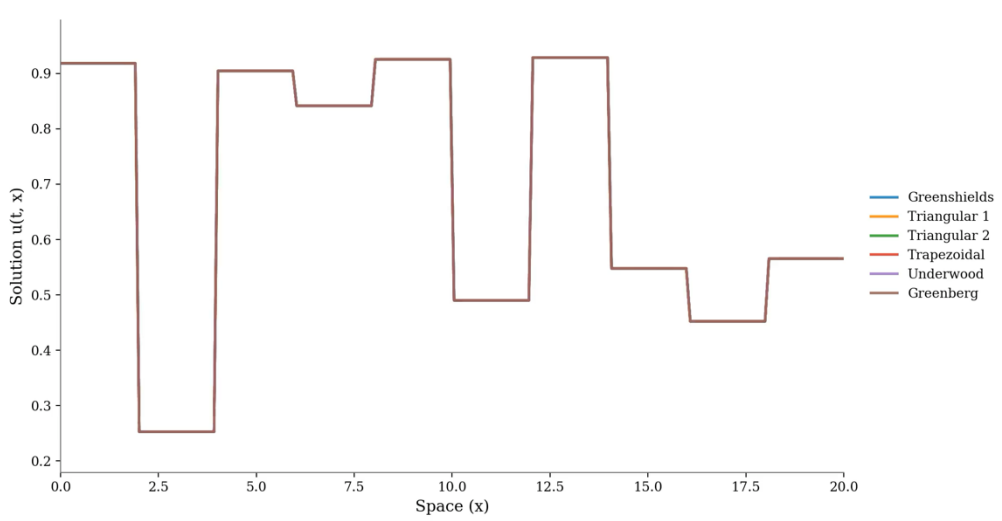


Greenshields Model: Monotonically decreasing and bounded velocity function $v \in C^1([0, \rho_{max}]; \mathbb{R}_+)$, with $v(0) = v_{max}$ and $v(\rho_{max}) = 0$, defined by

$$v: \rho \longmapsto v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right),$$

which leads to the flux $f \in C^2([0, \rho_{max}]; \mathbb{R}_+)$, which is genuinely nonlinear (strictly concave in this case) and defined by

$$f: \rho \longmapsto \rho v(\rho) = v_{max}\rho\left(1 - \frac{\rho}{\rho_{max}}\right).$$



NFVM Predictions Match Ground Truth

Predictions are accurate across LWR variants and diverse initial conditions

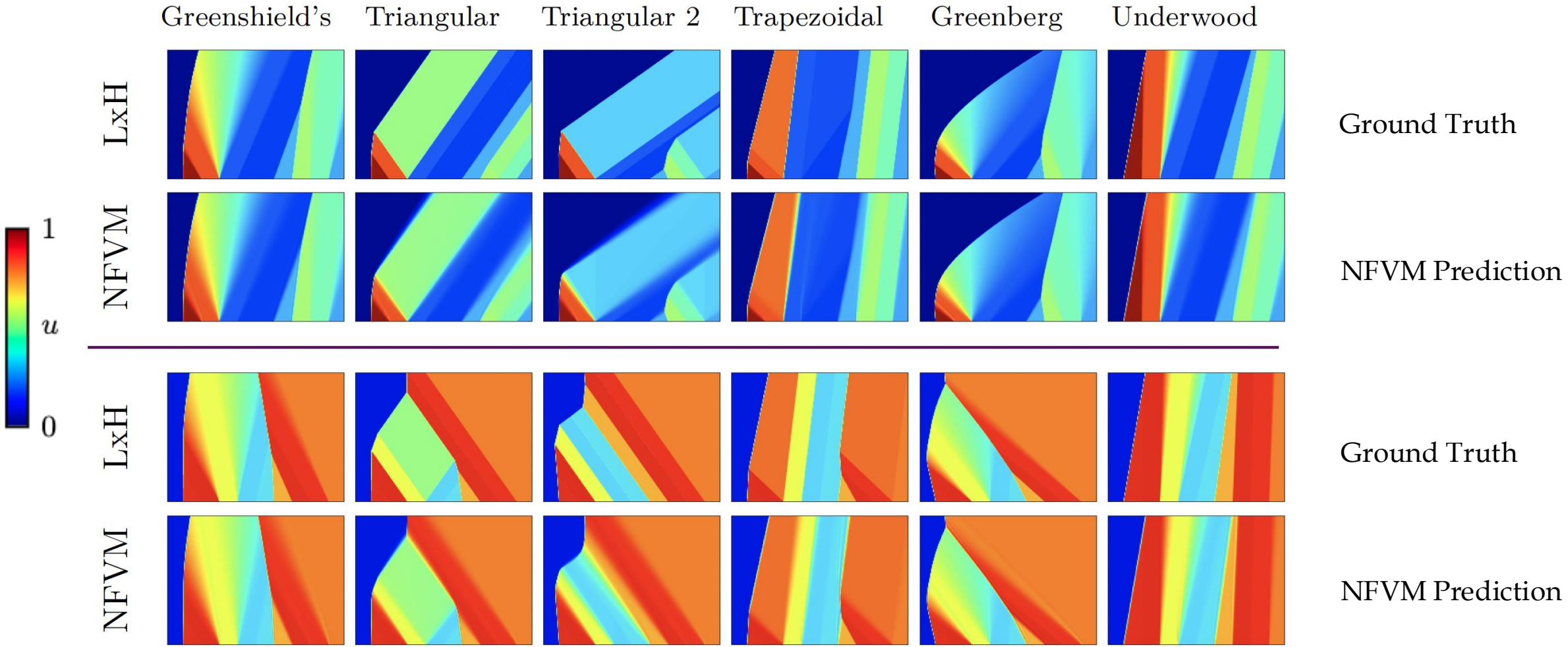


Figure: Ground truth vs. NFVM prediction on 2 different initial conditions.

L2 Error Comparison between NFVM & other Numerical Schemes

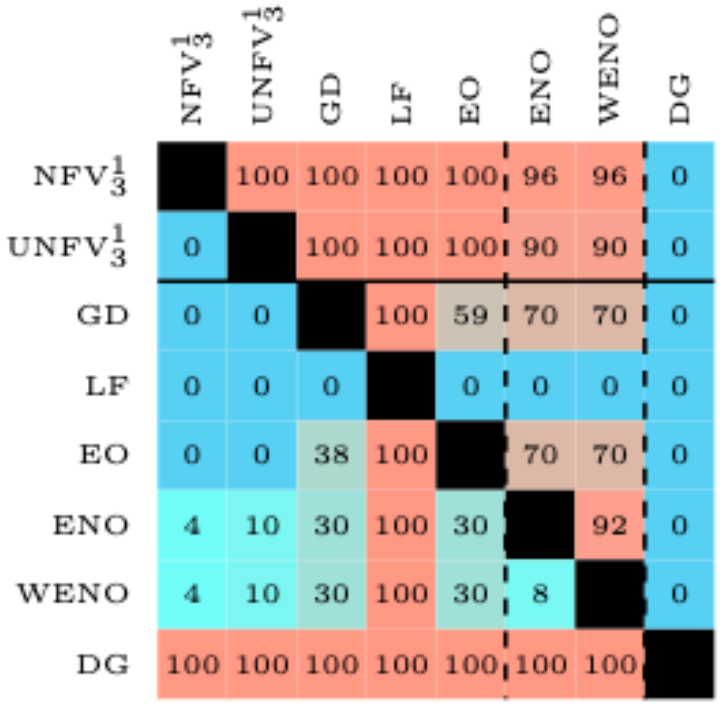
Beats 1st-order FVM, rivals higher-order FVMs, and approaches DG

	1st order FV					Higher order FV		FEM
	$\mathbf{NFV}_3^1$	$\mathbf{UNFV}_3^1$	GD	LF	EO	ENO	WENO	DG
G.shields	$\mathbf{1.3e^{-4}_{\pm 4e^{-5}}}$	$2.0e^{-4}_{\pm 6e^{-5}}$	$4.5e^{-4}_{\pm 2e^{-4}}$	$1.3e^{-2}_{\pm 4e^{-3}}$	$4.5e^{-4}_{\pm 2e^{-4}}$	$6.4e^{-4}_{\pm 4e^{-4}}$	$6.4e^{-4}_{\pm 4e^{-4}}$	$3.1e^{-5}_{\pm 1e^{-5}}$
Tri. 1	$\mathbf{1.4e^{-3}_{\pm 6e^{-4}}}$	$1.9e^{-3}_{\pm 9e^{-4}}$	$2.3e^{-3}_{\pm 1e^{-3}}$	$9.6e^{-3}_{\pm 4e^{-3}}$	$2.3e^{-3}_{\pm 1e^{-3}}$	$2.0e^{-3}_{\pm 2e^{-3}}$	$1.9e^{-3}_{\pm 2e^{-3}}$	$2.6e^{-4}_{\pm 1e^{-4}}$
Tri. 2	$\mathbf{2.4e^{-3}_{\pm 1e^{-3}}}$	$3.1e^{-3}_{\pm 2e^{-3}}$	$3.8e^{-3}_{\pm 2e^{-3}}$	$1.4e^{-2}_{\pm 8e^{-3}}$	$3.8e^{-3}_{\pm 2e^{-3}}$	$5.8e^{-3}_{\pm 4e^{-3}}$	$5.8e^{-3}_{\pm 4e^{-3}}$	$4.1e^{-4}_{\pm 2e^{-4}}$
Trapez.	$\mathbf{1.1e^{-3}_{\pm 4e^{-4}}}$	$1.6e^{-3}_{\pm 7e^{-4}}$	$2.1e^{-3}_{\pm 8e^{-4}}$	$2.5e^{-2}_{\pm 1e^{-2}}$	$2.1e^{-3}_{\pm 8e^{-4}}$	$6.2e^{-4}_{\pm 2e^{-4}}$	$5.3e^{-4}_{\pm 2e^{-4}}$	$2.9e^{-4}_{\pm 1e^{-4}}$
G.berg	$\mathbf{1.4e^{-4}_{\pm 9e^{-5}}}$	$3.8e^{-4}_{\pm 2e^{-4}}$	$4.9e^{-4}_{\pm 2e^{-4}}$	$5.3e^{-3}_{\pm 2e^{-3}}$	$4.9e^{-4}_{\pm 2e^{-4}}$	$1.1e^{-3}_{\pm 6e^{-4}}$	$1.2e^{-3}_{\pm 9e^{-4}}$	$3.4e^{-4}_{\pm 2e^{-3}}$
U.wood	$\mathbf{3.8e^{-4}_{\pm 1e^{-4}}}$	$6.9e^{-4}_{\pm 2e^{-4}}$	$9.2e^{-4}_{\pm 3e^{-4}}$	$2.7e^{-2}_{\pm 1e^{-2}}$	$9.2e^{-4}_{\pm 3e^{-4}}$	$1.1e^{-4}_{\pm 3e^{-5}}$	$9.8e^{-5}_{\pm 2e^{-5}}$	$5.9e^{-5}_{\pm 2e^{-5}}$
Burgers	$\mathbf{8.5e^{-4}_{\pm 3e^{-4}}}$	$1.3e^{-3}_{\pm 6e^{-4}}$	$1.9e^{-3}_{\pm 7e^{-4}}$		$2.6e^{-3}_{\pm 1e^{-3}}$	$2.7e^{-3}_{\pm 1e^{-3}}$	$2.8e^{-3}_{\pm 1e^{-3}}$	$1.0e^{-4}_{\pm 4e^{-5}}$

Discontinuous Galerkin (DG): a well-known FEM for improving the accuracy but incurring high computational costs.

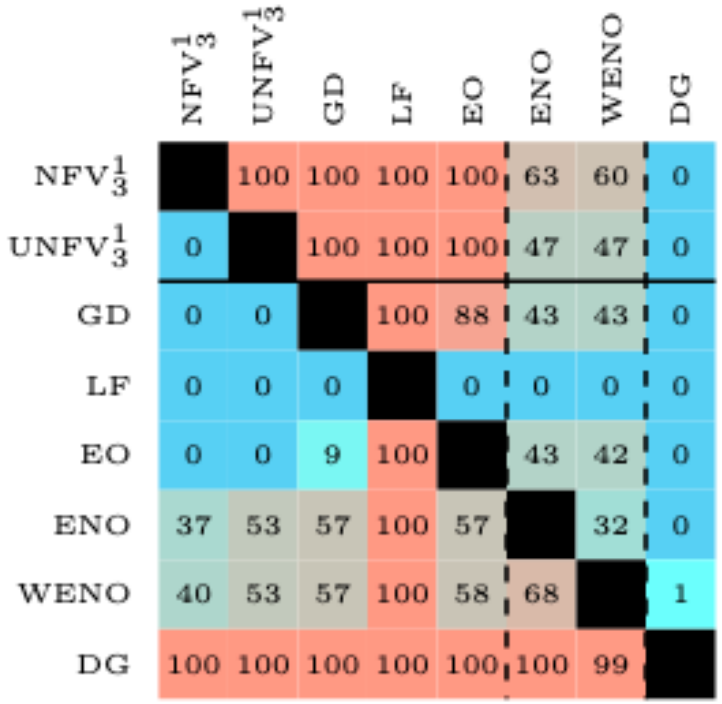
Win Rate Comparison between NFVM & other Numerical Schemes

NFVM dominates in pairwise accuracy



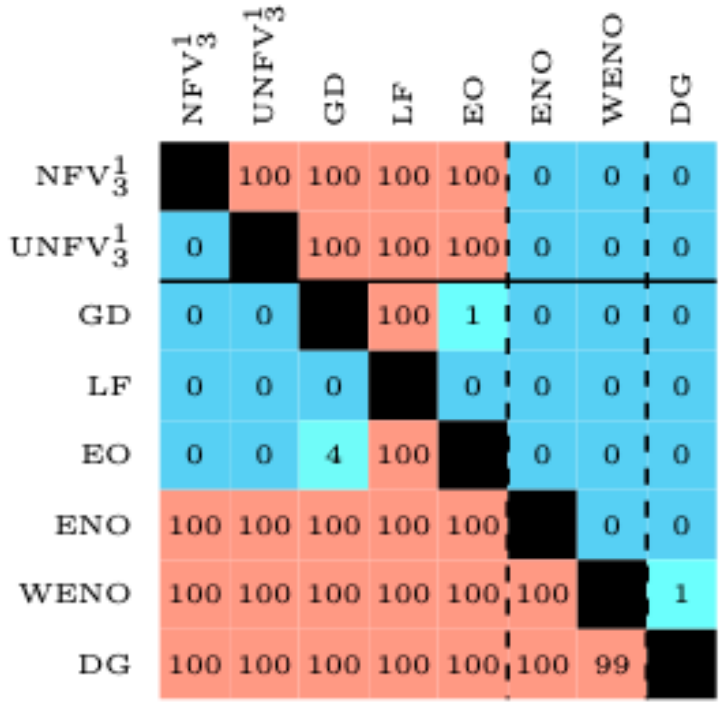
	NFV_3^1	$UNFV_3^1$	GD	LF	EO	ENO	WENO	DG
NFV_3^1		100	100	100	100	96	96	0
$UNFV_3^1$	0		100	100	100	90	90	0
GD	0	0		100	59	70	70	0
LF	0	0	0		0	0	0	0
EO	0	0	38	100		70	70	0
ENO	4	10	30	100	30		92	0
WENO	4	10	30	100	30	8		0
DG	100	100	100	100	100	100	100	

(a) Greenshields



	NFV_3^1	$UNFV_3^1$	GD	LF	EO	ENO	WENO	DG
NFV_3^1		100	100	100	100	63	60	0
$UNFV_3^1$	0		100	100	100	47	47	0
GD	0	0		100	88	43	43	0
LF	0	0	0		0	0	0	0
EO	0	0	9	100		43	42	0
ENO	37	53	57	100	57		32	0
WENO	40	53	57	100	58	68		1
DG	100	100	100	100	100	100	99	

(b) Triangular



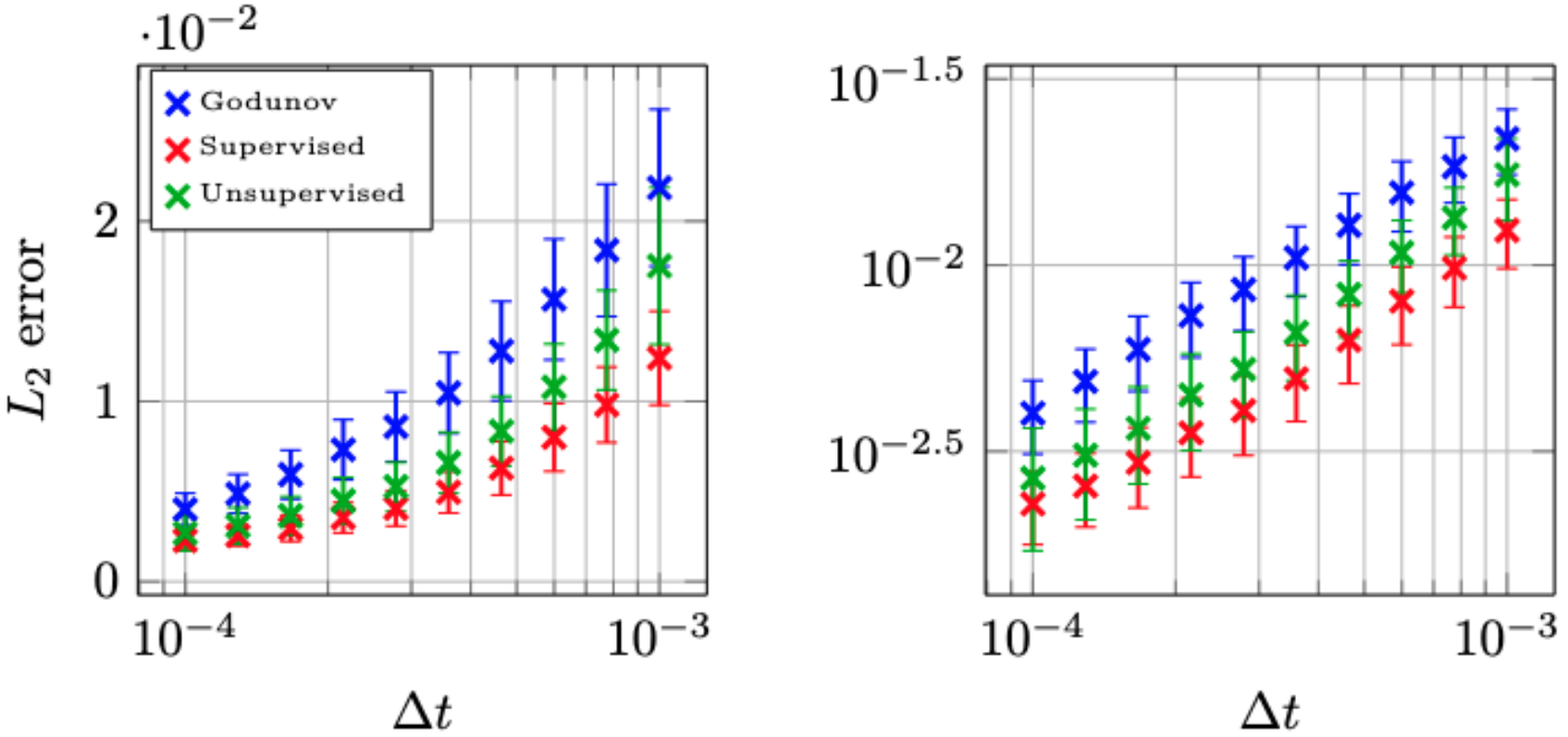
	NFV_3^1	$UNFV_3^1$	GD	LF	EO	ENO	WENO	DG
NFV_3^1		100	100	100	100	0	0	0
$UNFV_3^1$	0		100	100	100	0	0	0
GD	0	0		100	1	0	0	0
LF	0	0	0		0	0	0	0
EO	0	0	4	100		0	0	0
ENO	100	100	100	100	100		0	0
WENO	100	100	100	100	100	100		1
DG	100	100	100	100	100	100	99	

(c) Underwood

$$\textbf{Win Rate\%} = \frac{\#\ of\ the\ evaluation\ set\ that\ row\ scheme\ outperforms\ the\ column\ scheme}{\#\ of\ evaluation\ set}\ \textbf{X 100\%}$$

Predictive Capabilities of the NFVM & UNFVM

Both schemes outperform and converge beyond Godunov

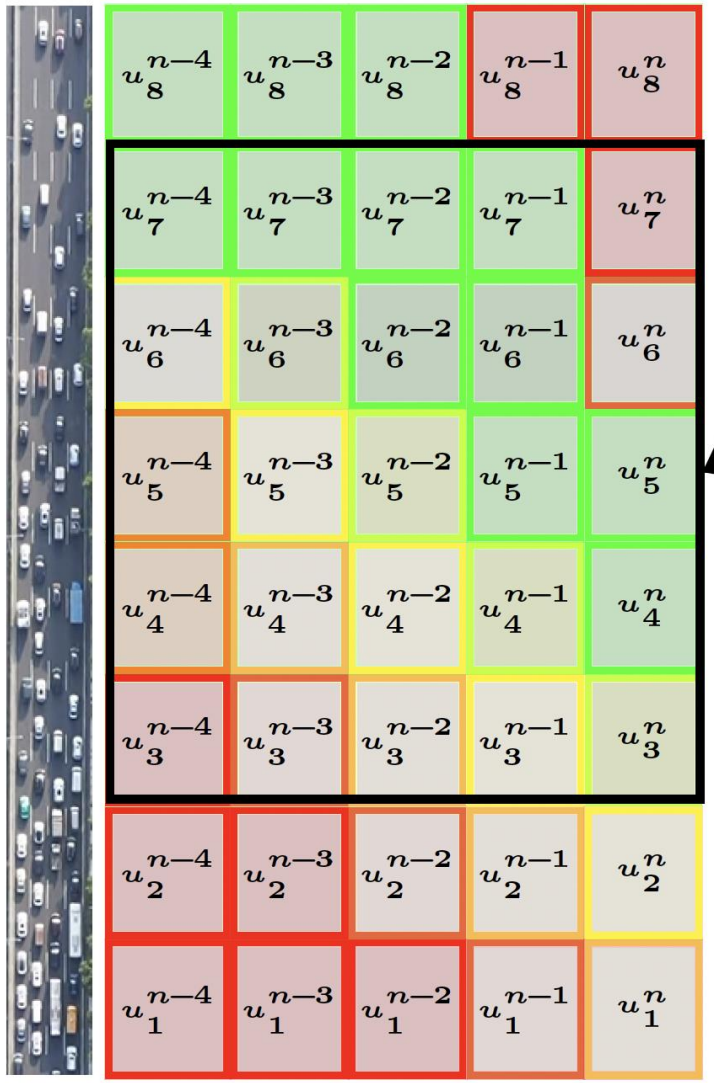


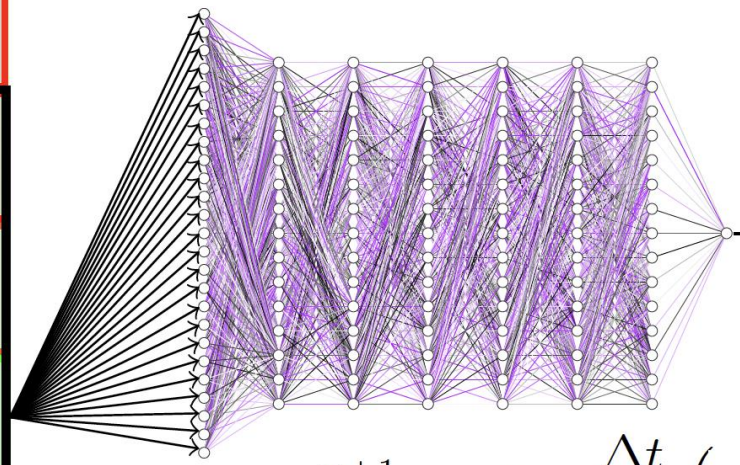
(a) Log-Linear scale. (b) Log-Log scale.

Figure: **Convergence plot for the Greenshield's flux.** Standard deviation is shown as error bars. The ratio $\Delta t/\Delta x = 0.1$ is fixed.

Unleashing Stencils: Consider 5 in Space and 5 in Time

NN offers the flexibility to scale input size across both space and time



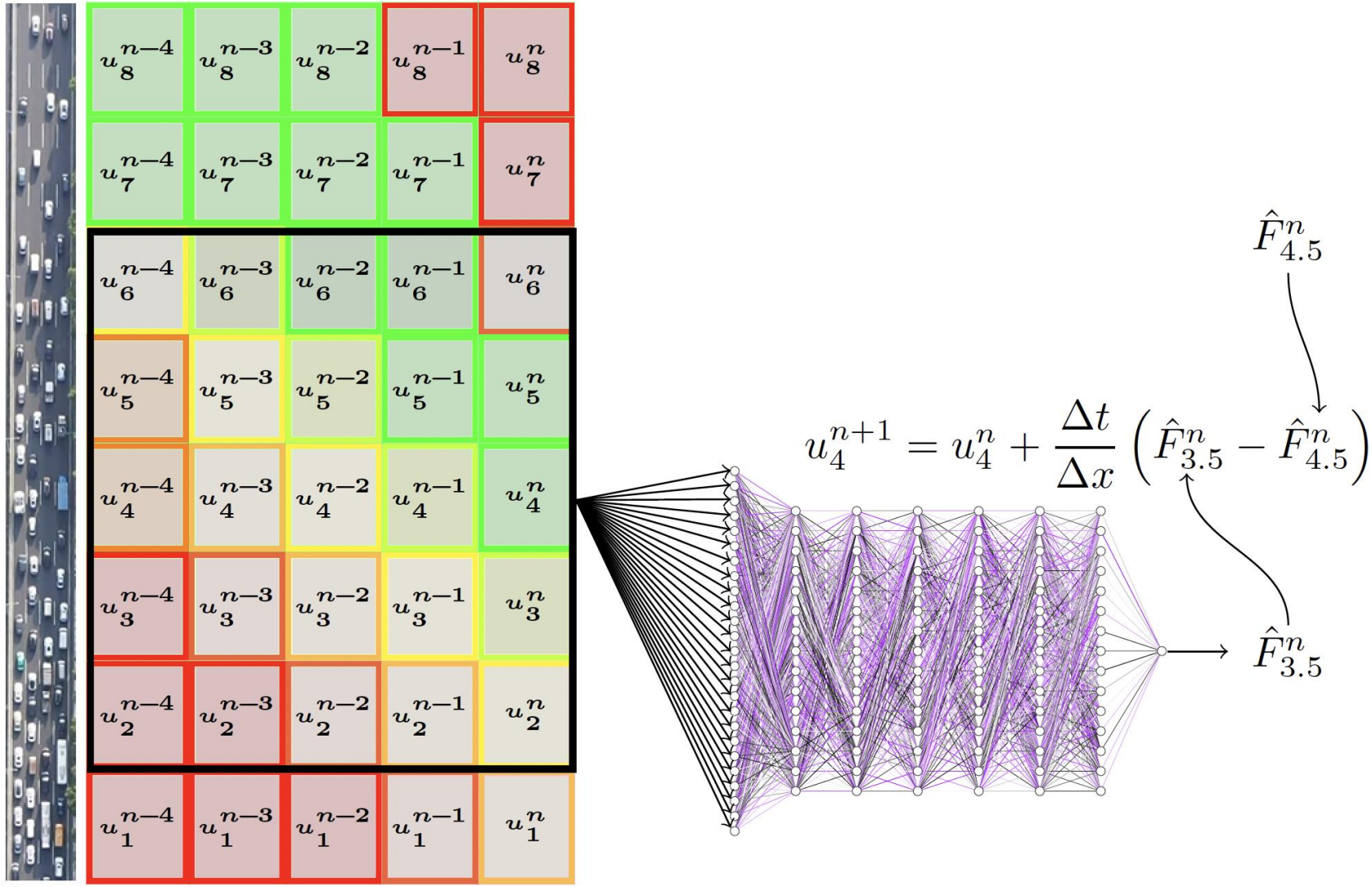


$$\hat{F}_{4.5}^n$$

$$u_4^{n+1} = u_4^n + \frac{\Delta t}{\Delta x}\left(\quad - \hat{F}_{4.5}^n \right)$$

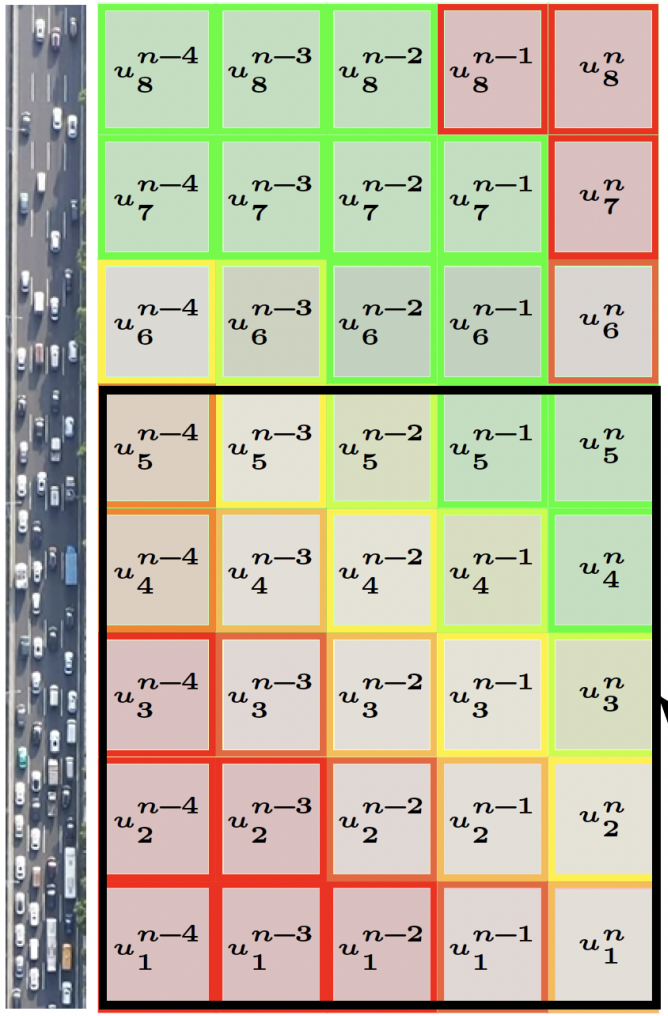
Unleashing Stencils: Consider 5 in Space and 5 in Time

NN offers the flexibility to scale input size across both space and time



Unleashing Stencils: Consider 5 in Space and 5 in Time

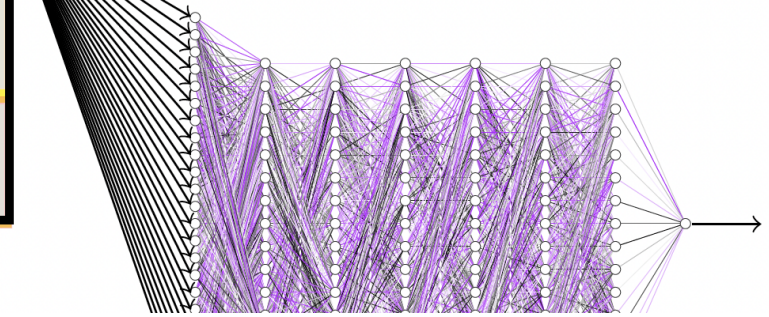
NN offers the flexibility to scale input size across both space and time



$$u_4^{n+1} = u_4^n + \frac{\Delta t}{\Delta x}\left(\hat{F}_{3.5}^n - \hat{F}_{4.5}^n\right)$$

$\hat{F}_{4.5}^n$

$\hat{F}_{3.5}^n$



$\hat{F}_{2.5}$

Extended NFVM Remains Low Computational Cost

Small, efficient, and designed to fit into the FVM update loop

Vanilla NFVM:

Architecture: 1D CNN

- Kernel size: # of stencils used (2 for 1st order) for the first layer
- Layers: 6
- Hidden layer size: 15
- Learned parameters: ~1200

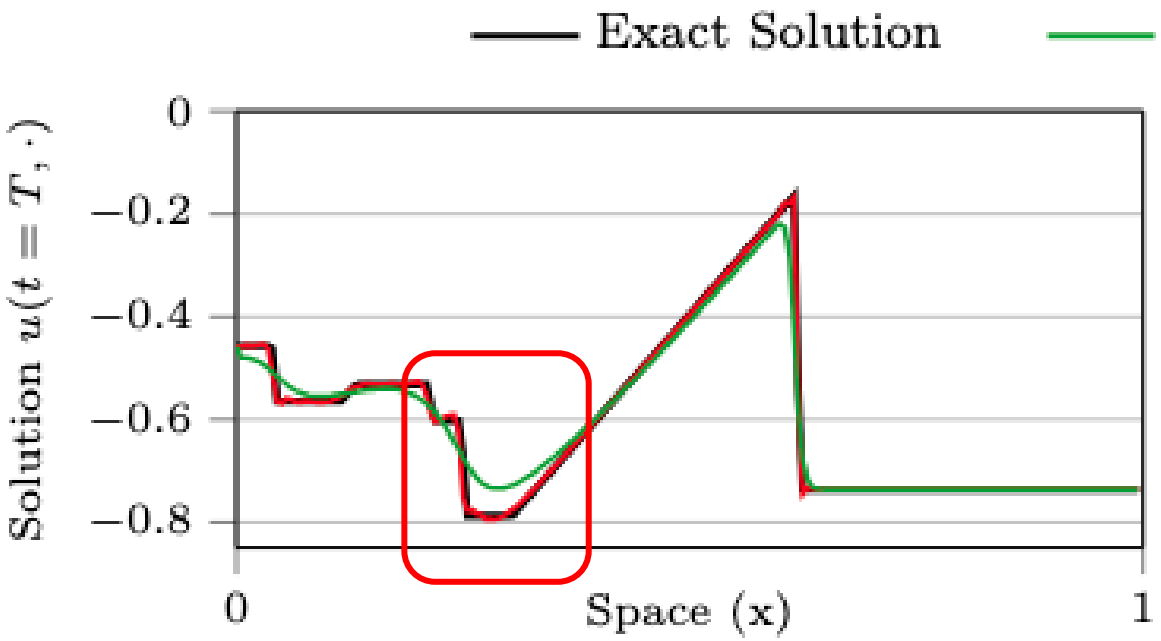
NFVM_5^5:

Architecture: 1D CNN

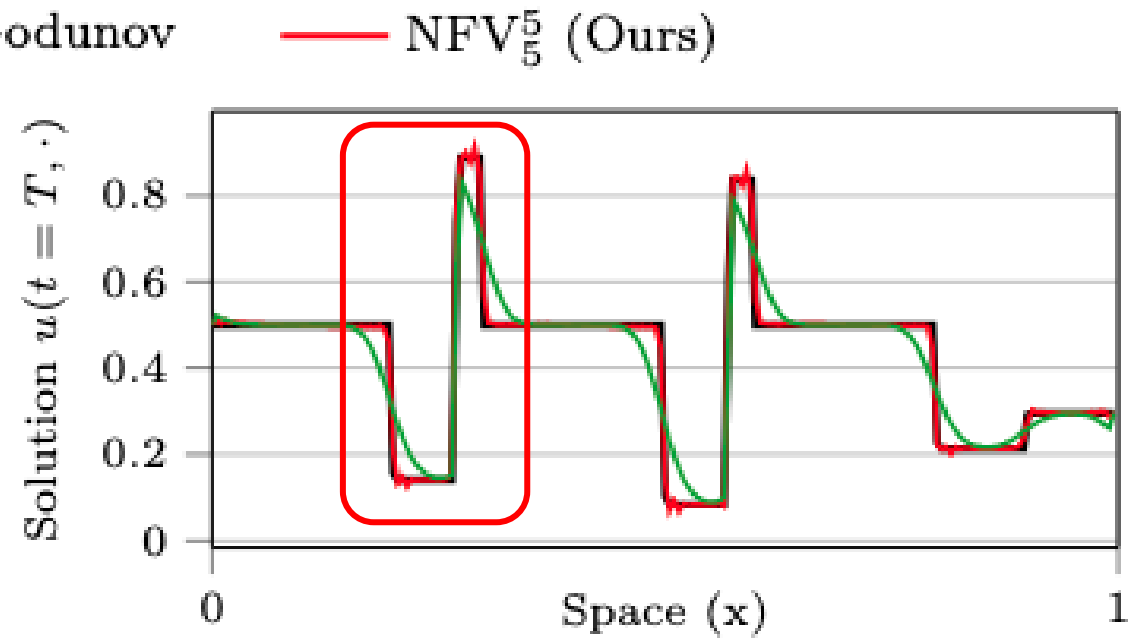
- Kernel size: 6 of stencils used for the first layer
- Timesteps: 5
- Layers: 6
- Hidden layer size: 16
- Learned parameters: ~1600

$\mathbf{NFVM_5^5}$ Outperforms Godunov Across Equations

Improved accuracy on Burgers and LWR, closely matching the exact solution



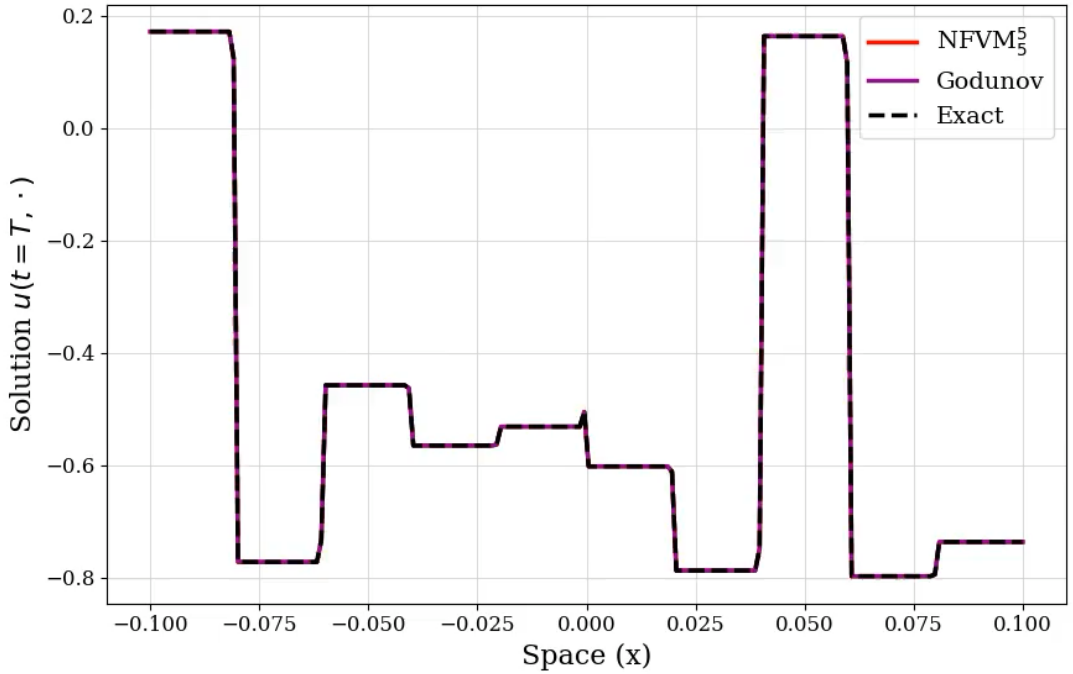
Comparison of the final density of the **Burgers' equation**



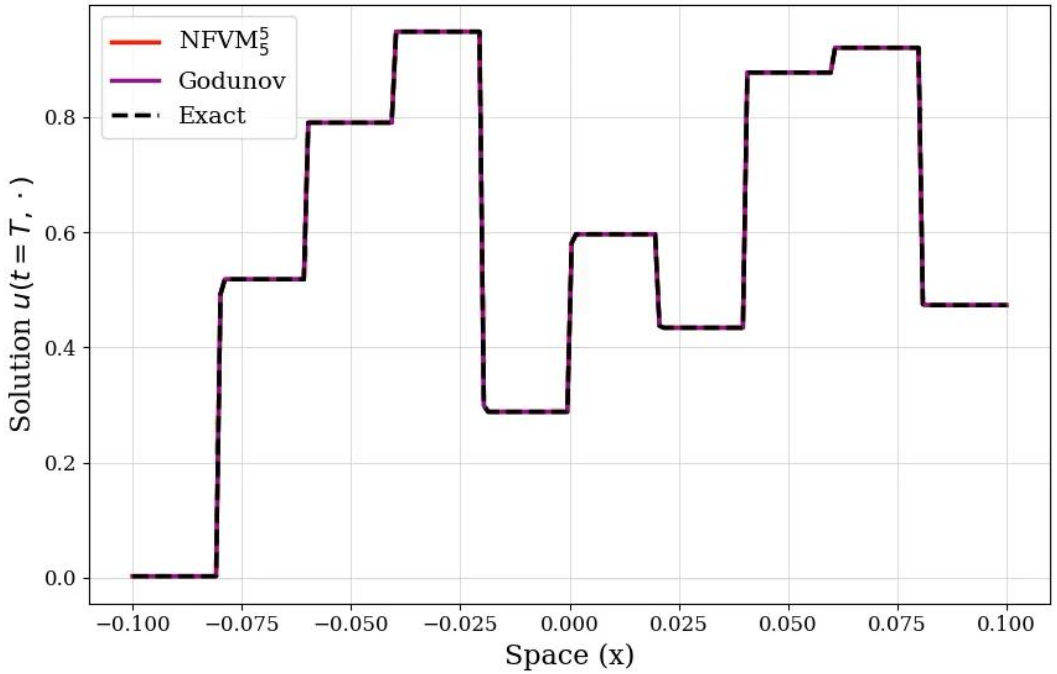
Comparison of the final density of the **LWR triangular equation**

$\mathbf{NFVM_5^5}$ Tracks the Exact Solution Over Time

Consistently outperforms Godunov throughout the full evolution



$\mathbf{NFVM_5^5}$ outperforms Godunov at predicting a solution of the **Burger's equation**



$\mathbf{NFVM_5^5}$ outperforms Godunov at predicting a solution of the **LWR triangular equation**

$\mathbf{NFVM_5^5}$: High Accuracy, Low Complexity

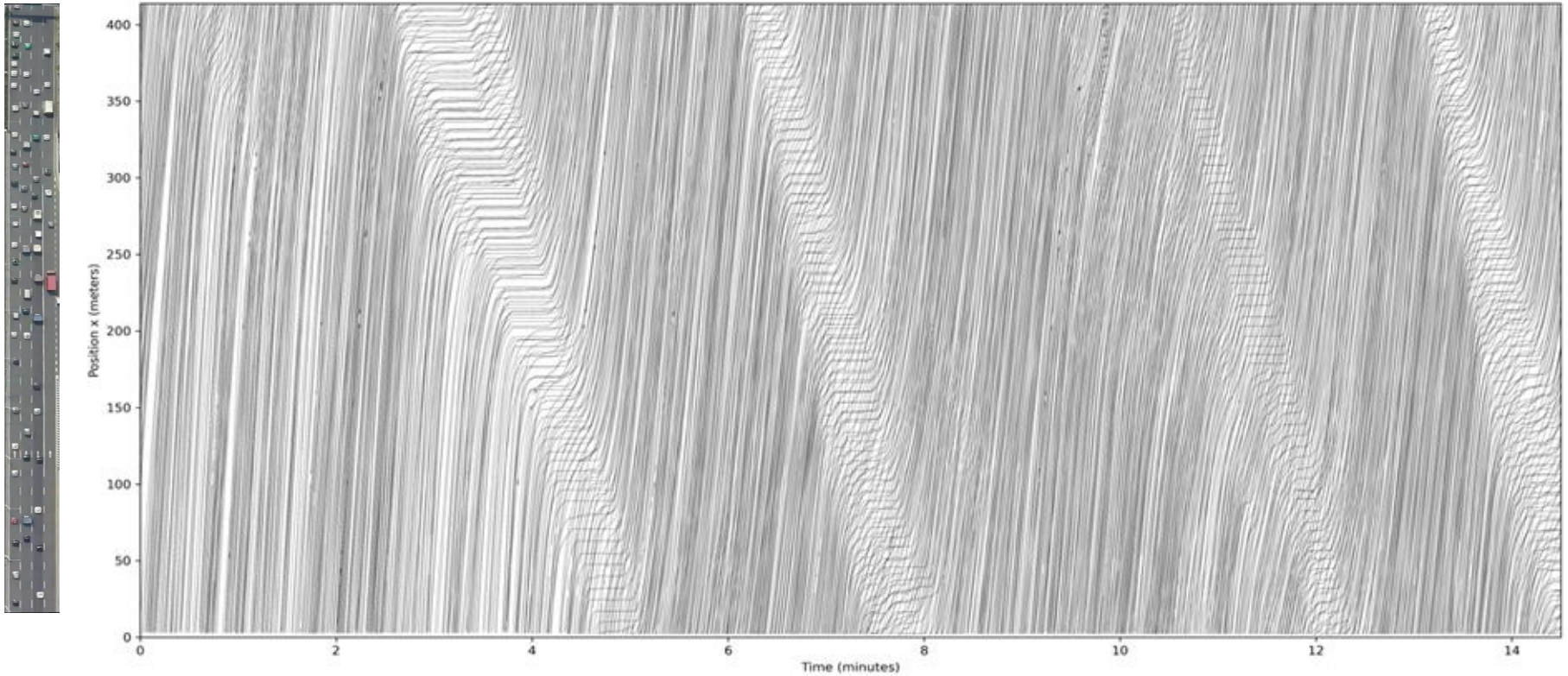
Up to 10x better than $\mathrm{NFVM_3^1}$, near DG accuracy with first-order FVM simplicity

	1st order FV	Higher order FV	(3, 1) variant	(5, 5) variant	FE
	Godunov	WENO	$\mathbf{NFV_3^1}$	$\mathbf{NFV_5^5}$	DG
Burgers'	$1.8e^{-3}_{\pm 6e^{-4}}$	$2.6e^{-3}_{\pm 1e^{-3}}$	$8.3e^{-4}_{\pm 3e^{-4}}$	$2.2e^{-4}_{\pm 1e^{-4}}$	$\mathbf{1.0e^{-4}_{\pm 4e^{-5}}}$
Greenshields	$4.1e^{-4}_{\pm 1e^{-4}}$	$6.9e^{-4}_{\pm 4e^{-4}}$	$1.2e^{-4}_{\pm 4e^{-5}}$	$4.6e^{-5}_{\pm 3e^{-5}}$	$\mathbf{4.2e^{-5}_{\pm 2e^{-5}}}$
Triangular	$2.2e^{-3}_{\pm 1e^{-3}}$	$2.0e^{-3}_{\pm 2e^{-3}}$	$1.3e^{-3}_{\pm 6e^{-4}}$	$2.9e^{-4}_{\pm 2e^{-4}}$	$\mathbf{2.7e^{-4}_{\pm 1e^{-4}}}$

From PDE Benchmarks to Traffic Field Data

Apply NFVM to drone-captured highway data (B3D dataset)

Figure: Trajectories are extracted from drone-recorded highway video data.



- **12** videos recorded by a drone above a highway (4K, 30 FPS)
- **15** mins of data
- **4** visible stop-and-go waves

Fangyu Wu et al. "Decentralized vehicle coordination: The Berkeley deepdrive drone dataset". In: arXiv preprint arXiv:2209.08763(2022).

Four stop-and-go waves exist

Density color-coded time-space diagram reveals shockwave-like patterns

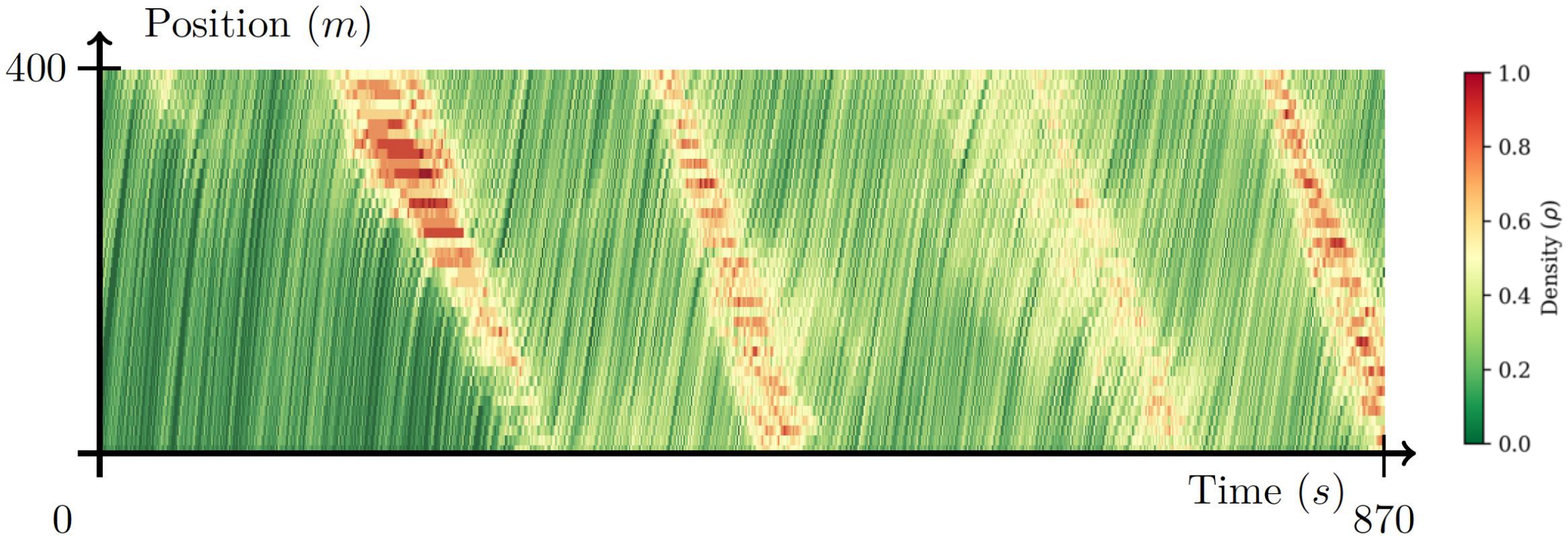


Figure: Time-space diagram of vehicle trajectories extracted from the drone video, color-coded by density.

Train on 8%, Predict the Remaining 92%

Only one wave (70s) used for training; model generalizes to the full dataset

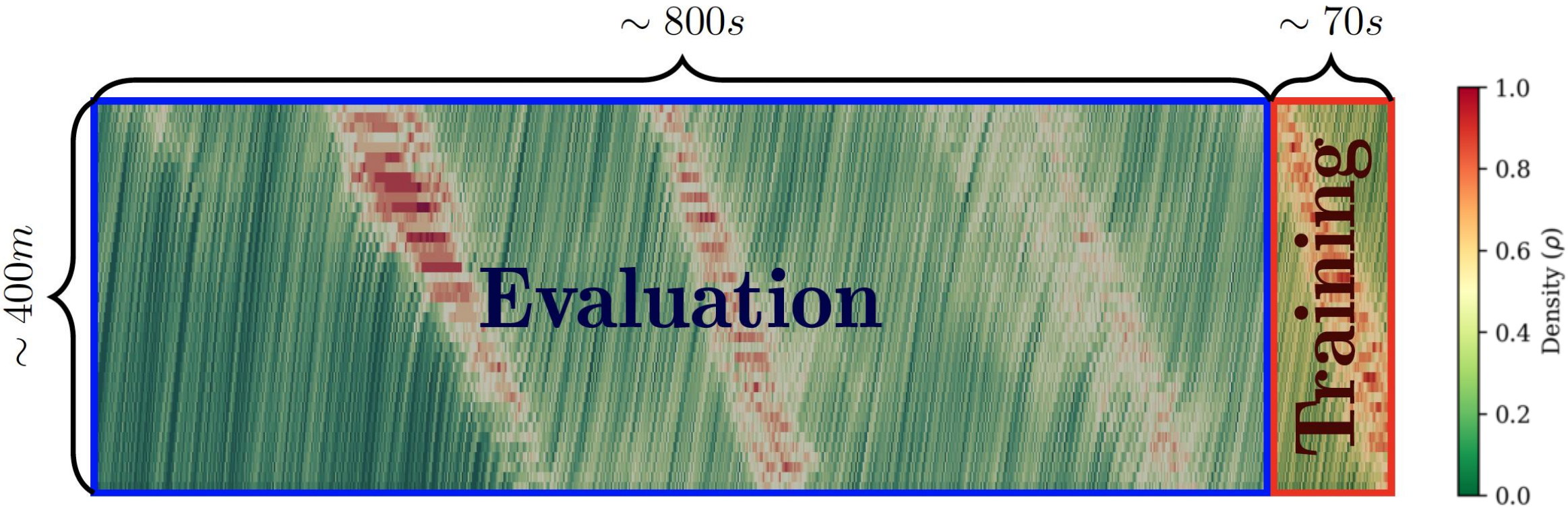


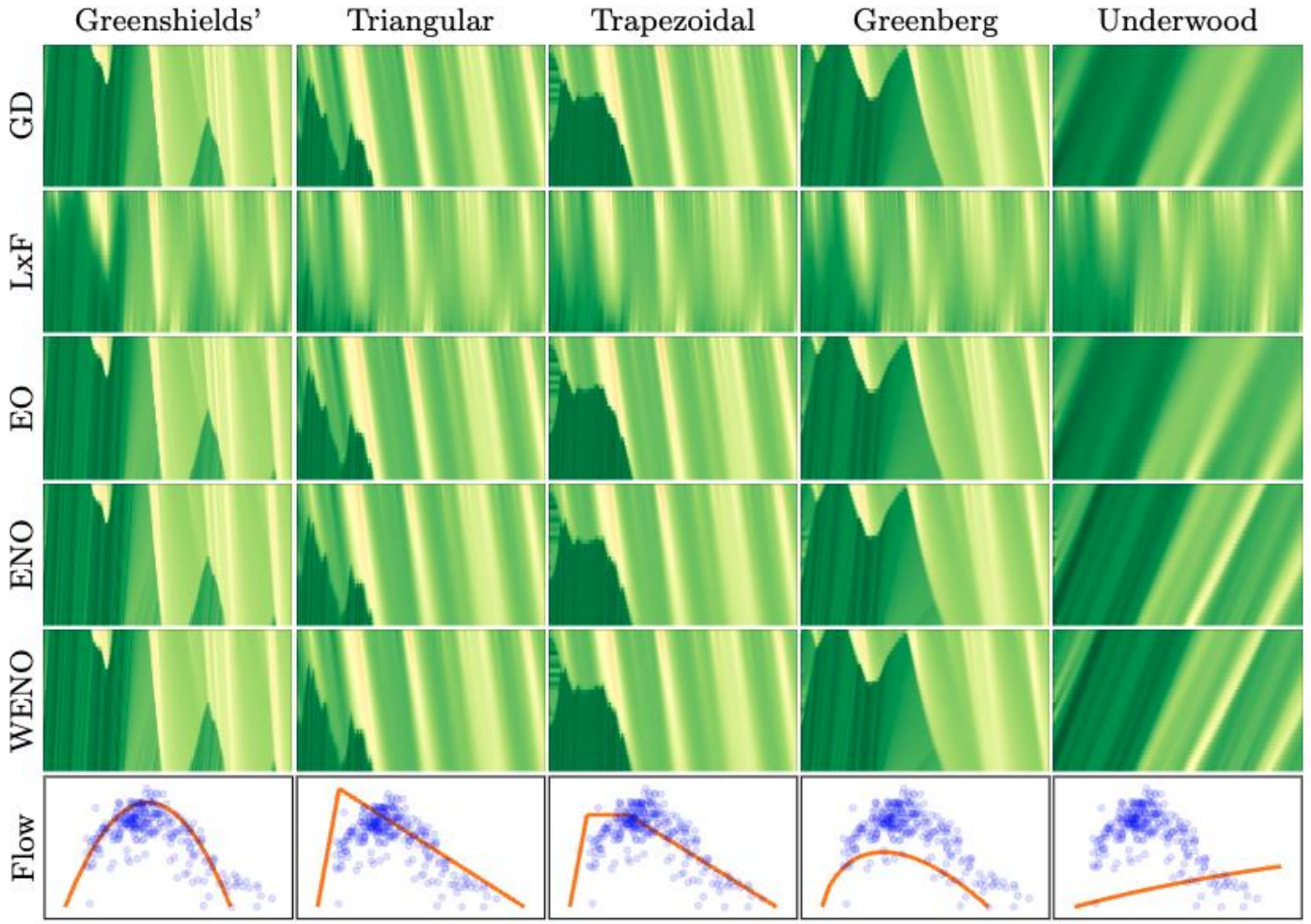
Figure: Time-space diagram of vehicle trajectories extracted from the drone video, color-coded by density.

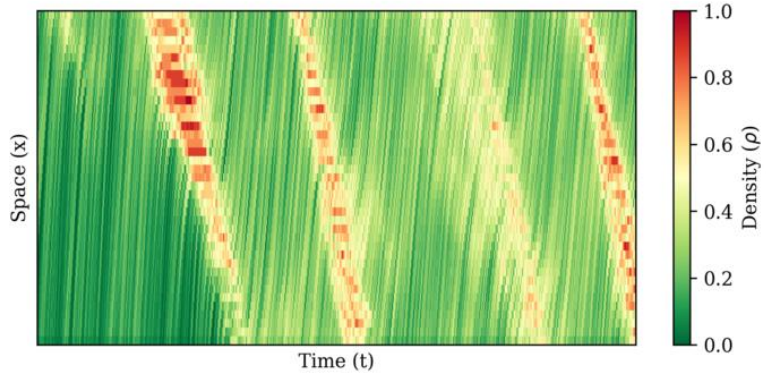
Forward Computation with Regular FV Schemes

Each flux/scheme pair produces a different forecast, most of them with significant error

Fluxes

Schemes

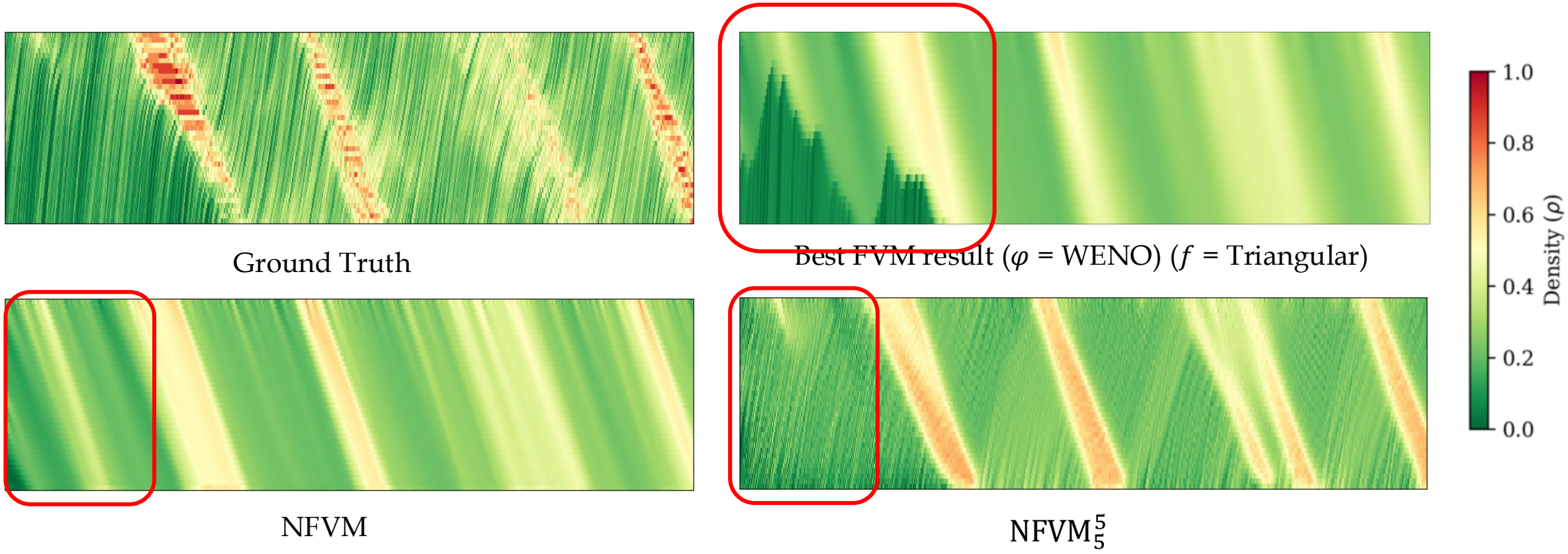




Ground Truth

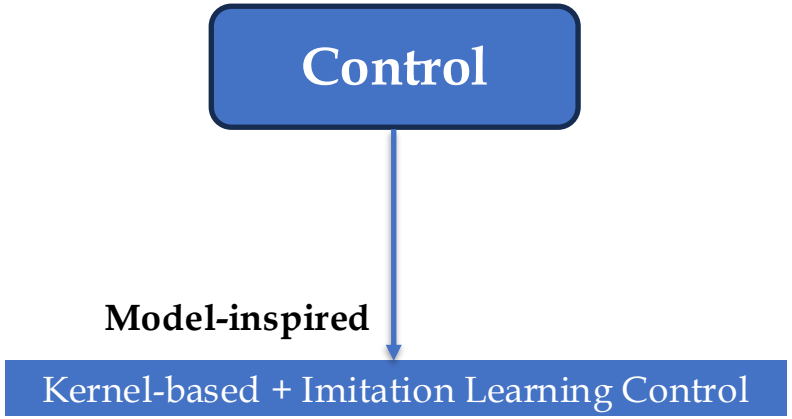
Winners' Comparison

NFVM outperforms all FVM baselines (flux/scheme pairs);
NFVM_5^5 captures key effects like wave dissipation, free-flow directions, high densities



Model-Inspired Control to Stabilize Flow and Save Energy

Design different control algorithms for AVs under different sensing and actuation paradigms



[F†AKB] TR-C;
[KFB] **IEEE ITSC;**
[F†KWLDB] IEEE IV

Source of Hardness	**Partial observability; limited actuation of AVs**
Solution Method	**Kernel-based** planner; Imitation Learning control

*** Equal first author; † Corresponding author**

Current Mixed Autonomy Stage

Partially automated; Limited number; Heterogenous

Automation Level:
Local Observe
(~Level 2, ACC)

Coordination Level:
Only few automated vehicles
are available on the road

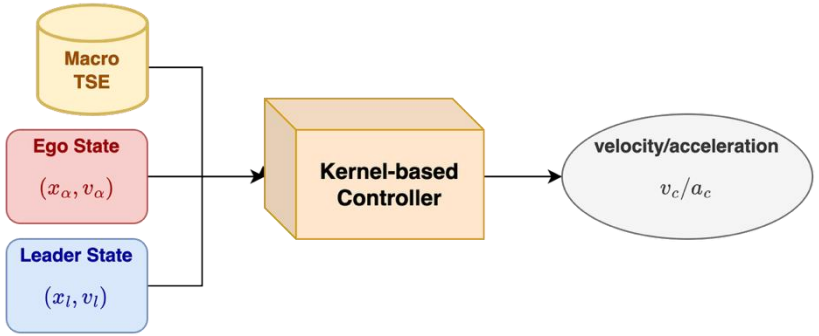
System Level:
Heterogenous Automation
(Various Sensing and
Actuation Abilities)

Can we utilize these limited number of partially automated and heterogeneous vehicles to stabilize traffic conditions for all vehicles within the system?

Model-Inspired Control to Stabilize Flow and Save Energy

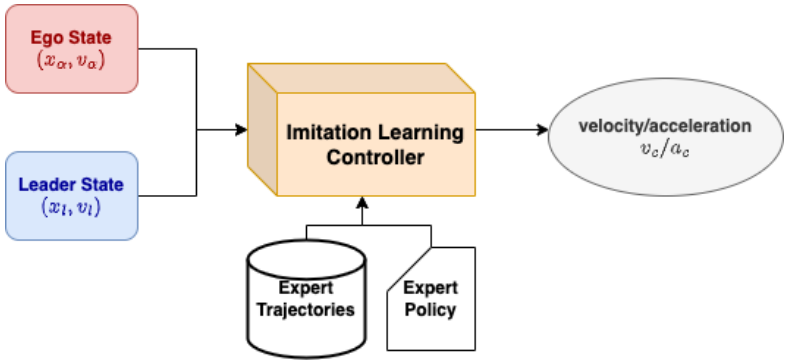
Design different control algorithms for AVs under different sensing and actuation paradigms

Kernel-based planner:
For strategic, long-horizon speed planning

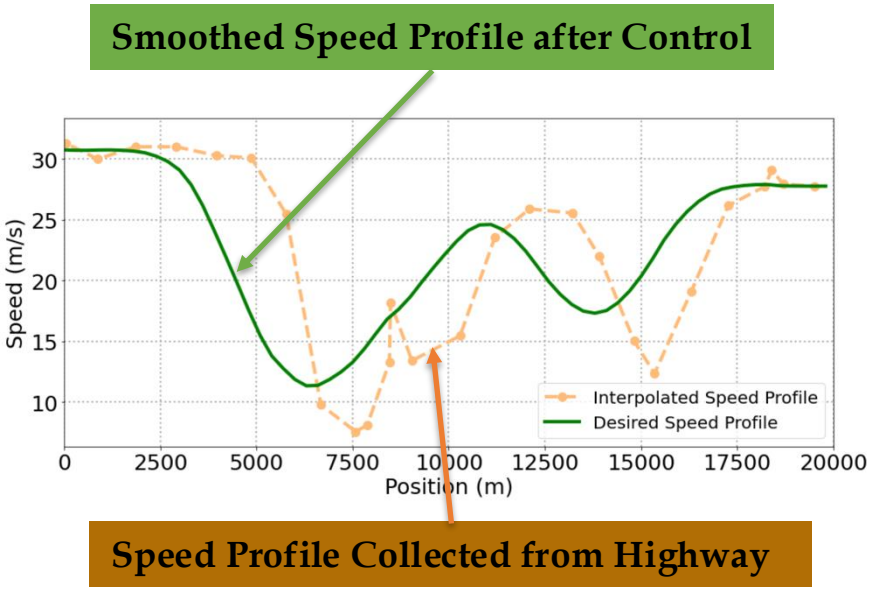


Use Kernel Planning for Target Speed Design

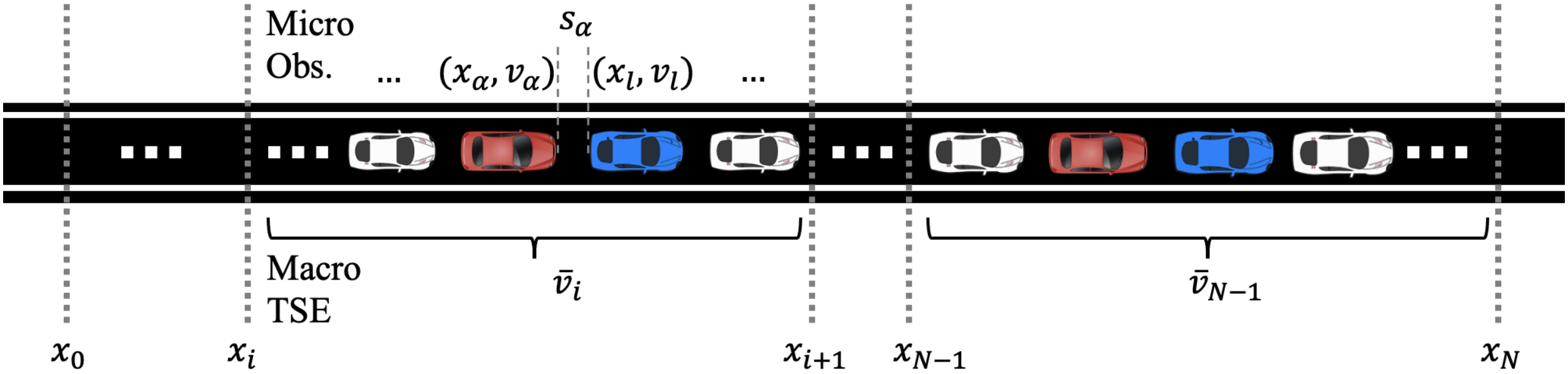
Imitation Learning Control:
To mimic optimal driving behavior under uncertainty.



Treat Kernel-based planner as expert to get rid of the reliance on TSE



In simulation with real traffic trajectories: **> 15 %** Energy Savings for whole system with **4%** AV Penetration



Microscopic Observations (Local states)

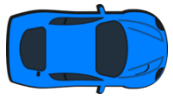
(x_α, v_α) Ego state

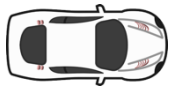
(x_l, v_l) Lead state

Macroscopic TSE (Global states)

$(x_i, \bar{v}_i)$ i is the i^{th} segment Space-mean speed for each segment


Controlled AV


Sensed HV


Other HV

x_α: controlled AV's position
v_α: controlled AV's speed
s_α: space gap between ego car and lead car
x_l: sensed HV's position
v_l: sensed HV's speed
x_i: the ith segment position
$\bar{v}_i$: the ith segment space-mean speed

Global sensing: (x_α, v_α) Ego state (x_l, v_l) Lead state $(x_i, \bar{v}_i)$ TSE

Local sensing: (x_α, v_α) Ego state (x_l, v_l) Lead state

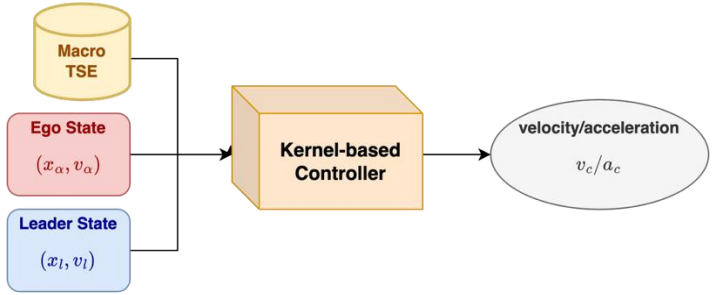
Semi sensing: (x_α, v_α) Ego state $(x_i, \bar{v}_i)$ TSE

Control Strategy Design under Different Sensing Paradigms

The "Teacher" → "Student"

Global Sensing

(x_α, v_α) Ego state (x_l, v_l) Lead state $(x_i, \bar{v}_i)$ TSE

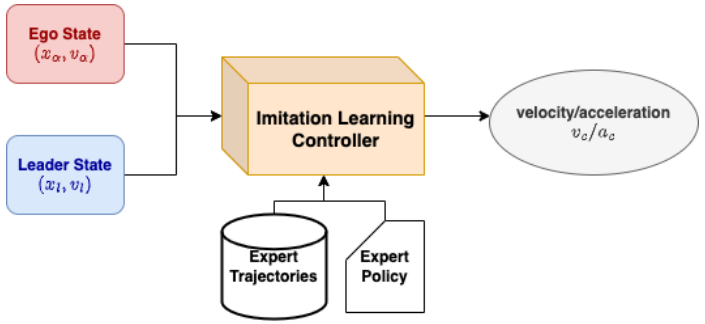


Kernel-based Feedback Control (The Teacher)

Use Kernel Planning for Target Speed Design; Leave Gap Regulation for Vehicle Level

Local Sensing

(x_α, v_α) Ego state (x_l, v_l) Lead state



Imitation Learning Control (The Student)

Treat Kernel-based as expert to get rid of the reliance on TSE

Kernel-based Feedback Control Architecture

$$v_c = \max(0, \min(v_{\text{target},\alpha} + k_p(h_\alpha - h_{\text{des}}) + k_d(v_l - v_\alpha), v_{\text{fs}})$$

Target speed | Gap Regulation | Safety Filter

$$v_{\text{target},\alpha} = f_t(x_\alpha, v_\alpha, x_l, w, [x_i, \bar{v_i}])$$

$$v_{\text{g}} = f_g(x_\alpha, v_\alpha, x_l, v_l, k_p, k_d, h_{\text{des}})$$

$$v_{\text{fs}} = f_s(x_\alpha, v_\alpha, x_l, v_l, \tau_s, s_{\min}, h_{\min})$$

Constants:

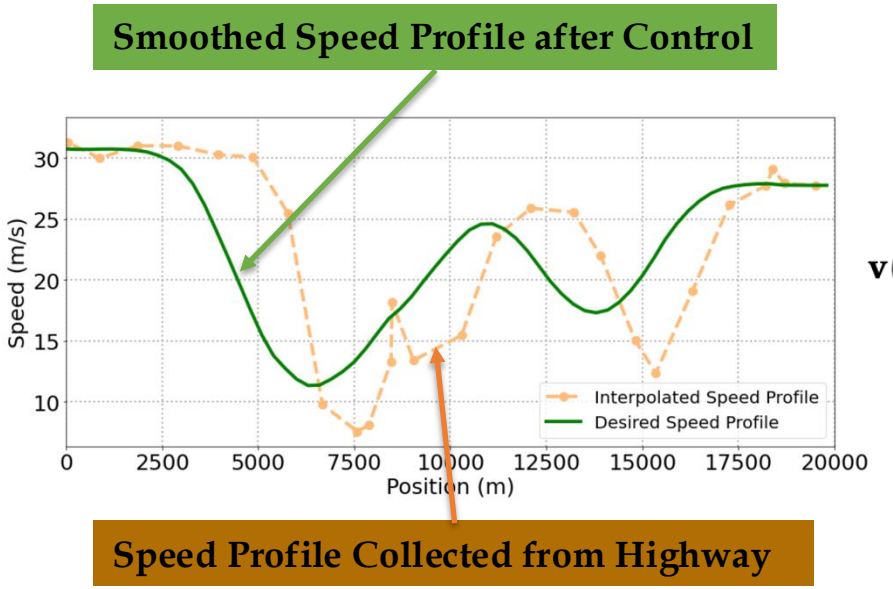
w: sliding window length for target speed estimation

k_p, k_d: proportional gain and differential gain

h_{des}: desired time gap

τ_s: safety decision-making time horizon

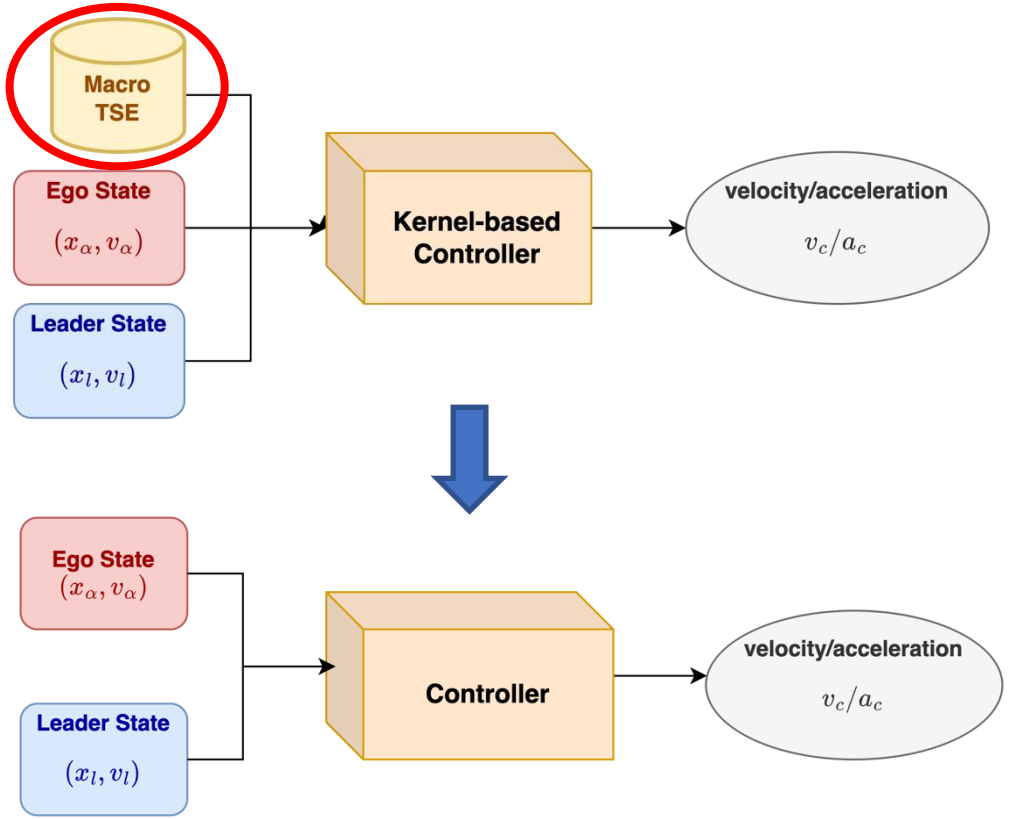
$s_{\min}, h_{\min}$: minimum space and time gap for safety concern



$$\mathbf{v}(t_\circ, x_\alpha) = \frac{\int_{x=x_\alpha}^{x_\alpha+\mu} K(x_\alpha, x) v(t_\circ, x) dx}{\int_{x=x_\alpha}^{x_\alpha+\mu} K(x_\alpha, x) dx},$$

Bridging the gap between global and local sensing through Imitation Learning

How can we design a controller that just use local sensing to stabilize traffic conditions?



Imitation Learning

Imitation Learning is a framework for learning a behavior policy from demonstrations.

$$\pi_\theta(s_t): S \rightarrow A$$

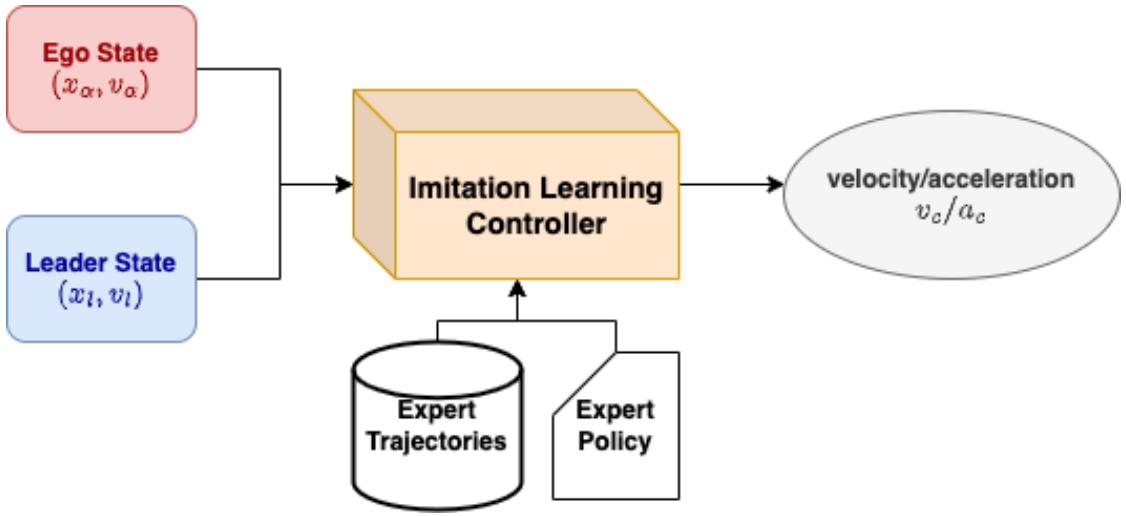
$$\theta = argmin_\theta[E_D[L(\pi_\theta(s_i), a_i)]]$$

$$S = \{(x_\alpha^i, v_\alpha^i, x_l^i, v_l^i)\}_{i=1}^N \quad A = \{(v_c^i)\}_{i=1}^N$$

$$D = \{(s_i, a_i)\}_{i=1}^N$$

$L(\cdot,\cdot)$: selected distance metric
π_θ: desired policy
θ: parameters for the policy
s_i: ith state
a_i: ith action
N: number of state-action pairs
D: dataset. of state action pairs

Imitation Learning Structure & Algorithm



We can treat those global sensing control strategies as expert and imitate them to reduce the reliance on global sensing.

Help to alleviate the common problem in IL: Covariate Shift

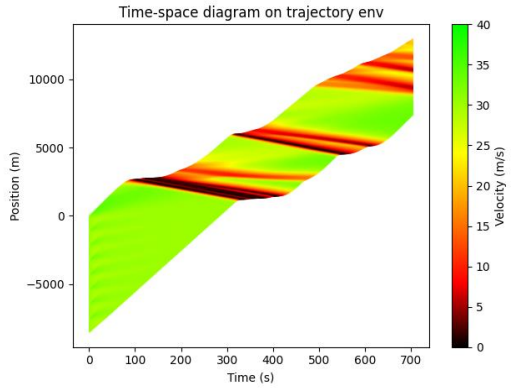
DAgger Algorithm

Initialize $D \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ to n do
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $D_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert **(kernel-based controller).**
  Aggregate datasets: $D \leftarrow D \cup D_i$.
  Train classifier $\hat{\pi}_{i+1}$ on D.
end for
Return best $\hat{\pi}_i$ on validation.

s: state
N: number of state-action pairs
D: dataset. of state action pairs
π^*: expert policy
$\hat{\pi}_i$: policy trained to minimize
β_i: a decreasing coefficient

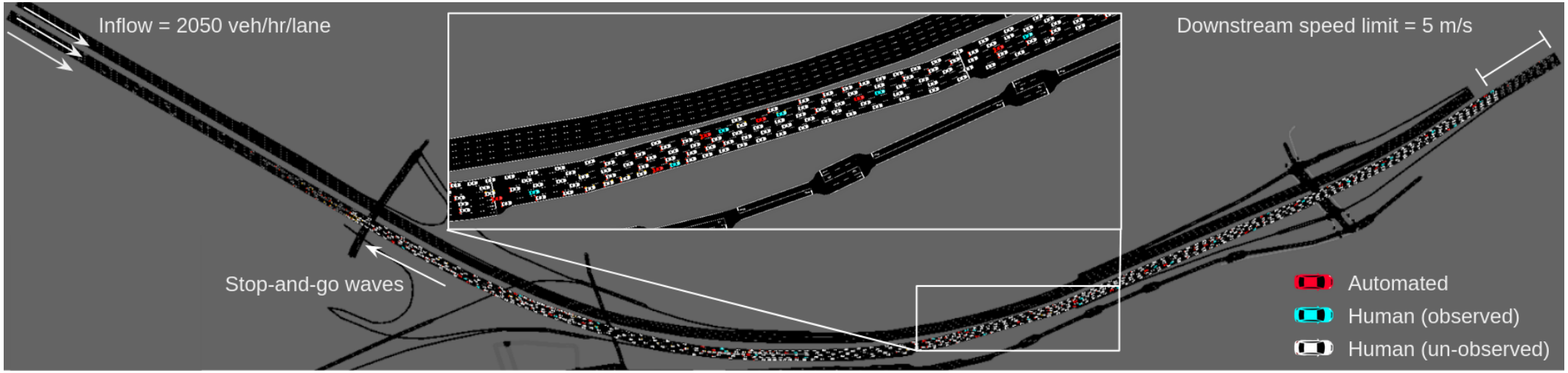
Trained and Evaluated in Two Cases

Case 1: Single Lane + Recorded Stop-and-go Trajectories



Example Time-Space diagram with 200 vehicles generated from our own simulator with recorded I-24 trajectory

Case 2: Multi-lane + Imbalanced Inflow & Outflow

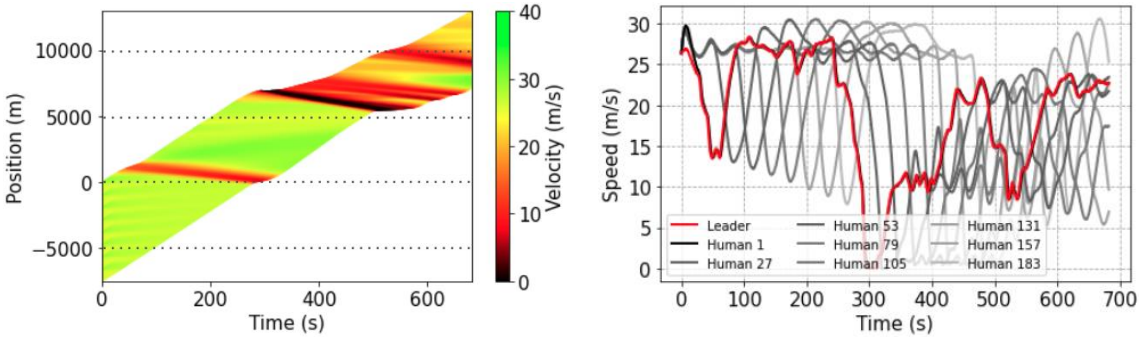


Screenshot of the simulation of I-210 in SUMO

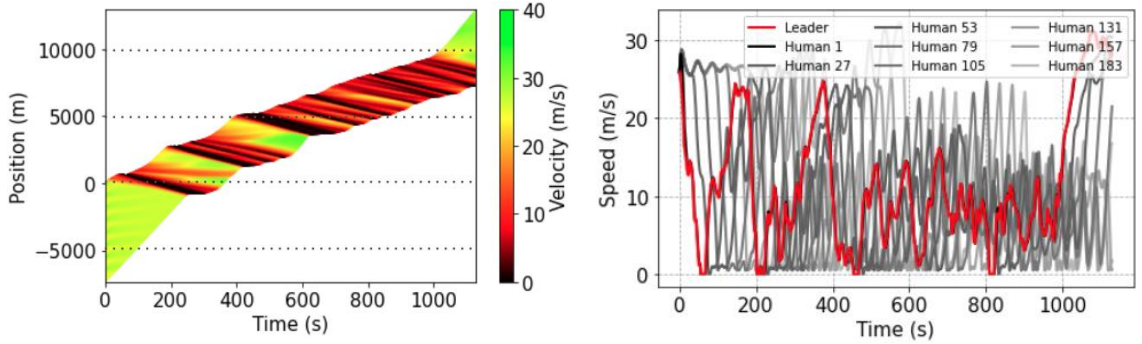
Case 1: Single Lane + Recorded Trajectories

We chose 10 different scenarios to simulate with different level of stop-and-go waves.

Light/Moderate Congestion (7 scenarios)



Heavy Congestion (3 scenarios)



Energy consumption (Miles-Per-Gallon) has been chosen as a proxy of the smoothness of the traffic flow.

Improvements compared with no control

Global sensing (expert):

- Light/Moderate: 14.21 %
- Heavy: 23.12%
- **Overall: 16.57%**

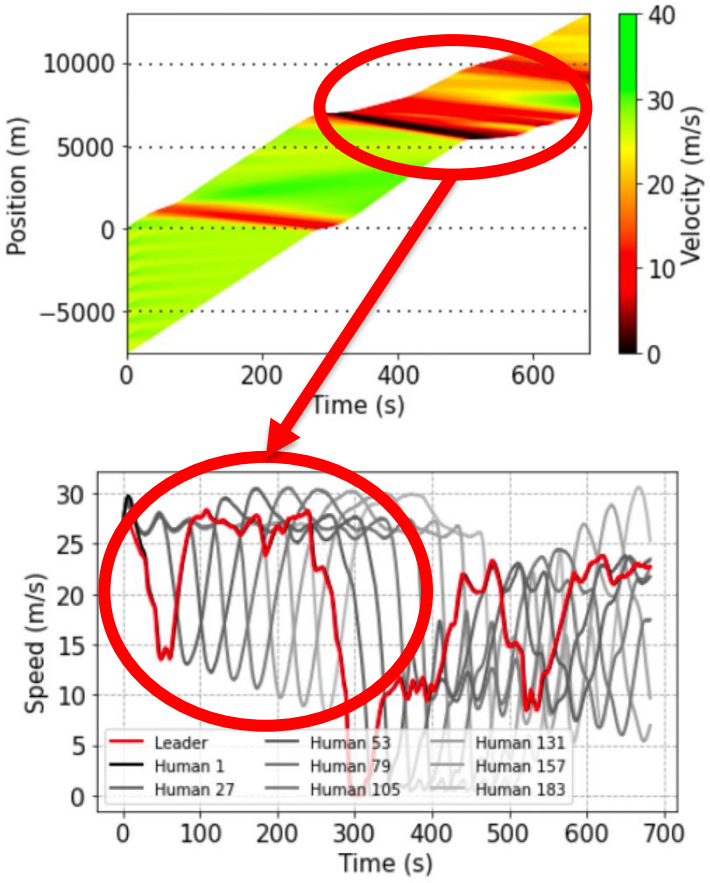
Local sensing (imitation learning):

- Light/Moderate: 13.83%
- Heavy: 16.37%
- **Overall: 14.50%**

Single-lane Simulations: I-24 freeway

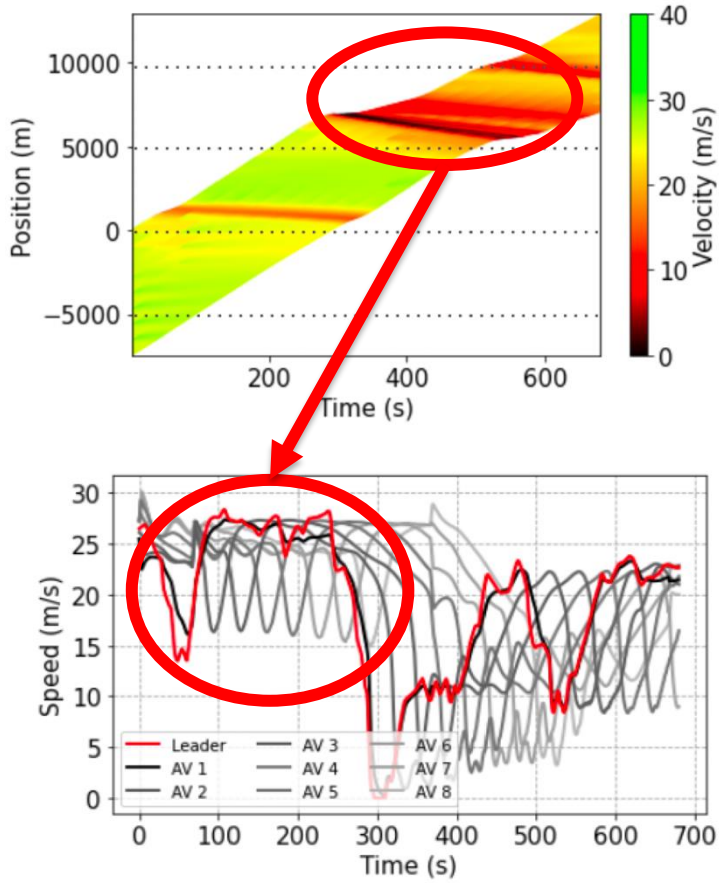
200 vehicles, 4% AVs

No Control



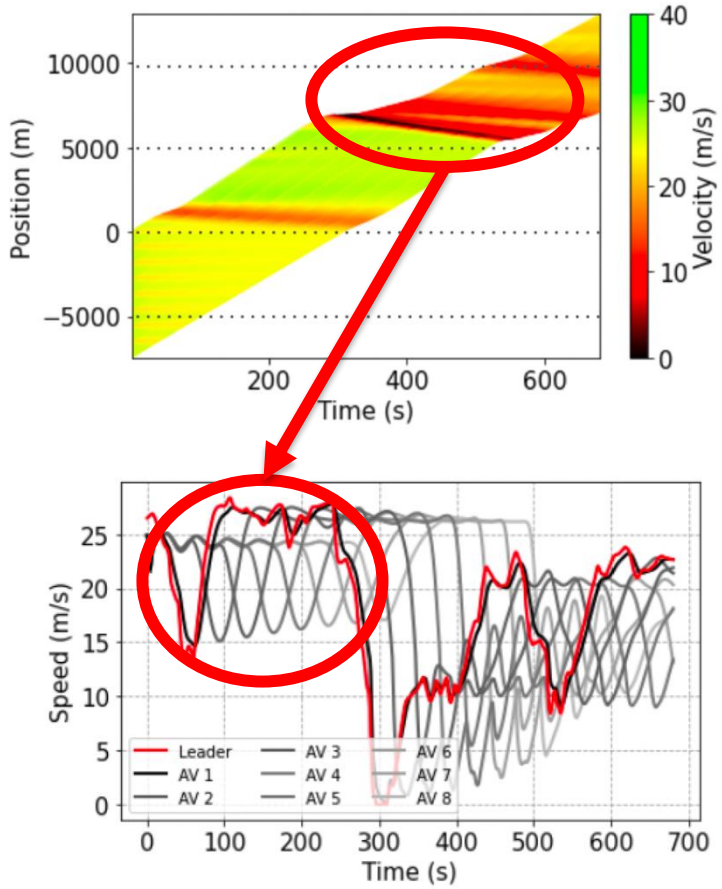
(a) Fully human-driven

Kernel-based Control



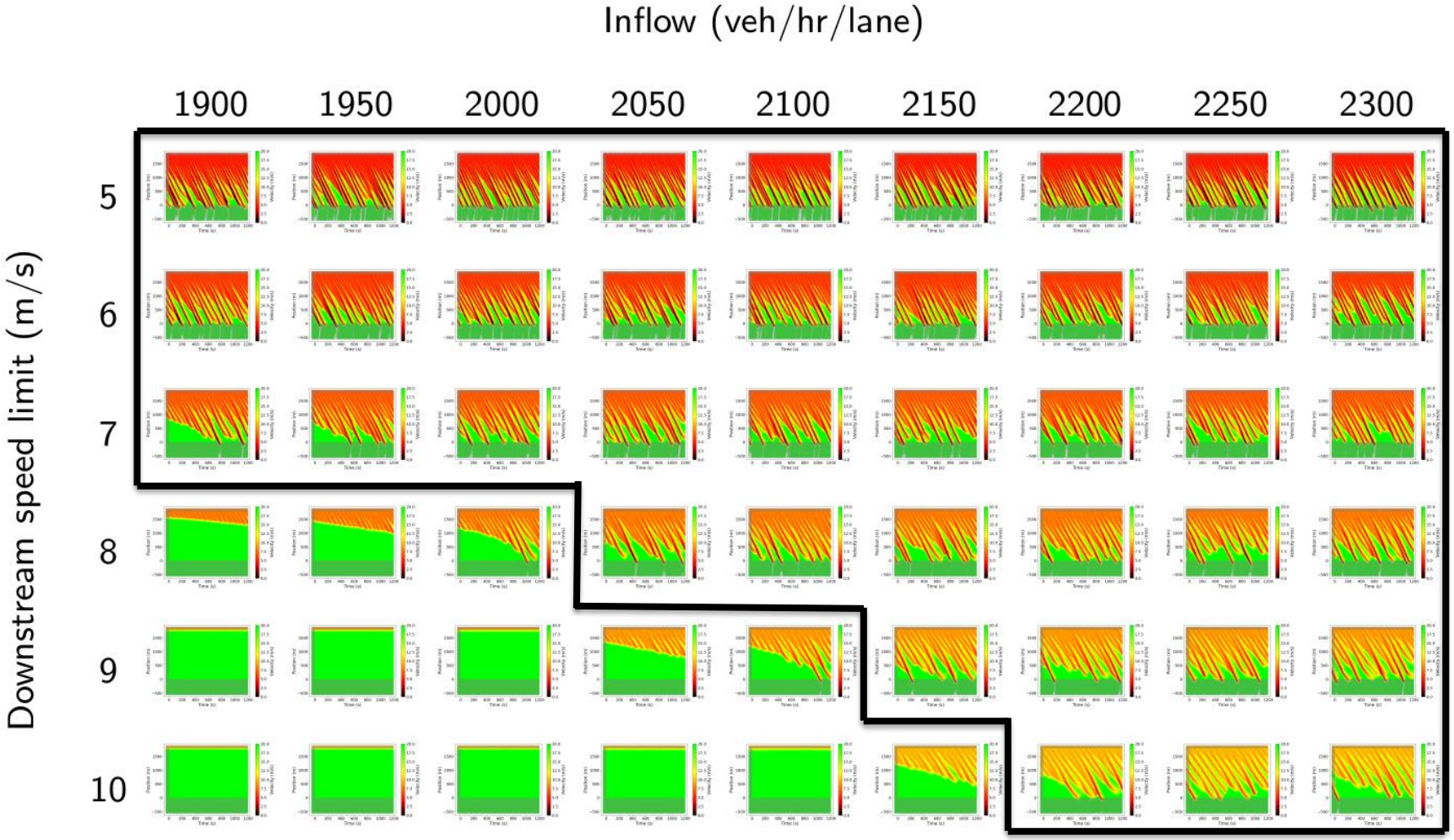
(b) Global sensing (expert)

Imitation Learning Control



(c) Local sensing (imitation)

Case 2: Multi-lane + Imbalanced Inflow & Outflow



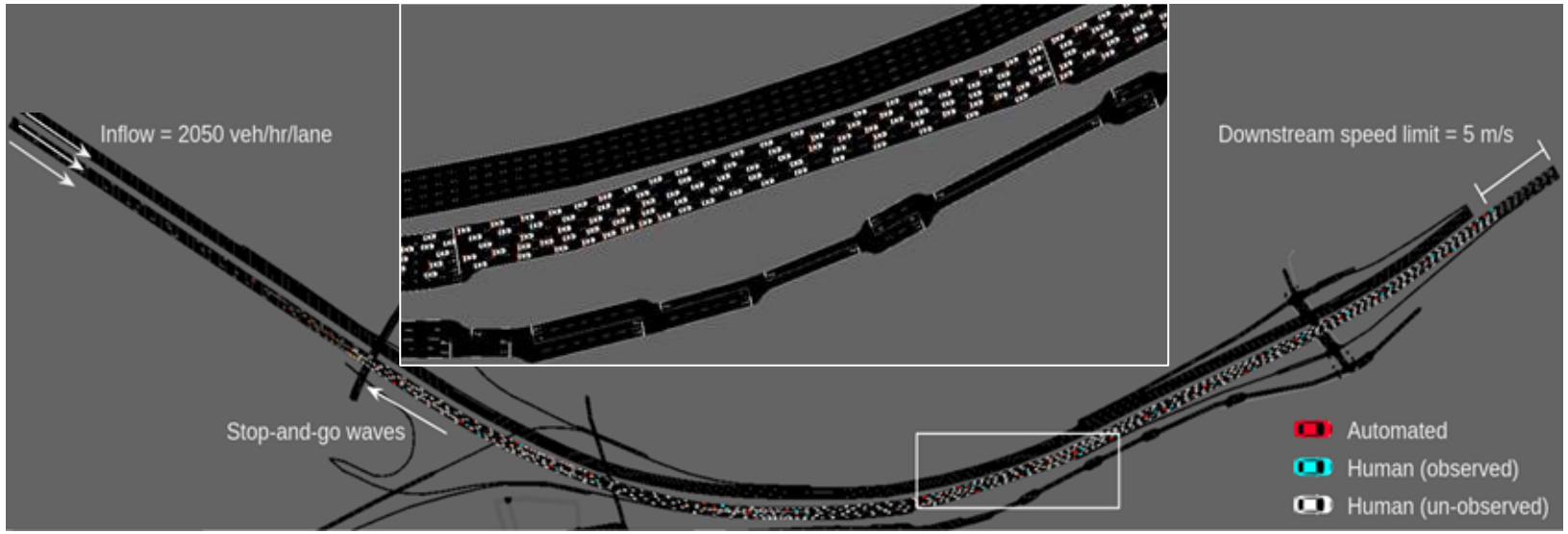
Time-Space Diagram Visualization for Training Sets

- Various inflow rates and downstream end speeds.
- Only domains with stop-and-go waves are considered in training.
- This allows the system to learn to match behaviors that dissipate waves as opposed to overfitting to a certain speed.

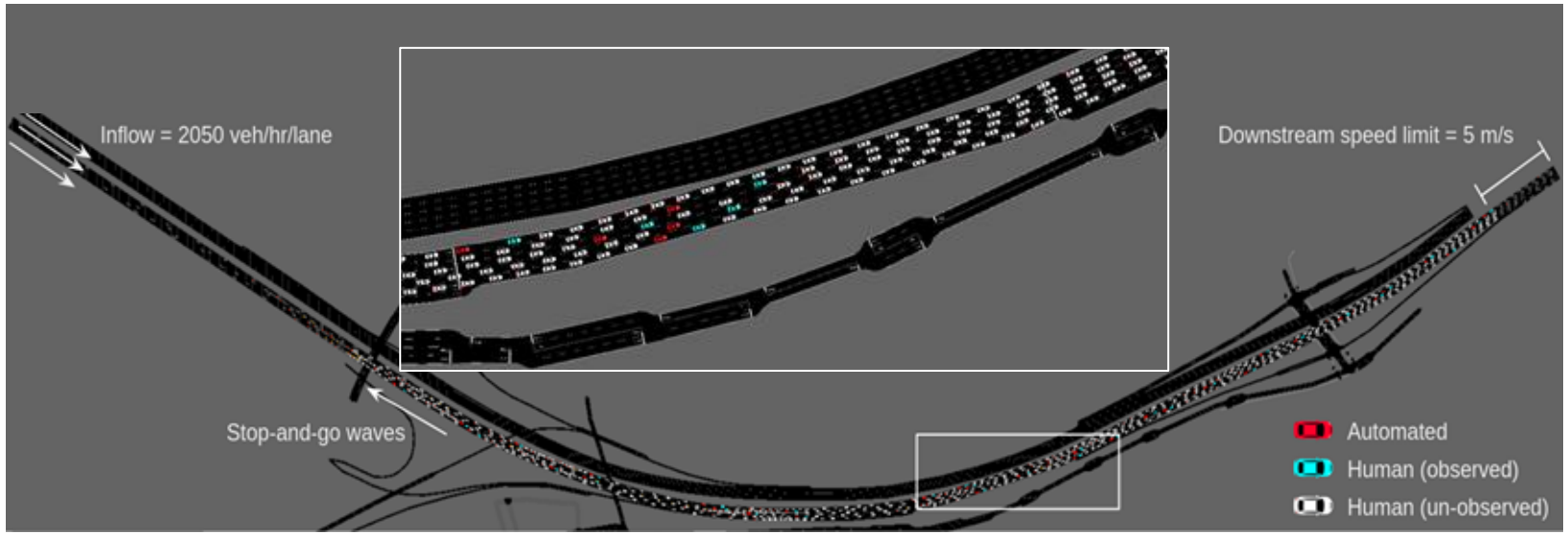
Implementation of algorithms on I-210 digital twin

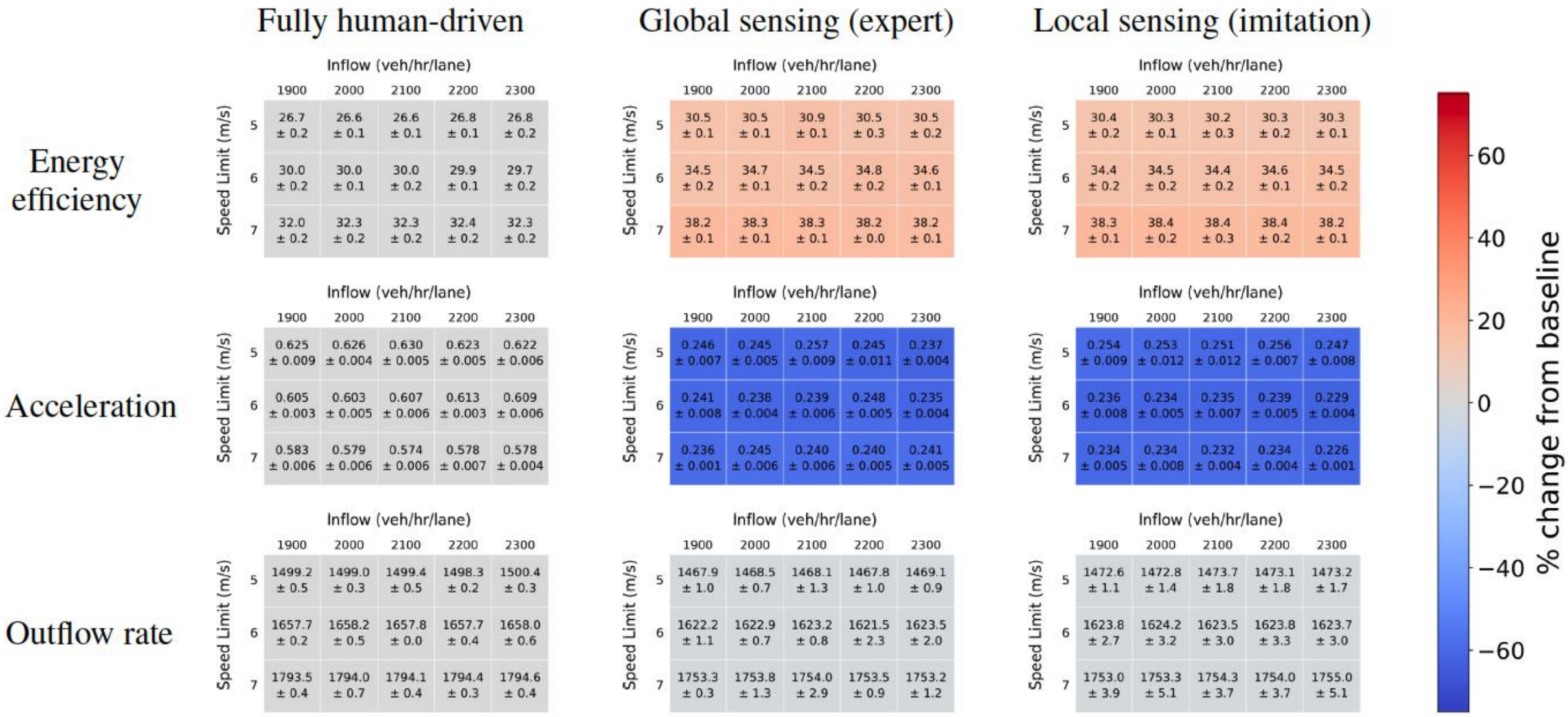
[1900, 2300] veh/hr/lane, **5%** AVs

No Control



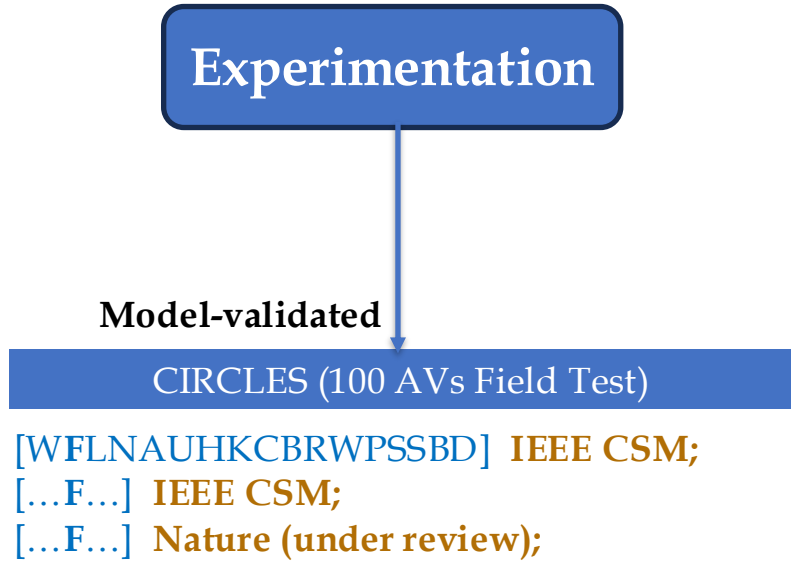
With Control




Fully human-driven
Global sensing (expert)
Local sensing (imitation)
Energy efficiency
Acceleration
Outflow rate
Inflow (veh/hr/lane)
Speed Limit (m/s)
1900 2000 2100 2200 2300
5 6 7
26.7 ± 0.2 | 26.6 ± 0.1 | 26.6 ± 0.1 | 26.8 ± 0.1 | 26.8 ± 0.2
30.0 ± 0.2 | 30.0 ± 0.1 | 30.0 ± 0.2 | 29.9 ± 0.1 | 29.7 ± 0.2
32.0 ± 0.2 | 32.3 ± 0.2 | 32.3 ± 0.2 | 32.4 ± 0.2 | 32.3 ± 0.2
30.5 ± 0.1 | 30.5 ± 0.1 | 30.9 ± 0.1 | 30.5 ± 0.3 | 30.5 ± 0.2
34.5 ± 0.2 | 34.7 ± 0.1 | 34.5 ± 0.2 | 34.8 ± 0.2 | 34.6 ± 0.1
38.2 ± 0.1 | 38.3 ± 0.1 | 38.3 ± 0.1 | 38.2 ± 0.0 | 38.2 ± 0.1
30.4 ± 0.2 | 30.3 ± 0.1 | 30.2 ± 0.3 | 30.3 ± 0.2 | 30.3 ± 0.1
34.4 ± 0.2 | 34.5 ± 0.2 | 34.4 ± 0.2 | 34.6 ± 0.1 | 34.5 ± 0.2
38.3 ± 0.1 | 38.4 ± 0.2 | 38.4 ± 0.3 | 38.4 ± 0.2 | 38.2 ± 0.1
0.625 ± 0.009 | 0.626 ± 0.004 | 0.630 ± 0.005 | 0.623 ± 0.005 | 0.622 ± 0.006
0.605 ± 0.003 | 0.603 ± 0.005 | 0.607 ± 0.006 | 0.613 ± 0.003 | 0.609 ± 0.006
0.583 ± 0.006 | 0.579 ± 0.006 | 0.574 ± 0.006 | 0.578 ± 0.007 | 0.578 ± 0.004
0.246 ± 0.007 | 0.245 ± 0.005 | 0.257 ± 0.009 | 0.245 ± 0.011 | 0.237 ± 0.004
0.241 ± 0.008 | 0.238 ± 0.004 | 0.239 ± 0.006 | 0.248 ± 0.005 | 0.235 ± 0.004
0.236 ± 0.001 | 0.245 ± 0.006 | 0.240 ± 0.006 | 0.240 ± 0.005 | 0.241 ± 0.005
0.254 ± 0.009 | 0.253 ± 0.012 | 0.251 ± 0.012 | 0.256 ± 0.007 | 0.247 ± 0.008
0.236 ± 0.008 | 0.234 ± 0.005 | 0.235 ± 0.007 | 0.239 ± 0.005 | 0.229 ± 0.004
0.234 ± 0.005 | 0.234 ± 0.008 | 0.232 ± 0.004 | 0.234 ± 0.004 | 0.226 ± 0.001
1499.2 ± 0.5 | 1499.0 ± 0.3 | 1499.4 ± 0.5 | 1498.3 ± 0.2 | 1500.4 ± 0.3
1657.7 ± 0.2 | 1658.2 ± 0.5 | 1657.8 ± 0.0 | 1657.7 ± 0.4 | 1658.0 ± 0.6
1793.5 ± 0.4 | 1794.0 ± 0.7 | 1794.1 ± 0.4 | 1794.4 ± 0.3 | 1794.6 ± 0.4
1467.9 ± 1.0 | 1468.5 ± 0.7 | 1468.1 ± 1.3 | 1467.8 ± 1.0 | 1469.1 ± 0.9
1622.2 ± 1.1 | 1622.9 ± 0.7 | 1623.2 ± 0.8 | 1621.5 ± 2.3 | 1623.5 ± 2.0
1753.3 ± 0.3 | 1753.8 ± 1.3 | 1754.0 ± 2.9 | 1753.5 ± 0.9 | 1753.2 ± 1.2
1472.6 ± 1.1 | 1472.8 ± 1.4 | 1473.7 ± 1.8 | 1473.1 ± 1.8 | 1473.2 ± 1.7
1623.8 ± 2.7 | 1624.2 ± 3.2 | 1623.5 ± 3.0 | 1623.8 ± 3.3 | 1623.7 ± 3.0
1753.0 ± 3.9 | 1753.3 ± 5.1 | 1754.3 ± 3.7 | 1754.0 ± 3.7 | 1755.0 ± 5.1
60
40
20
0
−20
−40
−60
% change from baseline

CIRCLES: The largest scientific traffic field experiment to date

Deploy 100 AVs on I-24 highway surrounded by human commuters during morning rush

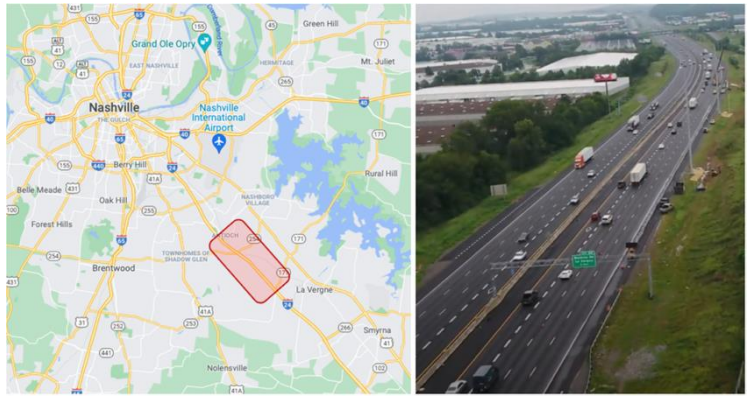


Source of Hardness	Communication delay; sim-to-real gap
Solution Method	**Model-validated** & feedback loop deployment

CIRCLES: The largest mixed traffic field experiment to date

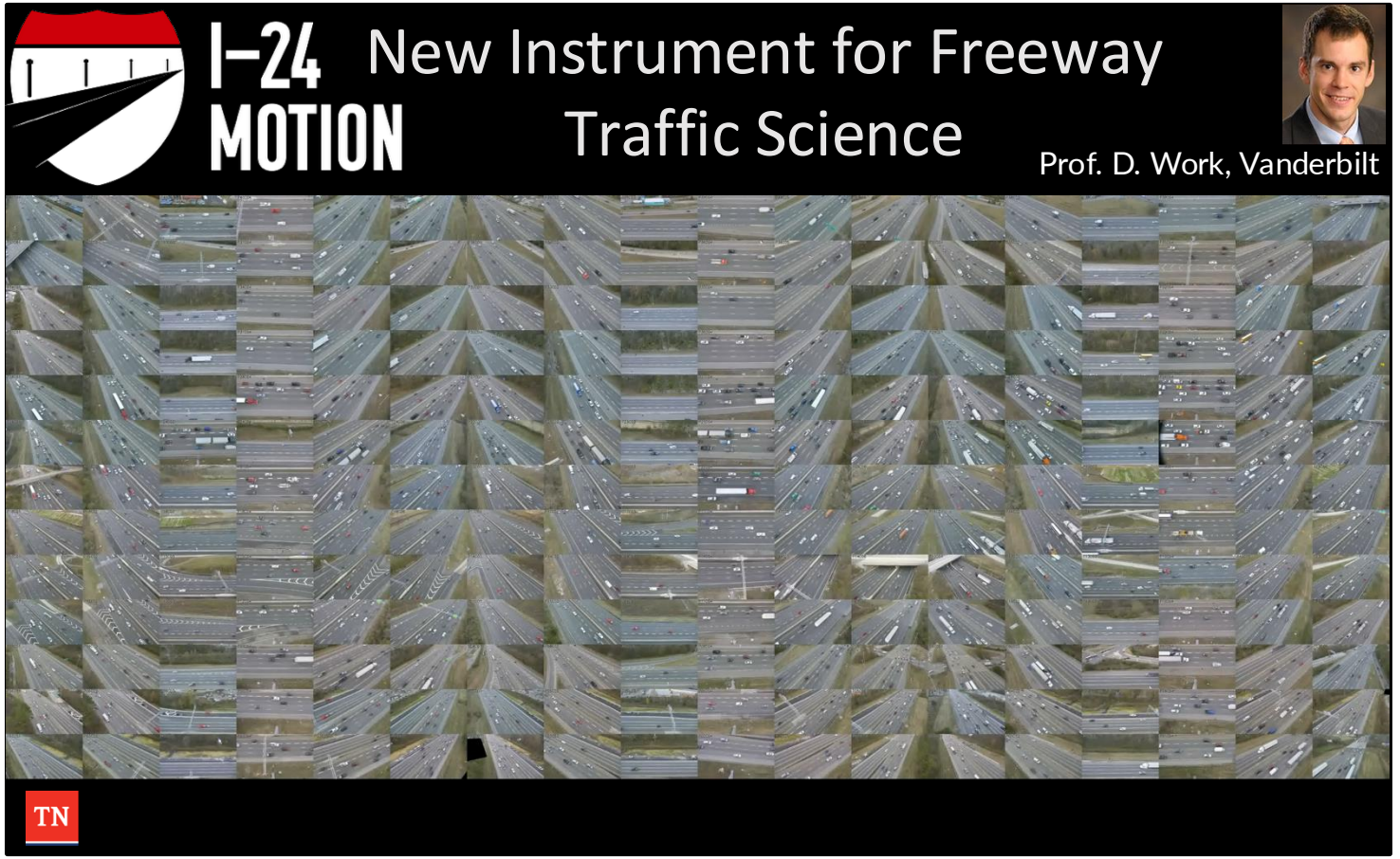
Deploy 100 AVs on I-24 highway surrounded by human commuters during morning rush

CIRCLES



I-24 Westbound in Nashville, Tennessee

Bird's-Eye View of 100 AVs in the Parking Lot at the HQ



- **4.2** miles; **276** 4K resolution cameras
- **230** million vehicle-miles of travel annually

A National-Scale Collaboration

National Science Foundation · U.S. Department of Energy · TN TDOT Department of Transportation · Toyota · Nissan Motor Corporation · GM Research & Development

With 4 universities;
Got 30+ Media outlets

During the 100 AVs Test

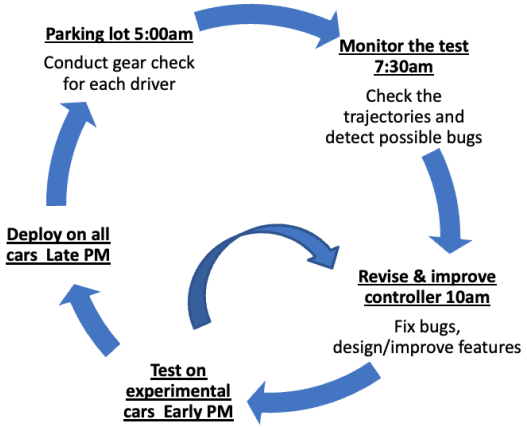
Team efforts & coordination (algorithms refine; hardware deployment; management, etc.)

Installing/uninstalling hardware on the 100 cars

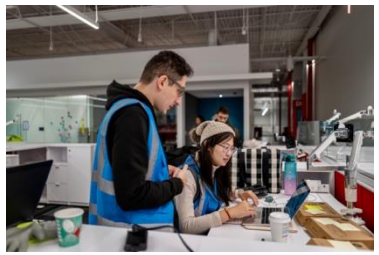
Fleet management at 5am at HQ parkinglot

My Contributions

- Design the core control algorithms
- Support the energy modeling and evaluation
- Deploy the controller on hardware & cars
- Support the experiment logistics



Gear Check 5am



Monitor Test & Detect Bugs 7:30am

Refine & Improve Controllers 10am

Test Controllers Early PM

Refine & Redeploy & Retest Early PM

Deploy Late PM

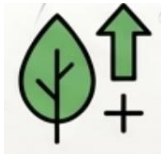
Empirical Proof: AVs Successfully Mitigated Stop-and-Go Waves

100

Partially automated vehicles

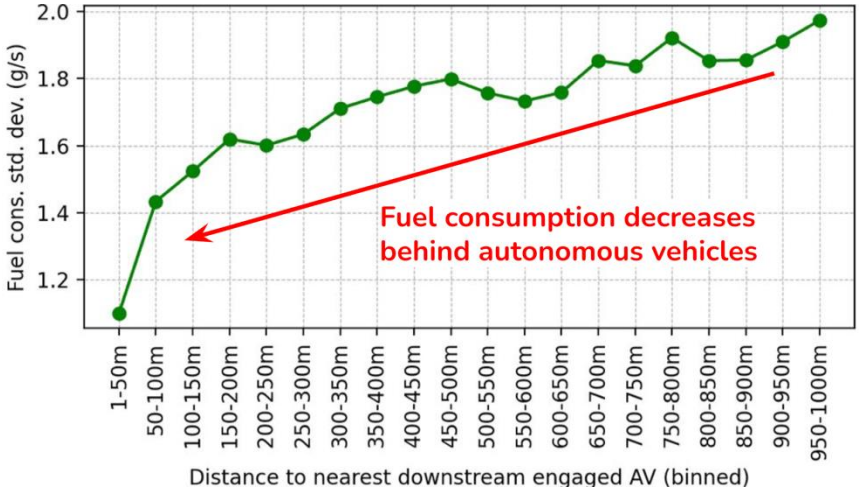
< 2 %

AV penetration

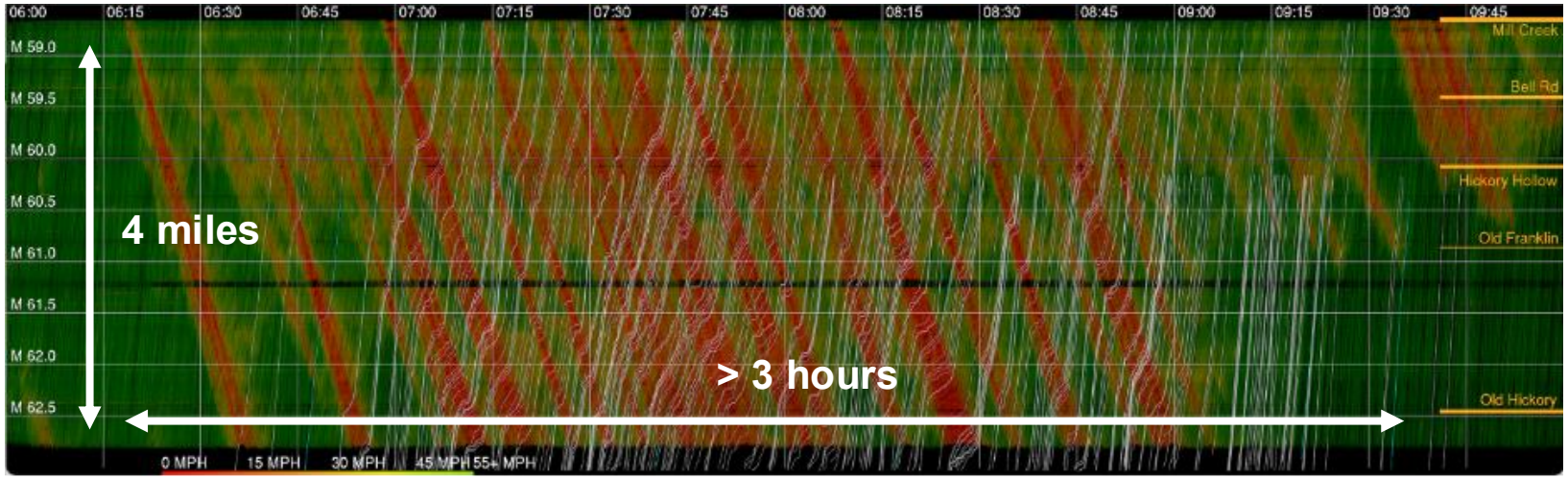


> 10 %

Measured system-wide energy savings



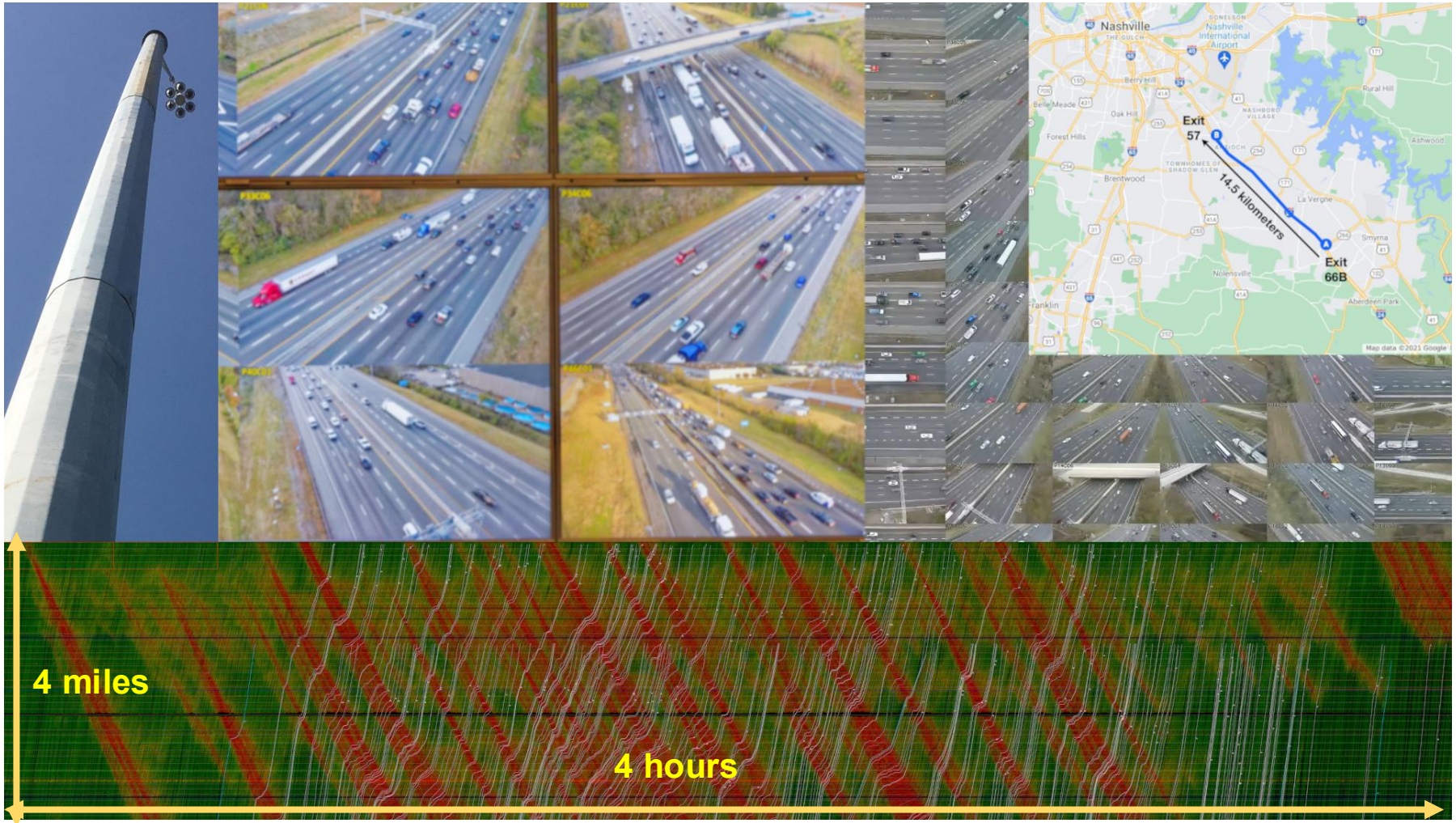
The nearer the human-driven vehicles to the AV are, the more fuel efficiency they could gain.



[The largest Time-space diagram credit to I-24 MOTION team (Gergely Zachár, Derek Gloudemans, Yanbing Wang, Junyi Ji, Matt Nice, Matt Bunting, Will Barbour, Jonathan Sprinkle, Dan Work) at Vanderbilt U.]

Close the Loop: Data-Driven Refinement of Modeling

NFVM applies to large-scale experimental traffic data with complex flow dynamics



I-24 MOTION dataset

- ~ 4 miles I-24 highway
- 10 days of data
- > 40 hrs morning rush hours
- >200 GB json file
- includes ~600,000 vehicle trajectories

I-24 MOTION illustration. High-definition camera poles are mounted along a portion of I-24 at regular intervals.

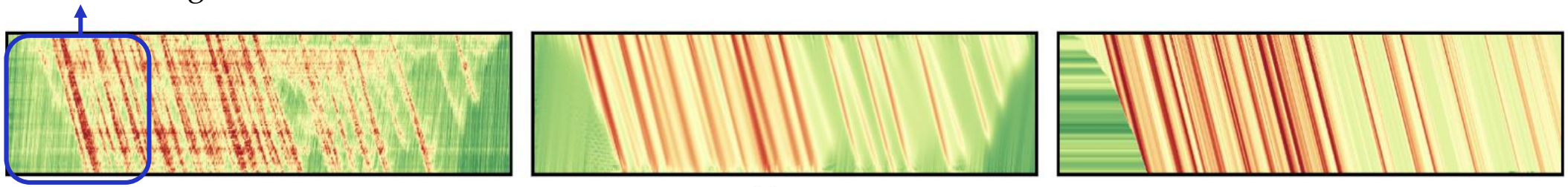
Derek Gloudeman et al. "I-24 motion: An instrument for freeway traffic science". In: Transportation Research Part C: Emerging Technologies, 155:104311, 2023

Generalizes Across 10 Days of Traffic

Trained on 1 hr of 1 day, NFVM outperforms Godunov across all data

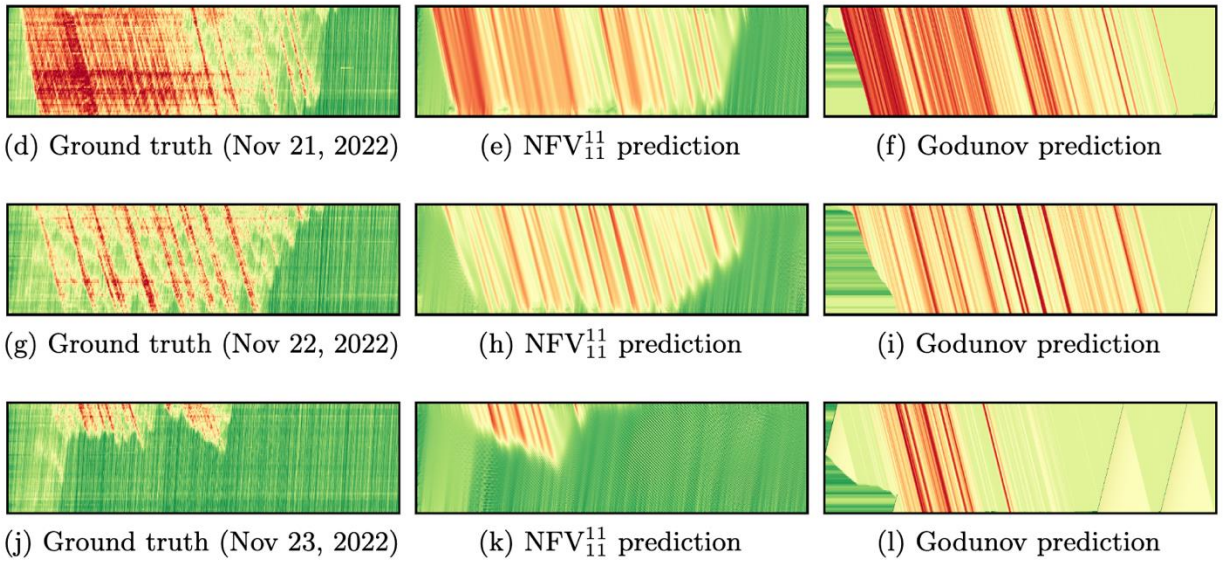
Only 1st hr data (2.5% of the dataset) is used as training set

Prediction on training day



(a) Ground truth (Nov 29, 2022) (b) NFV_{11}^{11} prediction (c) Godunov prediction

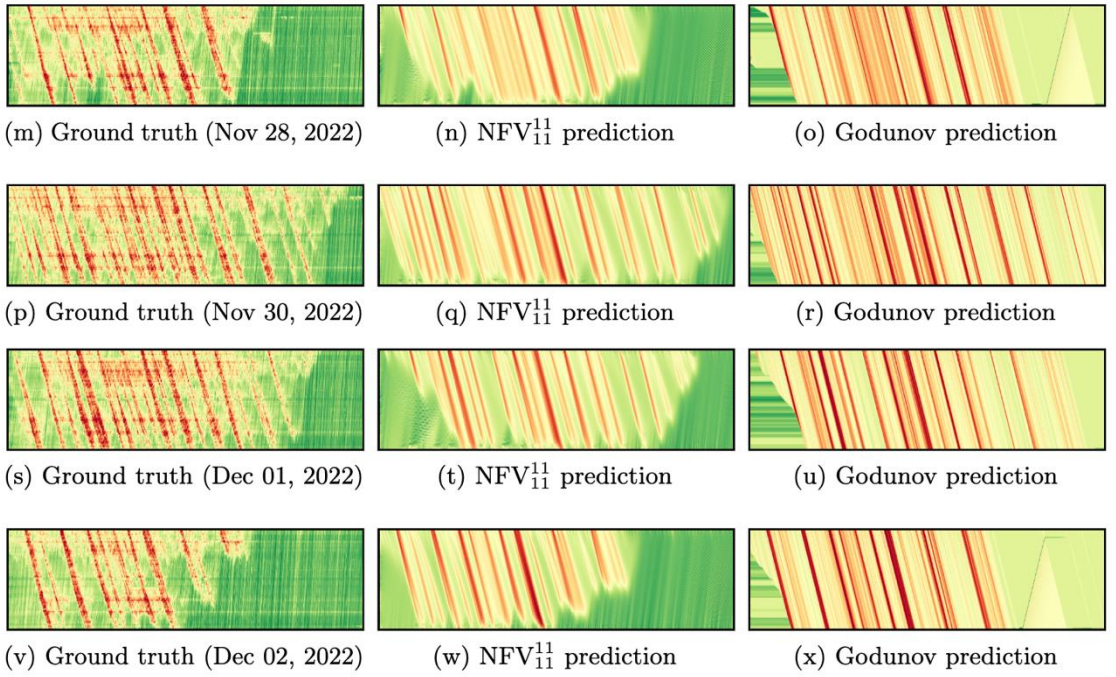
Generalization on unseen days



(d) Ground truth (Nov 21, 2022) (e) NFV_{11}^{11} prediction (f) Godunov prediction

(g) Ground truth (Nov 22, 2022) (h) NFV_{11}^{11} prediction (i) Godunov prediction

(j) Ground truth (Nov 23, 2022) (k) NFV_{11}^{11} prediction (l) Godunov prediction



(m) Ground truth (Nov 28, 2022) (n) NFV_{11}^{11} prediction (o) Godunov prediction

(p) Ground truth (Nov 30, 2022) (q) NFV_{11}^{11} prediction (r) Godunov prediction

(s) Ground truth (Dec 01, 2022) (t) NFV_{11}^{11} prediction (u) Godunov prediction

(v) Ground truth (Dec 02, 2022) (w) NFV_{11}^{11} prediction (x) Godunov prediction

Thanks to my main collaborators:


Arwa Alanqary


Zakaria Baba


Alexi Canesse


Xiaoyang Cao


Joy Carpio


Martin Drieux


Nicole Han


Aboudy Kreidieh


Nathan Lichtlé


Zihe Liu


Hossein Matin


Han Wang


Alexandre Bayen
Principal Investigator & Professor, UC Berkeley


Maria Laura Delle Monache
Co-Principal Investigator & Professor, UC Berkeley


Jonathan Lee
Chief Engineer & Co-Principal Investigator, UC Berkeley


Benedetto Piccoli
Co-Principal Investigator & Professor, Rutgers University


Benjamin Seibold
Co-Principal Investigator & Professor, Temple University


Jonathan Sprinkle
Co-Principal Investigator & Professor, Vanderbilt University


Daniel Work
Co-Principal Investigator & Professor, Vanderbilt University

Acknowledgement