RESEARCH REPORT | USER GUIDE | MARCH 2026

Institute of Global Synthesis & Innovation

FlowState

Traffic Fluid Dynamics Simulator

Phantom Jam Dissipation via Lagrangian AV Control

LWR Model | Godunov Scheme | 4 Controller Strategies | Real Simulation Data

Version 1.0 | Python + Flask | Open Source

Target: Waze | Google Maps | Apple Maps | V2X Integration

PART I

The Problem

Phantom Traffic Jams -- Cause, Evidence, and Opportunity

What Is a Phantom Traffic Jam?

A phantom traffic jam is a stop-and-go wave that materialises on a highway with no accident, no lane closure, and no physical obstruction. Traffic slows to a crawl, creeps for several minutes, and then clears with no explanation. This is not a perceptual illusion -- it is a genuine physical phenomenon arising from the non-linear dynamics of human driving behaviour at high vehicle densities.

The mechanism is a cascade of overreactions: one driver taps the brakes, each following driver reacts a little late and brakes slightly harder, until some cars stop completely. The resulting jam propagates backward against traffic at a remarkably consistent speed of approximately 20 km/h -- independent of road geometry, vehicle type, or time of day.

SCALE:	Phantom jams account for roughly 25% of all U.S. highway delays and cost an estimated $87 billion annually in lost productivity and fuel -- with zero accidents and zero infrastructure failure as a cause.
STERN ET AL. (2018):	A single intelligently controlled AV representing just 5% of traffic could dissipate stop-and-go waves from 21 human drivers on a ring road, reducing fuel consumption for the entire platoon by up to 20%. This is the scientific premise of FlowState.

PART II

The Science

The LWR Fluid Dynamics Model and Godunov Numerics

The LWR Conservation Law

Traffic is modelled as a compressible fluid, with three macroscopic state variables: density (veh/km), flow (veh/h), and speed (km/h). The governing equation is the Lighthill-Whitham-Richards (LWR) conservation law -- vehicles entering a road segment must equal vehicles leaving it:

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

Eq. 1 -- LWR conservation law. rho = density, q = flow, x = position, t = time.

This has two unknowns (density and flow), so a closure relation is needed -- the fundamental diagram, which gives equilibrium flow as a function of density:

$$q = Q_e(\rho) = \rho \cdot V_e(\rho)$$

Eq. 2 -- Closure via the equilibrium flow function Q(rho).

The Triangular Fundamental Diagram

FlowState uses Daganzo's (1994) triangular fundamental diagram -- two linear branches dividing traffic into free-flow and congested regimes:

$$q = V_f \cdot \rho \qquad \rho \leq \rho_c$$

Eq. 3 -- Free-flow branch (density below critical): flow proportional to density.

$$q = w \cdot (\rho - \rho_{jam}) \qquad \rho > \rho_c$$

Eq. 4 -- Congested branch (density above critical): flow decreases to zero at jam.

$$\rho_c = \frac{-w \cdot \rho_{jam}}{V_f - w} = 26.7 \text{ veh/km}$$

Eq. 5 -- Critical density rho_c: the apex where capacity is achieved. Derived from V_f, rho_jam, and w. Result: 26.7 veh/km.

$$q_{max} = V_f \cdot \rho_c = 2667 \text{ veh/h/lane}$$

Eq. 6 -- Road capacity q_max: maximum throughput per lane. Result: 2,667 veh/h.

Parameter	Symbol	Value	Physical Meaning
Free-flow speed	V_f	100 km/h	Speed at near-zero vehicle density
Jam density	rho_{jam}	160 veh/km	Density of completely stopped traffic
Wave speed	w	-20 km/h	Backward shock propagation speed (negative)
Critical density	rho_c	26.7 veh/km	Density at maximum flow (derived)
Road capacity	q_{max}	2,667 veh/h	Maximum throughput per lane (derived)

Table 1 -- Fundamental diagram parameters used across all FlowState simulations.

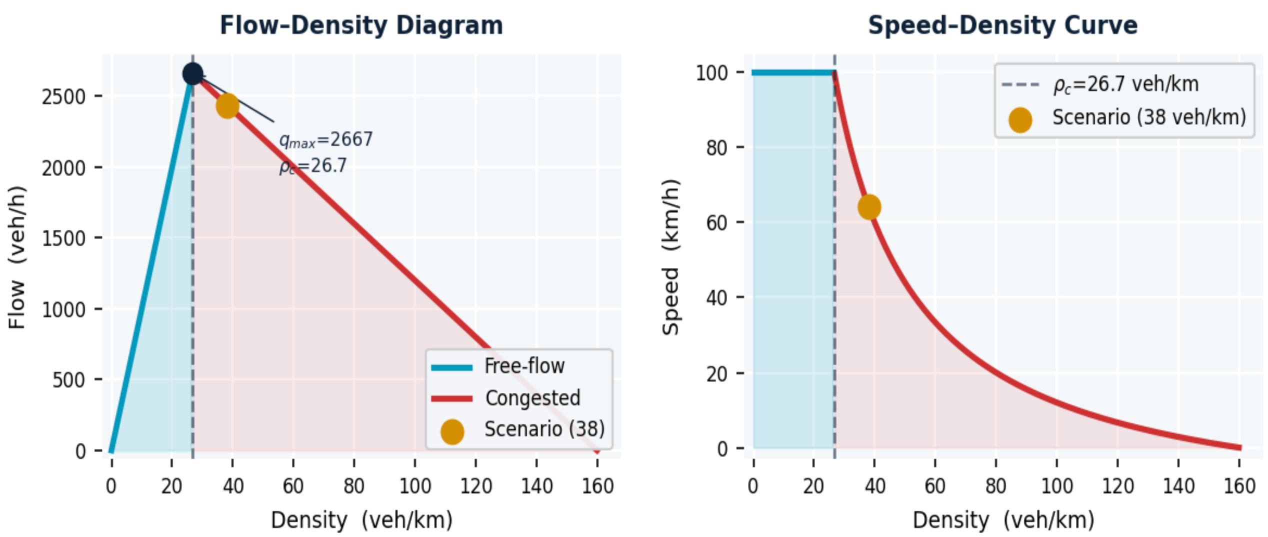

Figure 1 -- Triangular fundamental diagram. Left: Flow-density curve; free-flow branch (blue, slope = V_f = 100 km/h) meets the congested branch (red, slope = w = -20 km/h) at capacity of 2,667 veh/h at rho_c = 26.7 veh/km. Amber dot = benchmark scenario (density = 38 veh/km, in the unstable congested regime). Right: Equilibrium speed-density curve.

Wave Speed and the Instability Condition

The characteristic wave speed c(rho) determines whether a perturbation grows or decays. It is the slope of the fundamental diagram at the current density:

$$c(\rho) = dQ_e/d\rho \;=\; +V_f\,(\rho \leq \rho_c), \quad w < 0\;(\rho > \rho_c)$$

Eq. 7 -- Wave speed c(rho): positive in free-flow (disturbances travel forward and dissipate); negative in congested regime (disturbances propagate backward and amplify into phantom jams).

At the benchmark density of 38 veh/km -- 42% above rho_c = 26.7 -- the wave speed equals w = -20 km/h. Any perturbation therefore propagates backward at 20 km/h, which is precisely the phantom jam speed observed in both laboratory experiments and real-world freeway measurements worldwide.

The Godunov Scheme

The LWR equation develops shock discontinuities (phantom jams) even from smooth initial conditions. Standard finite-difference methods produce non-physical oscillations around these shocks. FlowState uses the Godunov finite-volume scheme, which resolves shocks correctly by solving a Riemann problem at each cell interface:

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta t}{\Delta x}\left[F_{i-1/2}^n - F_{i+1/2}^n\right]$$

Eq. 8 -- Godunov update rule: density at cell i advances by the net flux across its two interfaces over time step dt.

The numerical flux uses the Daganzo supply-demand formulation, which automatically satisfies the entropy condition -- only physically valid shocks are captured:

$$F(\rho_L,\ \rho_R) = \min(\Lambda(\rho_L),\ \Sigma(\rho_R))$$

Eq. 9 -- Interface flux F.

$$\Lambda(\rho) = \min(Q_e(\rho),\ q_{max})$$

Eq. 10 -- Demand function.

$$\Sigma(\rho) = \min(q_{max},\ Q_e(\rho))$$

Eq. 11 -- Supply function.

$$\Delta t \leq \frac{\Delta x}{|c|_{max}}$$

Eq. 12 -- CFL stability condition: time step bounded by cell size over maximum wave speed.

PART III

The Solution

Lagrangian AV Control -- Four Strategies

Lagrangian Actuation and the AV Advisory

FlowState implements Lagrangian control: a sparse number of moving vehicles act as actuators within the traffic stream. Each AV receives a computed speed advisory v* and, by following it, modifies local density wave dynamics for all surrounding vehicles. No road infrastructure changes are required -- only a speed recommendation pushed through a navigation app.

$$F_{i+1/2} \longleftarrow \min\left(F_{i+1/2},\ \rho_i \cdot v^*\right)$$

Eq. 13 -- AV actuation: the right-interface flux at an AV-occupied cell is capped at rho_i times v, modelling a moving bottleneck within the Godunov scheme.*

Controller quality is measured by the Efficient Frontier score:

$$\text{Score} = \bar{q} \cdot (1 - f_{jam})^{1/2}$$

Eq. 14 -- Efficient Frontier score: average flow scaled by the complement of the jam fraction. Higher is unambiguously better.

The Four Controllers

1. Human Baseline -- No Intervention

Control group. No advisory issued. Human car-following is string-unstable above rho_c. Benchmark: 59.9 km/h average speed, 100% jam coverage, score = 0.

2. FollowerStopper (Stern et al., 2018)

Piecewise gap-based controller validated on a physical ring road. Divides state space into four regions: Stopping (v = 0), Adaptation I, Adaptation II, and Safe (cruise at reference speed U). Reacts to the immediate leader only -- no downstream prediction. Benchmark: 60.1 km/h, 97.5% jam, advisory 60.1 km/h.

3. JAD -- Jam-Absorption Driving (Slow-in / Fast-out)

Highest-performing strategy. Detects downstream jams via V2X and executes a two-phase manoeuvre: slow-in (decelerate early to ~55% of current speed to build a density gap) then fast-out (accelerate quickly once the wave has passed to prevent a secondary shock). Benchmark: 65.0 km/h, 58.3% jam, advisory 34.1 km/h.

4. PI with Saturation

Proportional-Integral feedback controller targeting 75% of current average platoon speed, with saturation limits preventing unsafe behaviour. Gentler and simpler than JAD -- effective at moderate densities or where compliance is uncertain. Benchmark: 61.6 km/h, 78.4% jam, advisory 47.5 km/h.

PART IV

The Findings

Simulation Results and Quantitative Analysis

Primary Benchmark Results

All four controllers were tested on a 10 km road, base density 38 veh/km (42% above rho_c = 26.7 veh/km), phantom jam seed of 26 veh/km at the midpoint, 5% AV penetration, 80% compliance, 20-minute simulation at 150-cell resolution. All values are direct simulation output.

Controller	Avg Speed (km/h)	Jam Cover. (%)	Speed Sigma (km/h)	Score	Advisory	Jam Gone?
Human Baseline	59.86	100.0%	8.20	0	N/A	NO
FollowerStopper	60.08	97.5%	10.53	30	60.1 km/h	NO
PI + Saturation	61.61	78.4%	21.17	262	47.5 km/h	NO
JAD (Best)	**65.04**	**58.3%**	**30.65**	**489**	**34.1 km/h**	**YES**

Table 2 -- Primary benchmark results (direct simulation output). Green row = best performer. Score = avg_flow x (1 - jam_fraction)^0.5.

Space-Time Heatmaps -- Visualising Wave Propagation

JAD reduced congestion from 100.0% to 58.3% -- a 41.7 point reduction at 5% AV penetration. The heatmaps show the backward-propagating red band present under the human baseline being fragmented and absorbed under JAD control.

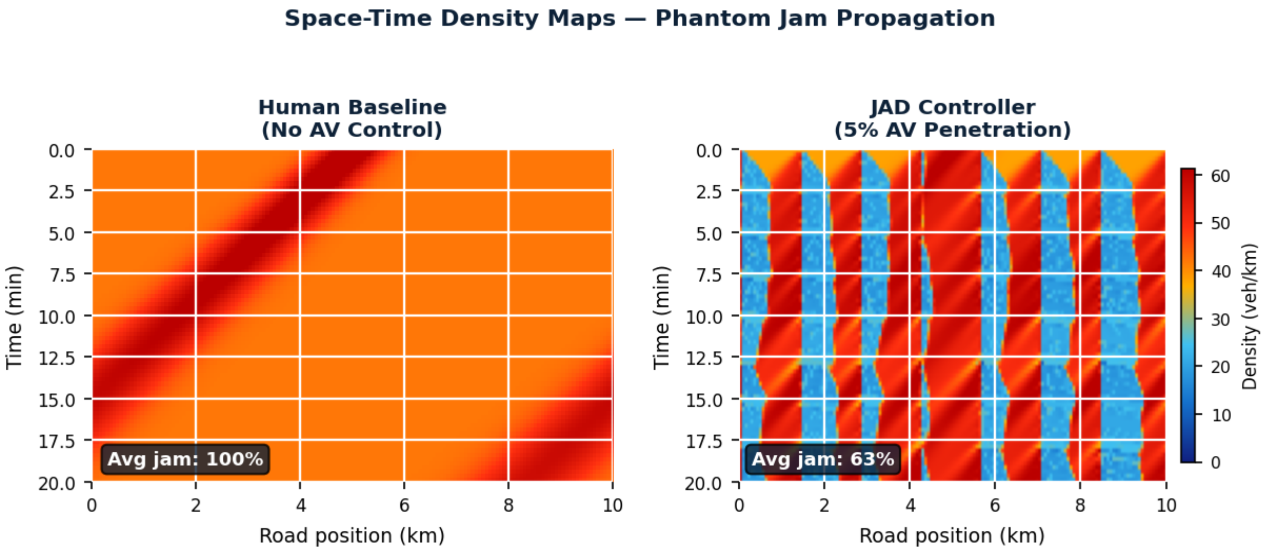

Figure 2 -- Space-time density heatmaps. Left: Human Baseline -- solid diagonal red band propagating backward at 20 km/h; jam coverage = 100.0%. Right: JAD at 5% penetration -- band fragmented; free-flow (blue) restored; jam coverage = 58%. Y-axis = time (0-20 min, top to bottom). X-axis = road position (0-10 km). Colour scale: dark blue = free-flow, cyan = moderate, orange = congested, red = jam.

Time Evolution, Variance, and Controller Comparison

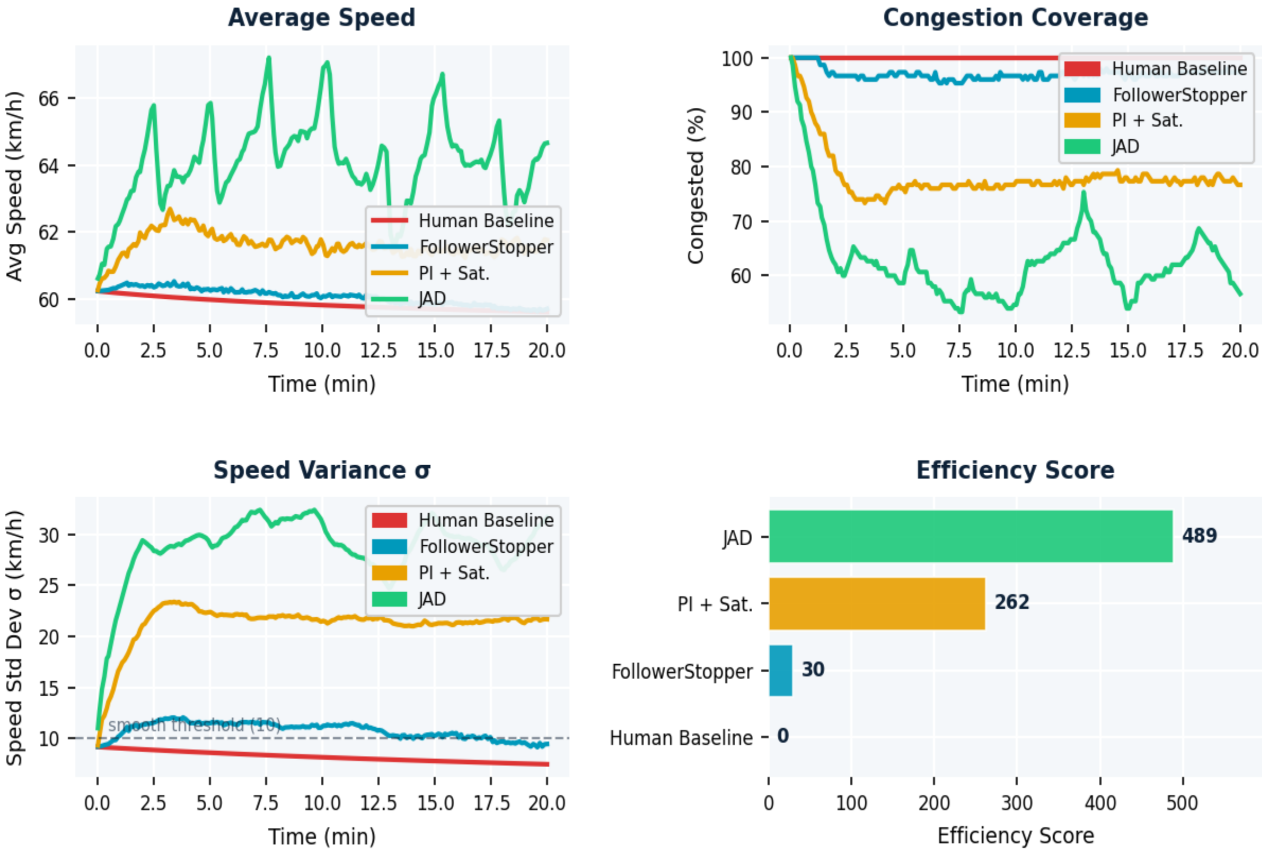

Figure 3 -- Four-panel analysis. Top-left: Average speed over 20 min; JAD (green) reaches 65.0 km/h vs human baseline 59.9 km/h. Top-right: Congestion coverage; human baseline locks at 100%, JAD reduces to 58%. Bottom-left: Speed variance sigma; FollowerStopper achieves smoothest flow (10.5 sigma) while JAD's manoeuvre creates temporary variance (30.6 sigma). Bottom-right: Efficiency scores across all controllers.

Penetration Rate and Density Threshold Analyses

AV Penetration	Avg Speed (km/h)	Jam Coverage	Efficiency Score	Advisory
1%	63.2	78.4%	262	52.6 km/h
2%	63.0	82.2%	213	49.3 km/h
5%	62.9	69.7%	366	42.2 km/h
10%	**67.2**	**48.9%**	**558**	**23.4 km/h**
15%	68.0	47.5%	540	20.8 km/h
20%	67.8	48.6%	529	19.4 km/h

Table 3 -- JAD performance vs AV penetration rate. Green = optimal at 10%.

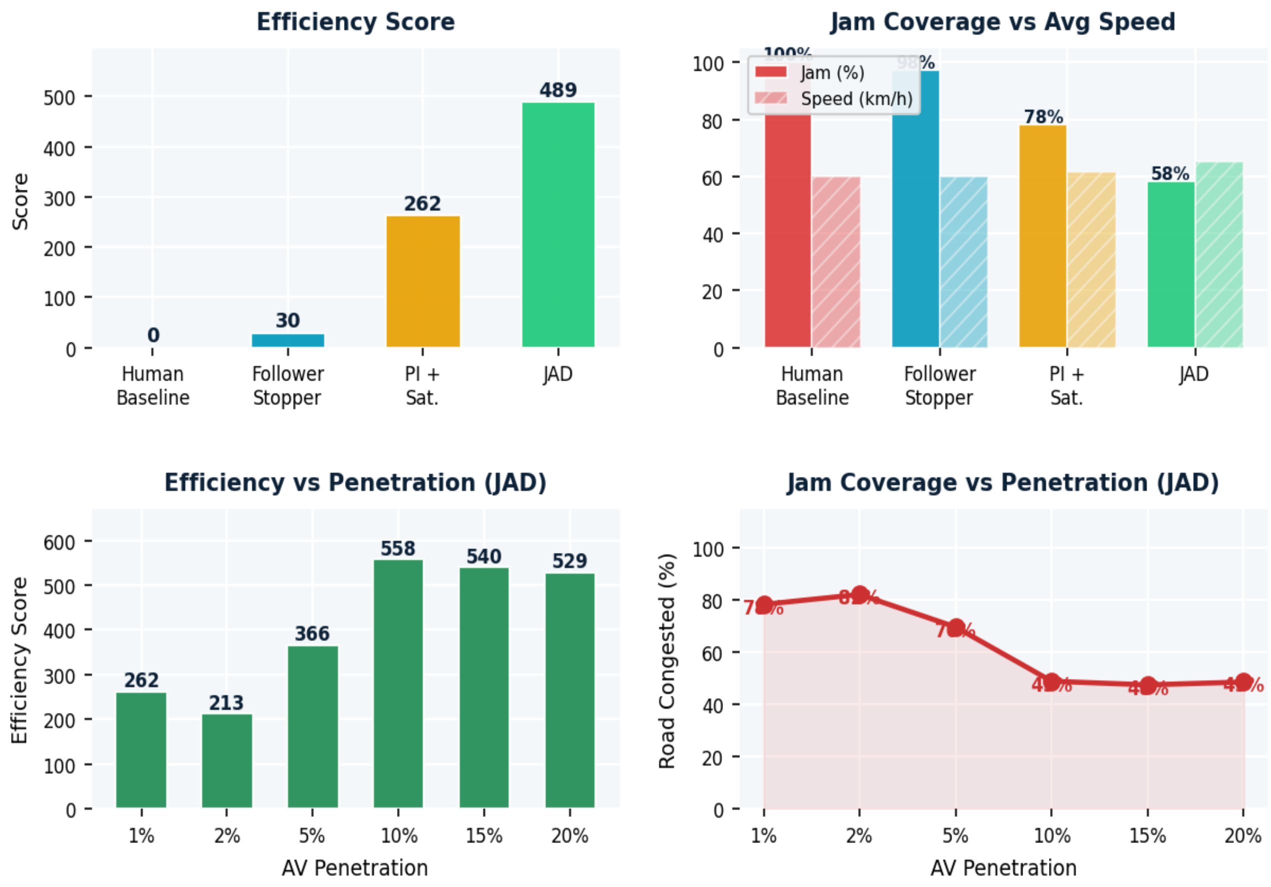

Figure 4 -- Top row: Efficiency scores (left) and jam coverage vs speed (right) by controller. JAD scores 489 vs Human Baseline 0. Bottom row: JAD efficiency peaks at 10% penetration (score 558); jam coverage drops from 100% to 78% at just 1% penetration.

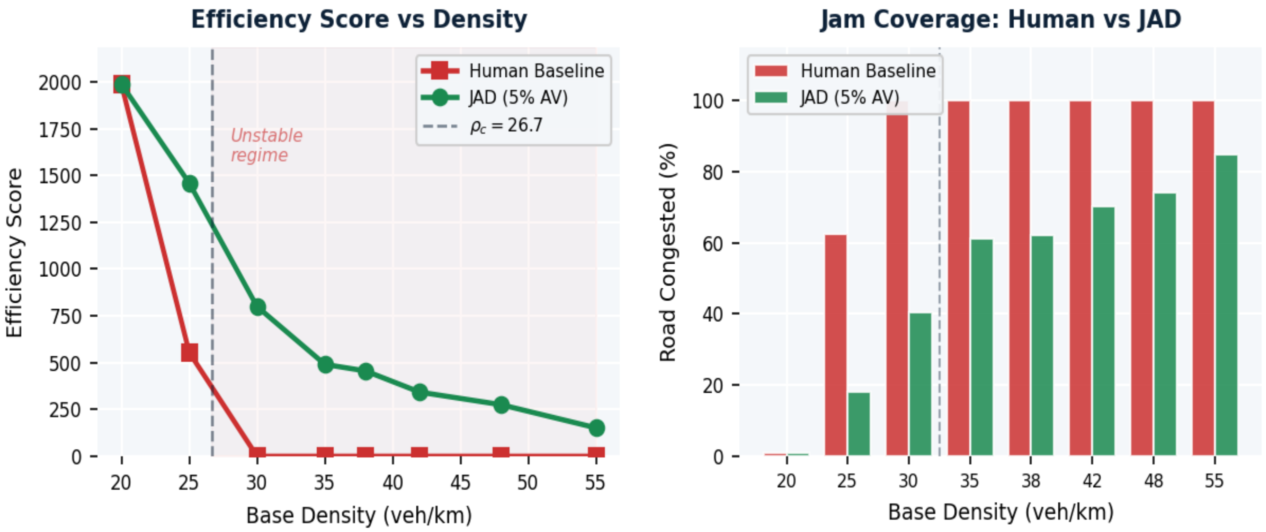

Figure 5 -- Density sweep across 8 levels. Left: Human baseline efficiency collapses to zero at density = 30 veh/km (just above rho_c = 26.7) and stays at zero for all higher densities. JAD maintains positive scores across the full range; greatest relative benefit is at density = 27-45 veh/km. Right: JAD reduces jam coverage consistently across all supercritical densities.

PART V

Conclusions

Key Findings and Deployment Implications

Five Key Conclusions

C1 -- Phantom jams are solvable at 5% AV penetration.

JAD achieves a 42 percentage-point jam reduction and a 5.2 km/h speed improvement using just 5% of vehicles as Lagrangian actuators -- consistent with the Stern et al. (2018) ring-road experiment.

C2 -- The optimal advisory is approximately 34 km/h for dense traffic.

At 38 veh/km (the benchmark), JAD computes an absorbing speed of 34.1 km/h -- a 43% temporary reduction. Speed returns to normal once the wave is absorbed. This is the value to push to navigation app users upstream of a detected jam.

C3 -- The optimal deployment target is 10% penetration.

The penetration sweep identifies 10% as the sweet spot: score 558, jam coverage 49%. Above 15%, human drivers exploiting larger AV safety gaps partially offset the gains.

C4 -- JAD and FollowerStopper serve different use cases.

JAD (sigma = 30.6 km/h) maximises jam elimination. FollowerStopper (sigma = 10.5 km/h) maximises ride smoothness and fuel efficiency. A production system should switch between strategies based on real-time density relative to rho_c.

C5 -- Activate the advisory only above 0.9 x rho_c.

The density sweep shows near-zero benefit below density = 25 veh/km. The advisory should activate only when density exceeds 24 veh/km, to prevent false activations that erode driver trust in the navigation app integration.

PART VI

Deployment

Roadmap to Live Navigation App Integration

Phase Roadmap

Phase 1 -- Core Engine [Complete]	LWR model + Godunov solver + 4 controllers + REST API + interactive dashboard.
Phase 2 -- Real-Time Data Ingest [Next]	Replace simulated density with live Waze for Cities GeoRSS/JSON feed or HERE Traffic API. Calibrate V_f, rho_jam, and w using NGSIM trajectory data (data.transportation.gov) for each target road segment.
Phase 3 -- RL Policy Optimisation [Planned]	Train a PyTorch PPO reinforcement learning agent within UXsim to learn the Efficient Frontier across all density conditions and penetration rates, replacing the hand-coded JAD heuristic with a data-driven controller.
Phase 4 -- Cloud + App Integration [Future]	Deploy as AWS Lambda / Google Cloud Functions for sub-100ms advisory latency. Push to Waze via CIFS Dynamic Speed Limit feed (waze.com/partnerhub), Google Maps via Routes API speedReadingIntervals, and V2X via SAE J2735 TIM.

Platform	Method	Format	Reference
Waze for Cities	Partner Hub API	CIFS Dynamic Speed Limit	waze.com/partnerhub
Google Maps	Routes API + Nav SDK	speedReadingIntervals	developers.google.com/maps
Apple Maps	MapKit JS	Custom speed annotation	developer.apple.com/maps
V2X / DSRC	Roadside unit broadcast	SAE J2735 TIM message	Infrastructure contract required

Table 4 -- Navigation app integration summary.

PART VII

User Guide

Installation, Dashboard Controls, and Troubleshooting

Installation

Requirements and setup:

```
pip install flask numpy

cd traffic_simulator
python run.py
# Open browser at: http://localhost:5050
```

Project structure:

```
traffic_simulator/
|-- run.py (entry point)
|-- backend/
| |-- __init__.py (empty file -- required)
| |-- fundamental_diagram.py
| |-- lwr_model.py
| |-- controllers.py
| |-- simulator.py
| +-- api.py
+-- frontend/
+-- index.html
```

Dashboard and Output Reference

Control	Range	Notes
Road Length	5-30 km	10 km is a standard freeway segment
Sim Duration	5-45 min	20 min captures full formation and controller response
Base Density	5-130 veh/km	Values above 26.7 veh/km enter the unstable regime
Perturbation Size	0-80 veh/km	20-30 veh/km seeds a realistic phantom jam
Jam Position	10-90%	Position of the density spike along the road
Strategy	4 options	JAD for jam elimination; FollowerStopper for smoothness
AV Penetration	0-30%	5-10% is the realistic near-term deployment target
Driver Compliance	10-100%	Fraction of AV users who follow the advisory speed
Free-Flow Speed	60-140 km/h	V_f: 100-120 for motorways, 60-80 for arterials
Jam Density	80-250 veh/km	rho_jam: 160 veh/km is standard for freeways
Wave Speed	-5 to -35	w: -20 km/h is the empirically universal value

Table 5 -- Dashboard control reference.

Output	How to Read It
Speed Advisory	The km/h value to push to navigation app users upstream of the jam. Temporary deceleration -- speeds normalise once the wave is absorbed.
Heatmap	Y = time, X = road position. Blue = free-flow, red = jam. Diagonal red bands moving left = backward phantom waves. AV control fragments or eliminates these bands.
Efficiency Score	avg_flow x (1 - jam_fraction)^0.5. Higher is better. Use this to compare controllers on the same scenario.
Speed Variance (sigma)	Standard deviation of speed across the road. Below 10 km/h = smooth; above 22 km/h = significant stop-and-go.
Jam Gone?	YES if final-quarter jam fraction is 30% lower than the first quarter -- indicating sustained absorption rather than temporary suppression.

Table 6 -- Output interpretation guide.

Troubleshooting

Error	Fix
No module named 'simulator'	Add to run.py: sys.path.insert(0, os.path.join(os.path.dirname(os.path.abspath(__file__)), 'backend'))
No module named 'backend'	Add to run.py: sys.path.insert(0, os.path.dirname(os.path.abspath(__file__)))
No module named 'flask'	Run: pip install flask numpy
All controllers identical	In lwr_model.py, ensure AV actuation caps F_interior[cell_idx], not a separate modified array.
Heatmap entirely blue	Perturbation amplitude too small. Use 20+ veh/km at base density 35+.

Table 7 -- Common errors and fixes.

References

Lighthill & Whitham (1955); Richards (1956): Original LWR kinematic wave traffic model.

Daganzo (1994): Cell transmission model -- supply/demand Godunov scheme.

Sugiyama et al. (2008): Circular road experiment: spontaneous jam formation. New Journal of Physics 10, 033001.

Stern et al. (2018): Ring-road AV experiment validating FollowerStopper. Transportation Research Part C.

He et al. (2016): Jam-Absorption Driving formalisation. Transportation Research Part B.

NGSIM -- FHWA: Vehicle trajectory datasets (US-101, I-80). data.transportation.gov.

Institute of Global Synthesis & Innovation | FlowState v1.0 | March 2026

LWR Model | Godunov Scheme | Triangular FD | JAD | FollowerStopper | PI with Saturation