TOWARD A TRAFFIC WORLD MODEL: LEARNING THE PHYSICS OF SIGNALIZED ROADS WITH GUIDED DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Traffic engineers describe how vehicles move along a road with physics-based models built from a few equations. These models are compact and interpretable, but each fits one kind of measurement and must be calibrated by optimization before use. We ask whether a generative model trained only on traffic states and the corresponding signal timing can learn the physics of a signalized road, and whether it can serve as a world model that answers questions from partial sensor data. We present DiffLink, a diffusion model over the time-space diagram of a signalized link. The state is a two-channel image of density and flow, the signal is the action, and no traffic equation enters training. On SUMO data, states generated from the signal alone reproduce queue formation and discharge. Their mean arrival rate, delay, and maximum queue length match the test set within 4%, and their fundamental diagram, discharge flow, and queue-tail growth rate match the simulator's. When the green is shortened, the model's queue grows by 4.9 m against the simulator's 5.1 m, at a correlation of 0.93 across the 32 simulations. Because the model is a prior over full states, any sensor becomes a query at sampling time, with no retraining. Guided by two stop-bar detectors, DiffLink estimates the arrival rate as well as a vehicle count does, and it reconstructs the maximum queue length per window with a correlation of 0.74, against 0.51 for deterministic queuing from the same counts. We close with what still separates DiffLink from a full traffic world model.

1 INTRODUCTION

A world model is a learned model of how an environment changes. An agent can query it in place of the environment itself (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2023). The idea now drives work on autonomous driving, where generative video models predict the scene ahead of a vehicle (Hu et al., 2023; Wang et al., 2024a; NVIDIA et al., 2025). Road traffic has had models of this kind for seventy years, but they were derived from first principles, not learned from data. The kinematic wave model (Lighthill & Whitham, 1955; Richards, 1956) and the cell transmission model (Daganzo, 1994) describe how density and flow evolve along a road with a few equations and a handful of parameters. They are compact and interpretable, and traffic engineers use them every day. Learning those dynamics from data is now a problem in its own right (Xiang et al., 2026).

Physics-based models have two costs. First, each one is tied to one kind of measurement. The cell transmission model works with loop detectors, which count vehicles at fixed points, but it has no natural way to use a sparse set of vehicle trajectories. Probabilistic time-space diagrams (Wang et al., 2024b) use trajectories well but are hard to fit with detector counts. Second, every model must be calibrated before use, and calibration is an optimization problem whose objective and constraints depend on the sensor (Zhou et al., 2024). Fusing several sensors means solving a new such problem each time.

This paper asks whether a generative model can take the place of the physics-based model. We use the term world model in a narrow sense. It is a learned, probabilistic model of the state of a physical system, conditioned on the action applied to the system. It can be queried with partial observations. We present DiffLink, which meets this definition on a single signalized road link. The

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

state is the time-space diagram of the link, encoded as a two-channel image of density and flow. The action is the signal timing. A diffusion model learns the distribution of states given the action from simulated data, and no traffic equation appears in its loss. Diffusion models already generate the full spatiotemporal field of a physical system in place of a numerical solver (Zhou et al., 2025), and the time-space diagram is that field for a road. Other learned traffic models write conservation and the road graph into the architecture (Xiang et al., 2026). We leave that structure out, so that what the samples do is evidence about what the data holds. Sensor data never enters training. At sampling time, an observation becomes a loss on the denoised state, and posterior sampling (Chung et al., 2022) steers generation toward states that agree with it. Loop detectors, probe trajectories, and both together are three such losses on one prior.

This design changes what calibration means. The prior already holds the relationship between signal, arrivals, queue, and discharge, so there is nothing to fit when a new sensor arrives, and every answer comes as a set of samples that shows its uncertainty. Grounding a learned simulator in sparse sensor readings is an established idea (Linkerhägner et al., 2023), but here the sensor is not part of the model, it is a loss at sampling time.

We test DiffLink on SUMO simulations of a two-lane signalized link. Three findings support the world-model claim. First, states generated from the signal alone behave like traffic. Queues build during red and discharge during green. The mean arrival rate, delay, and maximum queue length match the test set within 4%, and the fundamental diagram, the discharge flow, and the growth rate of the queue tail match the simulator's. Second, the model responds to the action. When the green is shortened, the model's queue grows by 4.9 m against the simulator's 5.1 m, with a correlation of 0.93 across the 32 simulations. Third, the learned model is useful downstream. Guided by two detectors at the stop bar, it estimates the arrival rate as well as a vehicle count does. It reconstructs the maximum queue length of each window with a correlation of 0.74 to the truth, against 0.51 for deterministic queuing from the same counts. The detectors do not see the queue tail. The information travels through the learned coupling between signal, discharge, and queue, and only the exact gradient of the observation loss through the denoiser carries it.

DiffLink is a first step. It covers one fixed window of time, and it is trained on simulation. Section 5 states the gaps. Our contributions are:

- A learned traffic flow model for a signalized link. We encode the time-space diagram as a two-channel image and the signal as an action vector, and we learn a diffusion prior over them from simulated data, with no traffic equation in the loss.
- Evidence that the prior learns the physics of the road. Samples reproduce queue formation and discharge and match the distributions of arrival rate, delay, and queue length. They also recover the simulator's fundamental diagram, discharge flow, and queue-tail growth rate, and they respond to a change of signal timing as the simulator does.
- State estimation and forecasting as downstream queries. Posterior sampling turns the prior into an estimator for any sensor written as a loss on the state. With stop-bar detectors it matches a vehicle count on the arrival rate and reconstructs the queue far better than deterministic queuing. Hiding the second half of a window turns the same sampler into a forecaster, whose queue correlation of 0.89 beats the nearest training window at 0.71 and deterministic queuing at 0.38. Adding probe trajectories improves the reconstruction further but exposes the need for an observation model of partial data.

2 Related work

Traffic models and heterogeneous data. Fusing detector and probe data is standard practice in traffic state estimation (Seo et al., 2017; Zhou et al., 2024). Deng et al. (2013) map loop counts, Bluetooth travel times, and GPS probes onto cumulative vehicle counts, and Clairais et al. (2016) fit the kinematic wave model with detectors as boundary conditions and probes as travel-time observations. Physics-informed neural networks add the kinematic wave model to the training loss and fuse detector and floating-car data (Shi et al., 2021; Rempe et al., 2021), and joint state and parameter estimators run online on freeways (Wang et al., 2022; Agalliadis et al., 2020). All of these keep an explicit model and fit its parameters, so every new sensor needs a new objective.

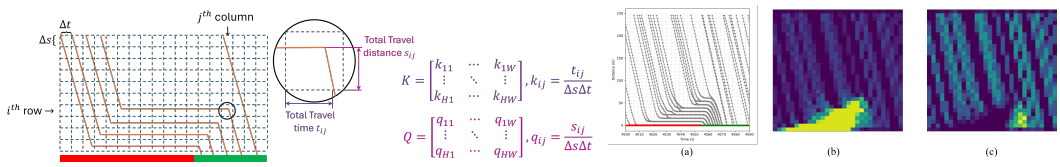

Figure 1: State encoding. Left: the time-space diagram of a link is cut into an $H \times W$ grid, and Edie's definitions give the flow and density of each cell. Right: one time-space diagram (a), its density channel K (b), and its flow channel Q (c). The signal is encoded as a vector with one entry per column.

Generative models of traffic states. Lei et al. (2024) estimate states at sensor-free locations with a conditional diffusion model, and Zhang et al. (2025) forecast network speeds conditioned on weather and calendar data. On arterials, Xu et al. (2024) reconstruct trajectories with generative adversarial networks guided by car-following physics, and Ranjan et al. (2026) learn an actor-level simulator of signalized intersections. The closest work is PMA-Diffusion (Liu et al., 2025), which trains an unconditional diffusion prior over freeway speed fields and fills gaps from sparse detectors and probes by posterior sampling with a physics-guided smoothing step. DiffLink differs in two ways. It models density and flow on a signalized link, where the signal drives the dynamics and enters as an action. And it asks what the prior has learned, not only how well it fills gaps.

World models and physical understanding. World models learn latent dynamics and use them for imagination and control (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2023). Driving world models generate video of the scene ahead of a vehicle, and are judged on how real the video looks (Hu et al., 2023; Wang et al., 2024a; NVIDIA et al., 2025). Kang et al. (2024) find that video models follow the training distribution more than physical law, and WorldBench isolates single physics concepts to test them (Upadhyay et al., 2026). Traffic offers a controlled setting for the same question, because simulation gives the full state and the governing physics is known.

Diffusion models for inverse problems. Classifier guidance (Dhariwal & Nichol, 2021) adds a gradient to each denoising step, and diffusion posterior sampling does the same with an observation loss (Chung et al., 2022). Projection methods such as DDNM and DiffPIR instead overwrite the observed part of the estimate (Wang et al., 2023; Zhu et al., 2023). We use posterior sampling with the gradient taken through the denoiser, because, as Section 3.4 explains, a projection-style update cannot move information between cells.

3 DiffLink: A Learned Traffic Flow Model with Sensors as Queries

3.1 State and Action

DiffLink models one road link that ends at a signalized intersection. Its state over a window of time is the time-space diagram, which shows every vehicle trajectory on the link. We follow Edie's generalized definitions (Edie, 1963) to turn the diagram into a grid. Let Δs be the segment length and Δt the time step, so a link of length $H\Delta s$ over a window of length $W\Delta t$ gives an $H \times W$ grid. For cell (i, j), let t_{ij} be the total time that all trajectories spend in it and s_{ij} the total distance they travel in it. Flow and density are

$$q_{ij} = \frac{s_{ij}}{\Delta t\, \Delta s}, \qquad k_{ij} = \frac{t_{ij}}{\Delta t\, \Delta s}. \tag{1}$$

Stacking the two matrices $K = [k_{ij}]$ and $Q = [q_{ij}]$ gives the state $\boldsymbol{x}_0 = \mathrm{stack}(K, Q) \in \mathbb{R}^{H \times W \times 2}$, a two-channel image in which row 1 is the upstream end of the link and row H is the stop bar (Figure 1).

The action is the signal timing of the link over the window. The raw signal is a binary vector with one entry per second, equal to 1 during green. We sum it over each Δt to get $\boldsymbol{c} \in \mathbb{R}^W$, whose entry is the number of green seconds in that step. State and action share the time axis, so the network can read the signal column by column.

Traffic engineers judge a link by a few quantities, and all of them are functions of $\boldsymbol{x}_0$. The arrival rate is the mean flow in the upstream row, $\lambda = \frac{1}{W}\sum_j q_{1j}$. The speed in a cell is $v_{ij} = q_{ij}/k_{ij}$ when $k_{ij} > 0$ and 0 otherwise. Maximum queue length counts cells whose speed falls below a stop threshold, and average delay compares each cell's speed with the free-flow speed. Appendix A gives both definitions. Because they are functions of the state, we compute them for a generated state exactly as for a simulated one.

3.2 THE PRIOR

Diffusion models (Ho et al., 2020; Song et al., 2020) learn a distribution by reversing a process that adds Gaussian noise. With a noise schedule $\{\beta_t\}_{t=1}^{T}$ and $\bar{\alpha}_t = \prod_{s\le t}(1-\beta_s)$, the noisy state at step t is

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

A network $v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ predicts the velocity target $\boldsymbol{v}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{x}_0$ (Salimans & Ho, 2022) under the min-SNR weighted loss of Hang et al. (2023),

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon},t}\left[w_t\,\|\boldsymbol{v}_t - v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})\|^2\right], \qquad w_t = \frac{\min(\mathrm{SNR}_t, \gamma_{\mathrm{snr}})}{\mathrm{SNR}_t + 1}, \quad \mathrm{SNR}_t = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}. \tag{3}$$

We predict velocity rather than noise because most cells of a traffic state are exactly empty and must come out exactly empty. The velocity target stays well conditioned at both ends of the schedule, and the weight keeps the nearly clean steps from dominating the average. The predicted noise and clean state follow from the velocity as $\boldsymbol{\epsilon}_\theta = \sqrt{\bar{\alpha}_t}\,v_\theta + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_0 = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\,v_\theta$.

Sampling starts from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applies the deterministic DDIM update (Song et al., 2020)

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{\boldsymbol{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}). \tag{4}$$

The prior learns the whole distribution rather than a point estimate, conditions on the action directly, and can be steered at sampling time. The next section builds on that last property.

3.3 OBSERVATIONS AS QUERIES

A sensor gives partial information $\boldsymbol{y}$ about $\boldsymbol{x}_0$. We write each sensor as a loss $\mathcal{L}_{\mathrm{obs}}(\hat{\boldsymbol{x}}_0; \boldsymbol{y})$ on the denoised estimate and leave the prior untouched. Figure 5 shows the two sensors we study.

Loop detectors. Detectors measure flow and occupancy at fixed points, usually near the stop bar and sometimes farther upstream. In our encoding a detector observes one row of both channels. Let M_{det} select the detector rows and $\boldsymbol{y} = M_{\mathrm{det}} \odot \boldsymbol{x}_0$, where $\odot$ is the element-wise product. The loss is

$$\mathcal{L}_{\mathrm{det}} = \|M_{\mathrm{det}} \odot \hat{\boldsymbol{x}}_0 - \boldsymbol{y}\|_2 . \tag{5}$$

Probe trajectories. A fraction p of vehicles report their trajectories. The encoding of Eq. equation 1 applied to these trajectories alone gives a partial state $\boldsymbol{x}'_0 = \mathrm{stack}(K', Q')$ with $k'_{ij} \le k_{ij}$ and $q'_{ij} \le q_{ij}$ in every cell. Let M_{traj} mark the cells with $k'_{ij} > 0$ and $\boldsymbol{y} = M_{\mathrm{traj}} \odot \boldsymbol{x}'_0$. The observation is a lower bound, so the loss has a one-sided term that penalizes states below the bound and a symmetric term that keeps the state near the observation,

$$\mathcal{L}_{\mathrm{traj}} = \xi\,\|\max(\boldsymbol{y} - M_{\mathrm{traj}} \odot \hat{\boldsymbol{x}}_0,\, \mathbf{0})\|_2 + \|\boldsymbol{y} - M_{\mathrm{traj}} \odot \hat{\boldsymbol{x}}_0\|_2 , \tag{6}$$

with ξ setting the balance. The symmetric term is convenient but biased in a specific way. A probe's contribution to a cell equals the truth only when it is the only vehicle there, so $\boldsymbol{y}$ sits below the full state wherever a probe shares a cell, and the term pulls the reconstruction down with it. Section 4.5 measures this bias.

Combining sensors. With both sensors the loss is a weighted sum,

$$\mathcal{L}_{\mathrm{obs}} = \alpha_{\mathrm{det}}\,\mathcal{L}_{\mathrm{det}} + \alpha_{\mathrm{traj}}\,\mathcal{L}_{\mathrm{traj}}. \tag{7}$$

We tune the weights on held-out windows by maximizing the correlation between the arrival rates and queue lengths of guided samples and their ground truth. Other queries take the same form. Observing the first columns of a window and leaving the rest free turns the sampler into a forecaster, which Section 4.4 evaluates.

3.4 POSTERIOR SAMPLING

Diffusion posterior sampling (Chung et al., 2022) adds a gradient step to Eq. equation 4,

$$\boldsymbol{x}_{t-1} \leftarrow \boldsymbol{x}_{t-1} - \gamma \, \nabla_{\boldsymbol{x}_t} \mathcal{L}_{\text{obs}}\big(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)\big), \tag{8}$$

where γ sets the guidance strength and the loss is evaluated separately for each sample in a batch. The gradient is taken with respect to $\boldsymbol{x}_t$ through the denoising network. This detail decides whether the prior takes part in the query. If the network output is treated as a constant, the gradient of the loss reduces to a rescaled copy of the observation residual and touches only observed cells. With the exact gradient, the Jacobian of the denoiser spreads the residual to unobserved cells according to the spatial and cross-channel structure the network has learned. For our model, the share of the squared gradient norm that falls on unobserved cells rises from 12% at $t = 50$ to 40% at $t = 300$ and above 70% for $t \geq 600$, and it is zero under the truncated form. The price is that the guidance weights depend on this choice and on the normalization of the state, and they must be re-tuned when either changes. We apply guidance for $0.05\,T < t < 0.95\,T$.

4 EXPERIMENTS

4.1 SETUP

Simulation. We generate ground truth with SUMO (Lopez et al., 2018) on a small network of two signalized intersections and model one two-lane link of 256 m that ends at the first of them. Each simulation draws a cycle length C from $\{80, 90, 100, 110, 120\}$ s. The four phases have minimum greens of 20 s for the link and 10 s for each other phase, and the remaining $C - 50$ s are split among them at three random cut points, so short greens for the link are more common than long ones (Figure 4). Demand is drawn per route. Each route takes a share of the capacity the signal allows, the saturation flow of 0.5 veh/s per lane times the number of lanes and the green split, scaled by a factor drawn uniformly between 0.45 and 0.9. A route through both intersections takes the smaller of its two values, so demand stays below capacity and queues never spill back. The speed limit is 13.89 m/s. Appendix B lists all settings.

Data. We simulate 1024 training runs of 6144 s each. With $\Delta t = 3$ s and $\Delta s = 8$ m, a 96 s window becomes a 32×32 grid, and training windows are cut at random offsets within each run. For evaluation we simulate 32 further runs with fresh signal plans and demands between 0.095 and 0.216 veh/s, discard the first 192 s of each so that no window starts on an empty link, and keep 8 consecutive windows per run, 256 windows in total. Each channel is scaled to $[-1, 1]$ with a square-root min–max map, $x' = 2\sqrt{(x - x_{\min})/(x_{\max} - x_{\min})} - 1$. Empty cells still map to -1, but the square root moves small non-zero values, such as a single free-flowing vehicle, away from that floor. With a linear map they sit within 5% of the range above that floor, and the trained model leaves a low-density haze over empty cells. The square-root map cut the share of generated cells in the 0.0013–0.005 veh/m band from 35% to 1.3%, against 1.1% in the ground truth.

Model and training. The denoiser is a U-Net (Ronneberger et al., 2015) with channel widths of 128, 256, and 384, built on the `diffusers` library (von Platen et al., 2022). The action enters twice, through a FiLM layer (Perez et al., 2018) on the time-step embedding and as a learned four-channel embedding of each column's signal state tiled along the space axis, so every column carries its own signal state at full resolution. Appendix C lists the training settings. All results use the epoch-400 checkpoint, after which the held-out loss no longer changed.

Sampling and evaluation. We sample with 1000 DDIM steps and $\gamma = 0.5$. The guidance weights are $\alpha_{\text{det}} = 0.2$ and $\alpha_{\text{traj}} = 1.0$ with both sensors, $\alpha_{\text{det}} = 0.1$ with detectors alone, and $\xi = 2$. Detectors occupy four rows at 0, 8, 48, and 56 m upstream of the stop bar, and the probe penetration rate is $p = 0.3$. Before any metric is computed, cells whose density or flow falls below 0.005 veh/m or 0.005 veh/s are set to zero. Ground-truth states contain exact zeros and are unaffected. A lower threshold lets background values register as stopped vehicles and inflate queue lengths, so the threshold is part of the protocol. Average delay divides by the number of arrivals, so we report it over windows with at least 0.03 veh/s, about three vehicles, which excludes one test window. Appendix C collects all hyperparameters.

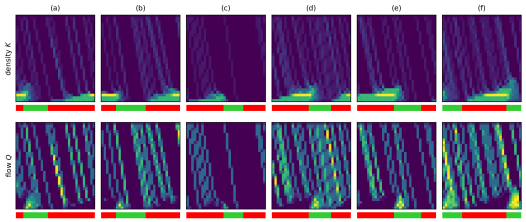

Figure 2: States generated from the signal alone, for six held-out signal plans (a)–(f). In each column the top image is density and the bottom image is flow, the stop bar is the bottom row, and the bar under each image is the signal, red or green, for each time step. Queues build upstream of the stop bar during red and discharge during green.

4.2 Does the prior learn the physics of the road?

Figure 2 shows states generated from the signal alone. Nothing else is given, and no traffic equation was used in training. Queues form upstream of the stop bar during red, the queue tail moves upstream as vehicles join, and the discharge front moves upstream once the signal turns green. These are the shockwaves that the kinematic wave model was written to describe, and the prior produces them from data.

We then ask whether the samples are right in aggregate. Because generation is conditioned only on the signal plans of the test set, the distribution of a metric over generated states should match its distribution over the test states. Figure 7 in Appendix H plots the joint distributions of arrival rate with average delay and with maximum queue length over the 256 test windows, computed with the definitions of Appendix A. Table 1 gives the means, standard deviations, and the Kolmogorov–Smirnov distance between the two distributions. The means match within 1% for delay and arrival rate and within 4% for queue length, and no distribution differs from its test-set counterpart at the 5% level. The model has learned a stochastic map from signal to state whose summary statistics agree with the simulator.

Summary statistics can agree while the mechanics differ, so we probe three relations that the kinematic wave model treats as laws (Figure 8 in Appendix H). The fundamental diagram of generated cells has the same shape as the simulator's, with a binned-mean flow that peaks at 0.69 veh/s against 0.67 at the same density of 0.07 veh/m, and a root-mean-square difference between the two binned curves of 0.09 veh/s. The discharge flow at the stop bar in the first 12 s of green after a red of at least 15 s averages 0.480 veh/s in generated states and 0.482 in the test set. The speed at which the queue tail moves upstream during red, fitted over every red phase of at least 24 s, is 0.505 ± 0.253 m/s against 0.502 ± 0.260 m/s. The prior does violate the physics at the margin. Among occupied generated cells, 6.0% exceed the two-lane jam density of 0.25 veh/m, against 2.0% in the test set.

A lookup table as control. Would a model that only memorized the training set pass these tests? For each test window we return the training window whose signal is nearest over all 1024 runs and every time offset, and 98% of the test signals have an exact match. This lookup matches the test distributions of delay and queue as well as DiffLink does (KS 0.082 for both) and misses only on the arrival rate (0.137). It passes the probes by construction, because it returns simulated states. Table 1 and the probes therefore show that the marginals are right, not that the model learned the dynamics. Three tests separate the two, and the lookup appears in each: the cycle lengths below, Table 2, and the counterfactual of Section 4.3.

Outside the training range. Training cycles run from 80 to 120 s. We generate 16 simulations each at 60 s and 140 s and repeat the probes (Table 7 in Appendix E). Two behaviors come apart. The marginal over demand does not extrapolate. Generated arrival rates regress toward the training mean, 0.158 against 0.179 veh/s at 60 s and 0.141 against 0.119 at 140 s, and the distribution match of Table 1 fails at the 5% level. The local physics does transfer. The peak of the fundamental diagram and the discharge flow stay within 1 to 9% of the simulator's, and queue growth within 13%. So does the estimator of Section 4.4, whose queue correlation with four detectors is 0.68 and 0.94 at 60 and 140 s against 0.47 and 0.66 for deterministic queuing. The lookup table has no plan to borrow at 60 s. Only 6% of its matches are exact, its arrival KS is 0.49, and the states it returns

Table 1: Arrival rate, average delay, and maximum queue length over 256 test windows and over states generated from their signal plans. KS is the Kolmogorov–Smirnov distance between the two distributions. The 5% critical value for two samples of 256 is 0.12.

Source	Arrival rate (veh/s) Mean	Std	Avg. delay (s) Mean	Std	Max queue length (m) Mean	Std
Test set	0.149	0.049	32.0	13.3	33.6	13.7
Generated states	0.148	0.050	31.7	16.5	32.3	13.2
Relative error of the mean	-0.4%		-0.8%		-3.9%	
KS distance	0.074		0.075		0.059	

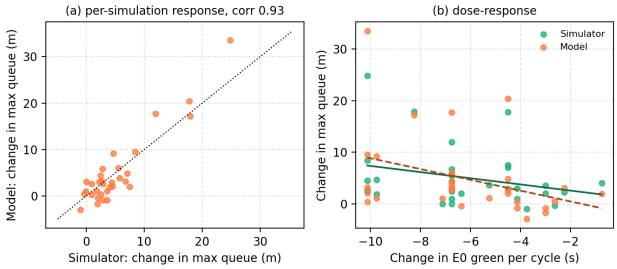

Figure 3: Response to a shorter green. (a) Change in maximum queue length per simulation when the link's green is shortened, simulator against DiffLink guided by the two upstream detector rows with the same initial noise under both plans. (b) The same changes against the change in green time per cycle, with linear fits.

no longer follow the queried signal. We measure this with the switch contrast, the stop-bar flow in the 6 s after the green starts over the flow in the 6 s before. It is 23 in the simulator, 19 for DiffLink, and 11 for the lookup, whose queue correlation with four detectors falls to 0.42, below deterministic queuing. The prior has learned how a queue forms and discharges, not which demand goes with a cycle length it never saw.

4.3 Does the model respond to the action?

A traffic flow model must predict what a change of signal timing does to the link. We test this with a paired counterfactual. The 32 test simulations are rerun with identical seeds and demands but with 8 s of green per cycle moved from the phase that serves the link to another phase, which shortens the link's green by 6.2 s on average after clamping at the minimum green. In the simulator, the maximum queue grows by 5.1 m per window and the delay by 7.3 s. We then ask the model the same question. Demand must be held fixed for the comparison, so both plans are sampled with guidance from the two upstream detector rows at 48 and 56 m, which count arrivals but do not see the queue, and with the same initial noise. Figure 3 shows the result. The model's queue grows by 4.9 m against the simulator's 5.1 m, and across the 32 simulations the two changes correlate at 0.93, with a bootstrap 95% interval of 0.75 to 0.97. The nearest training window, keyed on the same detectors, responds with 4.4 m and a correlation of 0.72, with an interval of 0.38 to 0.85. The supervised network of Section 4.4, given the same detectors, responds with 4.9 m and a correlation of 0.95. For delay the figures are 5.9 s against 7.3 s at a correlation of 0.84. Without the demand anchor the mean response is almost lost, 0.3 m, because unguided samples redraw the demand along with the plan. The model has learned the coupling between signal and queue as a response, not only as a correlation in the training set. The fitted slope, 1.05 m against 0.60 m of queue per second of green removed, does overstate the sensitivity.

4.4 DOWNSTREAM APPLICATION: STATE ESTIMATION FROM SENSORS

A learned traffic flow model is useful when it answers questions that sensors alone cannot. We evaluate three on the 256 test windows: the arrival rate at the upstream end of the link, the maximum queue length, and the average delay, each compared with its ground truth by the correlation across windows. We use three detector layouts: the four rows of Section 4.1, the two stop-bar rows alone, which are the common field installation, and the two upstream rows of Section 4.3. A detector counts vehicles, so the natural baselines are counts. For the arrival rate we take the mean flow at the most upstream detector. For the queue we take deterministic queuing, that rate times the longest red in the window, spread over two lanes at 8 m per vehicle. Table 2 reports both next to DiffLink.

Two things stand out. First, the prior adds little on the arrival rate. With a detector 56 m upstream the count alone reaches 0.89 and DiffLink 0.84, because queues reach that detector in only 10% of the windows, so it is an arrival counter. With stop-bar detectors alone the two tie at 0.67 and 0.68. Second, it adds a lot on the queue. Deterministic queuing from the same counts reaches 0.55 and 0.51, while DiffLink reaches 0.90 and 0.74. The nearest training window, keyed on the same signal and detector rows, reaches 0.76 and 0.67, because it can only copy a stored window whose detector rows happen to agree. DiffLink also returns a delay estimate, at 0.71 and 0.58, which no count provides. The queue tail lies upstream of the stop-bar detectors in most windows, so the queue is inferred through the learned coupling of signal, discharge, and queue. The prior alone, which sees only the signal, reaches 0.21. The estimator is the learned model itself, which is why the exact gradient of Section 3.4 matters.

Each reconstruction is a sample (Figure 6 in Appendix G), so we can draw many and read off an estimate and its spread. With eight samples under stop-bar guidance the central 80% interval covers the true queue in 66% of windows, against 81% for the prior alone, so the guided spread is too narrow (Appendix D).

A network trained for the layout sets the ceiling for a point estimate. We train the same U-Net to map the detector rows, their mask, and the signal straight to the full state, with the mean squared error in physical units as the loss and one network per layout (Appendix C). Its queue correlation is 0.97 against 0.90 for DiffLink with four detectors, 0.96 against 0.83 with the two upstream rows, and 0.78 against 0.74 with the stop-bar rows, where it also beats DiffLink on the arrival rate and the delay, and with four detectors its error in unobserved cells is a quarter of DiffLink's. The detector rows and the signal nearly determine the state, a network trained on that layout learns the map directly, and guidance carries the same information through the denoiser only in part, a gap a better inverse-problem solver would recover. What it cannot do is what the rest of this paper is about. It answers only for its layout, so the three layouts took three networks. It returns one estimate and no spread. Without sensors it has no output, so nothing in Sections 4.2 and 4.3 can be asked of it.

The same prior forecasts. A world model must also roll forward, and the mask interface poses this as a third query without retraining. The first 48 s of a window are observed in full, the plan is known, the last 48 s are hidden, and Table 8 in Appendix F scores them. DiffLink forecasts their maximum queue at a correlation of 0.89 with an unbiased mean, against 0.71 for the nearest training window, 0.38 for deterministic queuing from the observed arrivals, and 0.21 for persistence. A network trained for this mask reaches 0.90, but as a mean over futures it blurs the queue to 21.4 m against 27.3, while the median of eight DiffLink samples reaches 0.92 and their central 80% interval covers the true queue in 70% of windows, with a mean width of 8.8 m. A graph network simulator (Linkerhägner et al., 2023) adapted to the link and rolled forward from the observed half reaches 0.87 on the queue but forecasts 0.03 veh/s of arrivals against 0.15, since a step-by-step update has no arrival process at the boundary, which the prior carries. No method forecasts the arrivals above a correlation of 0.36. Seven vehicles in 48 s are mostly noise, so the queue forecast rests on the signal and the carried-over state.

4.5 ADDING PROBE TRAJECTORIES

Trajectories alone are a weak sensor at a 30% penetration rate, with correlations of 0.34 for arrivals and 0.69 for the queue against 0.21 for the prior alone. Added to detectors with Eq. equation 6, they raise the queue correlation from 0.74 to 0.80 with stop-bar detectors and lower the error in unobserved cells from 0.030 to 0.028 veh/m. They also raise the delay correlation from 0.58 to 0.73

Table 2: Sensor queries on the 256 test windows. Correlations are between per-window estimates and ground truth; the true arrival rate is 0.149 veh/s. The count baseline takes the mean flow at the most upstream detector as the arrival rate and deterministic queuing, that rate times the longest red over two lanes at 8 m per vehicle, as the queue. The nearest training window is the lookup of Section 4.2 keyed on the signal and the detector rows. The supervised U-Net maps the detector rows and the signal to the full state, one network per layout. Delay is over the 255 windows with at least 0.03 veh/s of arrivals. The last block is the upstream layout of Section 4.3; Appendix D gives bootstrap intervals.

Detectors	Guidance	Arrival (veh/s)	Arr. corr.	Queue corr.	Delay corr.
–	none (prior only)	0.148	−0.02	0.21	0.29
–	trajectories only	0.135	0.34	0.69	0.57
0, 8, 48, 56 m	count baseline	0.153	0.89	0.55	–
0, 8, 48, 56 m	nearest training window	0.140	0.68	0.76	0.64
0, 8, 48, 56 m	supervised U-Net, this layout	0.144	0.95	0.97	0.96
0, 8, 48, 56 m	detectors	0.145	0.84	0.90	0.71
0, 8, 48, 56 m	detectors + probes, Eq. equation 6	0.115	0.77	0.91	0.68
0, 8 m	count baseline	0.150	0.68	0.51	–
0, 8 m	nearest training window	0.147	0.59	0.67	0.64
0, 8 m	supervised U-Net, this layout	0.139	0.79	0.78	0.83
0, 8 m	detectors	0.153	0.67	0.74	0.58
0, 8 m	detectors + probes, Eq. equation 6	0.125	0.62	0.80	0.73
48, 56 m	nearest training window	0.127	0.62	0.71	0.50
48, 56 m	supervised U-Net, this layout	0.144	0.95	0.96	0.94
48, 56 m	detectors	0.139	0.80	0.83	0.53

with stop-bar detectors, but they bias the arrival estimate down by 16 to 23%. The mechanism is the one stated in Section 3.3. The observation is never above the full state, and the symmetric term pulls the reconstruction toward it, so stopped cells came out at 0.13 veh/m against 0.25 in truth. A correction derived from binomial thinning removed the bias at low demand but over-corrected at high demand, and an observation model that separates shared cells from single-vehicle cells is left for future work. The point is general. A world model answers queries only through an observation model, and a misspecified one turns a well-calibrated prior into a biased estimator.

5 Limitations and what a traffic world model still needs

Four gaps separate DiffLink from a full traffic world model. First, it covers one 96 s window. Section 4.4 forecasts 48 s, but a world model must roll forward for longer, which needs a different architecture. Second, it is trained on simulation, so the observation models transfer as written but the prior does not, and the gap to real arrivals is unknown. Third, sampling takes seconds per state with 1000 steps, and fewer steps or distillation are the obvious remedies. Fourth, guidance weights depend on the normalization and the observation model and must be re-tuned when either changes, and partial sensors such as probe trajectories still need an observation model that the loss of Section 3.3 only approximates.

6 Conclusion

DiffLink learns the traffic flow model of a signalized link as a diffusion prior over the time-space diagram, conditioned on the signal, with no traffic equation in its training. The prior reproduces queue formation and discharge, matches the simulator's distributions of arrival rate, delay, and queue length, recovers its fundamental diagram, discharge flow, and queue growth, and responds to a shorter green as the simulator does. Sensors as losses on the denoised state turn the same prior into an estimator and a forecaster, far better than deterministic queuing, while a network trained for one layout estimates better still and answers nothing else. DiffLink's quality as a world model will rest on how sensors are modeled and sampled as much as on the learned dynamics.

AI USE STATEMENT

In this work, we used generative AI tools to edit the text for grammar and concision, to help edit, debug, and organize code, and to help run experiments and monitor their progress. We have not used generative AI tools to generate research ideas, to produce or select experimental results, or to write related-work summaries without verification, and the remaining tasks with required disclosure are not applicable to this work. We have reviewed all AI-assisted work. AI-assisted code was checked and tested by the authors, and every number in the paper comes from experiments the authors designed and checked. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses simulated traffic data only. No human subjects, personal data, or real vehicle trajectories were involved. The method is meant for traffic state estimation and planning support. We see no direct path to harmful use beyond those of traffic modeling in general.

REPRODUCIBILITY STATEMENT

Section 4.1 and Appendices B and C give the simulation, data, model, training, and sampling settings. Appendix A defines every metric. The evaluation protocol, including the post-processing threshold, is stated in Section 4.1. Code for data generation, training, guided sampling, and evaluation is included in the supplementary material, with a README that maps each table and figure to the script behind it.

REFERENCES

Ioannis Agalliadis, Michail Makridis, and Anastasios Kouvelas. Traffic estimation by fusing static and moving observations in highway networks. In *20th Swiss Transport Research Conference (STRC 2020)(virtual)*. STRC, 2020.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Aurélien Clairais, Aurélien Duret, and Nour-Eddin El Faouzi. Calibration of the fundamental diagram based on loop and probe data. *Transportation Research Record*, 2560(1):17–25, 2016.

Carlos F Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation research part B: methodological*, 28(4): 269–287, 1994.

Wen Deng, Hao Lei, and Xuesong Zhou. Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach. *Transportation Research Part B: Methodological*, 57:132–157, 2013.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Leslie C Edie. Discussion of traffic stream measurements and definitions. In *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow*, pp. 139–154, 1963.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

Da Lei, Min Xu, and Shuaian Wang. A conditional diffusion model for probabilistic estimation of traffic states at sensor-free locations. *Transportation Research Part C: Emerging Technologies*, 166:104798, 2024.

Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves II. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A*, 229(1178): 317–345, 1955.

Jonas Linkerhägner, Niklas Freymuth, Paul Maria Scheikl, Franziska Mathis-Ullrich, and Gerhard Neumann. Grounding graph network simulators using physical sensor observations. In *International Conference on Learning Representations*, 2023.

Lindong Liu, Zhixiong Jin, and Seongjin Choi. PMA-Diffusion: A physics-guided mask-aware diffusion framework for TSE from sparse observations. *arXiv preprint arXiv:2512.06183*, 2025.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL `https://elib.dlr.de/124092/`.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

Yash Ranjan, Rahul Sengupta, Anand Rangarajan, and Sanjay Ranka. A generative model for closed-loop microsimulation of signalized intersections. *arXiv preprint arXiv:2606.23588*, 2026.

Felix Rempe, Allister Loder, and Klaus Bogenberger. Estimating motorway traffic states with data fusion and physics-informed deep learning. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 2208–2214. IEEE, 2021.

Paul I Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Toru Seo, Alexandre M Bayen, Takahiko Kusakabe, and Yasuo Asakura. Traffic state estimation on highway: A comprehensive survey. *Annual reviews in control*, 43:128–151, 2017.

Rongye Shi, Zhaobin Mo, Kuang Huang, Xuan Di, and Qiang Du. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11688–11698, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Rishi Upadhyay, Howard Zhang, Jim Solomon, Ayush Agrawal, Yunhao Ba, Alex Wong, Celso M de Melo, and Achuta Kadambi. WorldBench: Benchmarking physical understanding of world models by isolating physics concepts. *arXiv preprint arXiv:2601.21282*, 2026.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *European Conference on Computer Vision*, 2024a.

Xingmin Wang, Zachary Jerome, Zihao Wang, Chenhao Zhang, Shengyin Shen, Vivek Vijaya Kumar, Fan Bai, Paul Krajewski, Danielle Deneau, Ahmad Jawad, et al. Traffic light optimization with low penetration rate vehicle trajectory data. *Nature communications*, 15(1):1306, 2024b.

Yibing Wang, Mingming Zhao, Xianghua Yu, Yonghui Hu, Pengjun Zheng, Wei Hua, Lihui Zhang, Simon Hu, and Jingqiu Guo. Real-time joint traffic state and model parameter estimation on freeways with fixed sensors and connected vehicles: State-of-the-art overview, methods, and case studies. *Transportation Research Part C: Emerging Technologies*, 134:103444, 2022.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023.

Bairan Xiang, Chenguang Zhao, and Huan Yu. Micro-macro coupled koopman modeling on graph for traffic flow prediction. In *International Conference on Learning Representations*, 2026.

Mengyun Xu, Jie Fang, Prateek Bansal, Eui-Jin Kim, and Tony Z Qiu. Physics informed deep generative model for vehicle trajectory reconstruction at arterial intersections in connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 2024.

Huipeng Zhang, Honghui Dong, and Zhiqiang Yang. TSGDiff: Traffic state generative diffusion model using multi-source information fusion. *Transportation Research Part C: Emerging Technologies*, 174:105081, 2025.

Anthony Zhou, Zijie Li, Michael Schneier, John Buchanan, and Amir Barati Farimani. Text2PDE: Latent diffusion models for accessible physics simulation. In *International Conference on Learning Representations*, 2025.

Xuesong Simon Zhou, Xiangyong Roy Luo, Mohammad Abbasi, Zhitong Huang, Ankur Tyagi, et al. Emerging data cleaning and fusion for traffic model calibration–data fusion for microsimulation model calibration. Technical report, United States. Department of Transportation. Federal Highway Administration, 2024.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

A PERFORMANCE METRICS

All metrics are computed from the state $\boldsymbol{x}_0 = \mathrm{stack}(K, Q)$ with the speed field $v_{ij} = q_{ij}/k_{ij}$ for $k_{ij} > 0$ and $v_{ij} = 0$ otherwise.

Maximum queue length. A cell is stopped when its speed is below a threshold v_{stop} and it is not empty,

$$\delta_{ij} = \begin{cases} 1, & v_{ij} < v_{\text{stop}} \text{ and } k_{ij} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The queued distance at time step j weights each stopped cell by how close its speed is to zero,

$$L_j^{\text{queue}} = \sum_{i=1}^{H} \left(1 - \frac{v_{ij}}{v_{\text{stop}}}\right) \delta_{ij} \, \Delta s, \tag{10}$$

and the maximum queue length of a window is $L^{\text{queue,max}} = \max_{1 \le j \le W} L_j^{\text{queue}}$.

Average delay. The delay accumulated in cell (i, j) is

$$d_{ij} = k_{ij} \, \Delta s \, \Delta t \, \max\left(0, \, 1 - \frac{v_{ij}}{v_{\text{ff}}}\right), \tag{11}$$

where v_{ff} is the free-flow speed. The total delay is $D = \sum_{i,j} d_{ij}$, and the average delay per vehicle divides it by the number of arrivals, $d = D/(\lambda W \Delta t)$, with λ the arrival rate of Section 3.1.

B Simulation settings

Table 3 lists the SUMO configuration and Figure 4 shows the sampled arrival rates and green splits. The network has two signalized intersections, and the modeled link is the two-lane approach to the first. Its four phases share the slack $C - G_{\min}$ at three random cut points, so the link's green split is skewed toward short greens, and a route that crosses both intersections takes the smaller of the two demands their signals allow. The speed limit of the link is 13.89 m/s. SUMO draws each vehicle's desired speed from its default speed-factor distribution, a normal with mean 1 and standard deviation 0.1 times the limit, so free-flowing vehicles exceed the limit by up to about 20%. The delay metric therefore uses a free-flow reference of $v_{\text{ff}} = 13.89/0.9 \approx 15.4$ m/s, so that vehicles driving at or somewhat above the limit register no delay.

Table 3: SUMO configuration for each simulation

Parameter	Meaning (unit)	Value
L	Link length (m)	256
n	Number of lanes	2
C	Cycle length set (s)	{80, 90, 100, 110, 120}
$g_{\min}$	Minimum green, link / other phases (s)	20 / 10
$G_{\min}$	Total minimum green of the four phases (s)	50
g_s	Green split of the link	$(20 + S)/C$, S a spacing of $C - G_{\min}$
λ_c	Saturation flow rate (veh/s/lane)	0.5
a	Route demand (veh/s)	Uniform(0.45, 0.9) $\times$ share of $n\lambda_c g_s$
v_{ff}	Free-flow reference for delay (m/s)	15.4
v_{stop}	Stop threshold (m/s)	0.2
p	Probe penetration rate	0.3
–	Detector rows, distance to stop bar (m)	0, 8, 48, 56

C Model and sampling settings

Table 4 lists the settings of the prior. It is trained for 600 epochs with batch size 128, a linear warm-up followed by cosine decay of the learning rate from 10^{-4} to 10^{-5}, gradient clipping at 1.0, bfloat16 precision, and an exponential moving average of the weights. The noise schedule is sigmoid with $T = 1000$, and $\gamma_{\text{snr}} = 5$. The held-out loss did not change between epochs 400 and 600. The supervised network of Section 4.4 is the same U-Net with six data channels in, the detector state and its mask and the probe state and its mask, and two out, with the time-step input held at zero and the signal entering through the same two paths. It is trained on the same windows for 120

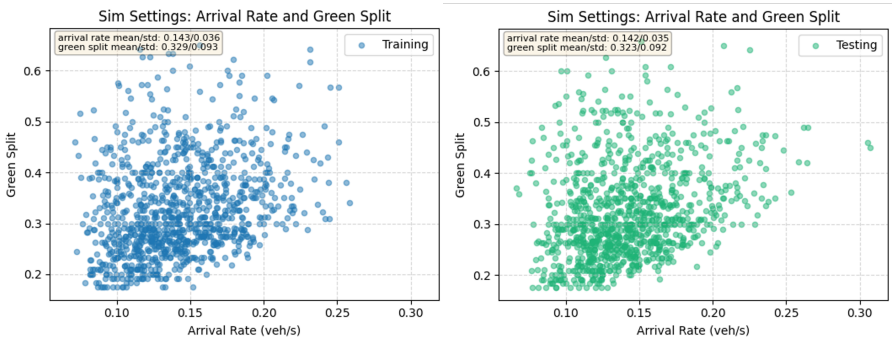

Figure 4: Arrival rate and green split of the training and test simulations.

epochs with batch size 128, AdamW at a learning rate of 2×10^{-4} with warm-up and cosine decay to 2×10^{-5}, gradient clipping at 1.0, bfloat16 precision, and an exponential moving average of the weights. The loss is the mean squared error after undoing the square-root map, that is in physical units up to the scale of each channel. A mean taken in the square-root scale squares back to less than the mean state, and a first version trained in that scale underestimated the arrival rate by up to 45% while reaching the same correlations. One network is trained for each detector layout of Table 2, and each takes about 95 minutes on one L40S.

Table 4: Data, model, and sampling settings

Setting	Value
Simulations for training / test	1024 / 32 (8 windows each)
Duration of each simulation (s)	6144
Δt / Δs	3 s / 8 m
Grid $H \times W$	32×32
Input channels	2 (density, flow)
Normalization	square-root min–max to $[-1, 1]$
U-Net channel widths	128, 256, 384
Signal conditioning	FiLM on time embedding + 4 tiled channels
Prediction target / loss weight	velocity / min-SNR with $\gamma_{\text{snr}} = 5$
Noise schedule / T	sigmoid / 1000
Batch size / epochs (used)	128 / 600 (400)
Learning rate	warm-up, cosine $10^{-4} \to 10^{-5}$
Gradient clipping / precision / EMA	1.0 / bfloat16 / yes
Sampling steps	1000 (DDIM)
Guidance step γ	0.5
α_{det}	0.2 (both sensors) / 0.1 (detectors only)
α_{traj} / ξ	1.0 / 2
Guidance window	$0.05\,T < t < 0.95\,T$
Post-processing cutoff	0.005 veh/m, 0.005 veh/s

D BOOTSTRAP INTERVALS AND CALIBRATION

Table 5 gives 95% bootstrap intervals, from 2000 resamples of the 256 test windows, for the correlations of Table 2. For the counterfactual of Section 4.3, resampling the 32 simulations gives 0.75 to 0.97 for the queue correlation of 0.93 and 0.60 to 0.94 for the delay correlation of 0.84; the model's mean queue change of 4.9 m has an interval of 2.6 to 7.8 m against 3.3 to 7.2 m for the simulator's 5.1 m. For the nearest training window keyed on the same upstream detectors, the queue correlation of 0.72 has an interval of 0.38 to 0.85 and its mean change of 4.4 m one of 2.5 to 6.4 m. For the supervised network, the correlation of 0.95 has an interval of 0.81 to 0.98 and its mean change of 4.9 m one of 3.4 to 6.9 m.

Table 6 reports the calibration of the sample spread on the first 32 test windows with eight samples each, drawn with different initial noise. Coverage is the share of windows whose ground truth lies inside the central 80% interval of the eight samples. The prior alone is close to nominal. Stop-bar guidance narrows the intervals by a factor of about two and lowers coverage to 66 to 72%, so the guided spread understates the remaining uncertainty; its width still tracks the error, with a correlation of 0.57 between spread and absolute error for the queue. The median of the eight samples is a better estimator than a single sample, with a queue correlation of 0.78 against 0.74 in Table 2.

Table 5: Correlations of Table 2 with 95% bootstrap intervals over windows.

Detectors	Guidance	Arrival	Queue	Delay
–	none (prior only)	-0.02 [-0.16, 0.12]	0.21 [0.08, 0.33]	0.29 [0.19, 0.40]
–	trajectories only	0.34 [0.22, 0.46]	0.69 [0.57, 0.79]	0.57 [0.47, 0.69]
0, 8, 48, 56 m	count baseline	0.89 [0.86, 0.92]	0.55 [0.45, 0.65]	–
0, 8, 48, 56 m	nearest training window	0.68 [0.60, 0.75]	0.76 [0.67, 0.82]	0.64 [0.57, 0.71]
0, 8, 48, 56 m	supervised U-Net	0.95 [0.93, 0.96]	0.97 [0.96, 0.98]	0.96 [0.94, 0.97]
0, 8, 48, 56 m	detectors	0.84 [0.79, 0.88]	0.90 [0.87, 0.93]	0.71 [0.65, 0.78]
0, 8, 48, 56 m	detectors + probes	0.77 [0.71, 0.82]	0.91 [0.87, 0.94]	0.68 [0.53, 0.83]
0, 8 m	count baseline	0.68 [0.60, 0.75]	0.51 [0.39, 0.61]	–
0, 8 m	nearest training window	0.59 [0.50, 0.66]	0.67 [0.60, 0.74]	0.64 [0.57, 0.72]
0, 8 m	supervised U-Net	0.79 [0.73, 0.84]	0.78 [0.70, 0.85]	0.83 [0.77, 0.87]
0, 8 m	detectors	0.67 [0.59, 0.73]	0.74 [0.68, 0.80]	0.58 [0.49, 0.68]
0, 8 m	detectors + probes	0.62 [0.52, 0.71]	0.80 [0.71, 0.88]	0.73 [0.67, 0.79]
48, 56 m	count baseline	0.89 [0.85, 0.92]	0.55 [0.45, 0.65]	–
48, 56 m	nearest training window	0.62 [0.54, 0.70]	0.71 [0.61, 0.78]	0.50 [0.38, 0.61]
48, 56 m	supervised U-Net	0.95 [0.93, 0.96]	0.96 [0.94, 0.97]	0.94 [0.93, 0.96]
48, 56 m	detectors	0.80 [0.75, 0.85]	0.83 [0.76, 0.87]	0.53 [0.43, 0.63]

Table 6: Calibration of the sample spread, 32 windows with eight samples each. Coverage of the central 80% interval, its mean width, the median absolute error of the eight-sample median, and the correlation between spread and absolute error.

Guidance	Quantity	Coverage	Mean width	Median abs. error	Corr. spread, error
none (prior only)	arrival rate (veh/s)	0.78	0.102	0.030	0.13
none (prior only)	max queue (m)	0.81	26.3	6.6	0.05
none (prior only)	delay (s)	0.72	21.5	5.2	-0.02
stop-bar detectors	arrival rate (veh/s)	0.72	0.061	0.020	0.33
stop-bar detectors	max queue (m)	0.66	14.9	2.9	0.57
stop-bar detectors	delay (s)	0.69	15.2	4.4	0.11

E Out-of-distribution cycle lengths

F Forecasting query

Table 8 scores the forecasting query of Section 4.4. The guidance weight of DiffLink was chosen on 256 held-out windows of the training distribution, windows 3 to 10 of 32 further simulations. On the first 128 of them the weights 0.1, 0.2 and 0.4 all gave a queue correlation of 0.90 on the hidden part. The error in hidden cells fell with the weight, but so did the forecast arrival rate, from 0.140 to 0.118 veh/s against 0.138, so the largest weight with an arrival bias within 5%, 0.2, is used on the test set. Eight samples per window with different initial noise give a median whose queue correlation is 0.92 and a central 80% interval that covers the true queue in 70% of windows, with a mean width of 8.8 m and a correlation of 0.41 between the spread and the absolute error of the median.

The graph network simulator follows Linkerhägner et al. (2023). The 32 cells of the link are nodes joined to their neighbours, each carrying its density and flow, its position, and the green seconds of

Table 7: Out-of-distribution cycle lengths, 128 windows each. The 80–120 s row is the in-distribution test set. Each cell is test / generated. Arrival rate is in veh/s with its KS distance in parentheses, where the 5% critical value for 128 windows is 0.17. FD peak and discharge flow are in veh/s and queue growth in m/s, as in Figure 8. Switch is the stop-bar flow in the 6 s after the green starts over the flow in the 6 s before, pooled over all switches. The last column is the queue correlation with four detectors for the generating model / deterministic queuing. The upper block is DiffLink, the lower block the lookup table of Section 4.2.

Cycle	Arrival (KS)	FD peak	Discharge	Growth	Switch	Queue corr.
60 s	0.179 / 0.158 (0.23)	0.72 / 0.73	0.339 / 0.341	0.55 / 0.48	23.2 / 19.3	0.68 / 0.47
80–120 s	0.149 / 0.148 (0.07)	0.67 / 0.69	0.482 / 0.480	0.50 / 0.51	23.3 / 21.6	0.90 / 0.55
140 s	0.119 / 0.141 (0.31)	0.72 / 0.69	0.517 / 0.471	0.43 / 0.48	16.2 / 22.6	0.94 / 0.66
60 s	0.179 / 0.132 (0.49)	0.72 / 0.71	0.339 / 0.352	0.55 / 0.44	23.2 / 11.1	0.42 / 0.47
80–120 s	0.149 / 0.143 (0.14)	0.67 / 0.69	0.482 / 0.448	0.50 / 0.49	23.3 / 22.5	0.76 / 0.55
140 s	0.119 / 0.151 (0.35)	0.72 / 0.70	0.517 / 0.485	0.43 / 0.50	16.2 / 25.1	0.84 / 0.66

the coming step; the same message-passing network as in the original work, five blocks of width 128 and 0.43 M parameters, predicts the change of density and flow of every cell over one 3 s step and adds it to the state, as GGNS does for mesh positions. It is trained on the same 1024 runs for 200 epochs with Gaussian noise added to the input state, a standard deviation of 0.01 chosen on held-out runs, and the checkpoint chosen by the rollout error on those runs. For the forecast it starts from the true column 15 and feeds its own predictions back for 16 steps with no observation nodes. It keeps the queue but not the demand: with a standard deviation of 0.05 its queue correlation is 0.90 and its forecast arrival rate 0.003 veh/s. The same collapse of the arrival rate appears in every rollout of this network, with or without detector observations, because nothing in a step-by-step update supplies vehicles at the upstream boundary. Sensor nodes at the detector rows do not help it either. The 48 and 56 m rows are 22 cells from the entrance, and five message-passing blocks per step cannot carry a reading that far within a step, while across steps the trained network shows no measurable response at the entrance.

Table 8: Forecast of the last 48 s of each test window from its first 48 s and the signal plan. Correlations are with the truth of the hidden part over 256 windows, with 95% bootstrap intervals, and the delay over the 253 windows with at least 0.03 veh/s of arrivals in the hidden part. Persistence repeats the value of the observed part. Deterministic queuing is the observed arrival rate times the longest red of the hidden part over two lanes at 8 m per vehicle. The nearest training window is keyed on the signal and the observed part. The supervised U-Net is the network of Section 4.4 trained for this mask with the same loss in physical units. The graph network simulator is GGNS (Linkerhägner et al., 2023) adapted to the link, rolled forward from the last observed column with the signal alone. Both return one state per window, so they have no spread to report. The arrival rate of the graph network is 0.027 veh/s against 0.147 in truth.

Method	Queue (m)	Arr. corr.	Queue corr.	Delay corr.
Truth of the hidden part	27.3			
Persistence	26.0	0.23 [0.10, 0.35]	0.21 [0.07, 0.33]	-0.22 [-0.31, -0.15]
Deterministic queuing	18.1	0.23 [0.11, 0.36]	0.38 [0.25, 0.49]	–
Nearest training window	24.8	0.14 [0.03, 0.26]	0.71 [0.63, 0.77]	0.64 [0.51, 0.78]
Supervised U-Net	21.4	0.36 [0.24, 0.46]	0.90 [0.88, 0.93]	0.81 [0.76, 0.87]
Graph network simulator	23.8	0.06 [-0.06, 0.21]	0.87 [0.84, 0.90]	0.76 [0.71, 0.83]
DiffLink	27.3	0.15 [0.02, 0.27]	0.89 [0.85, 0.92]	0.58 [0.47, 0.74]

G SENSORS AND A RECONSTRUCTION EXAMPLE

Figure 5 shows the two sensors of Section 3.3 on one time-space diagram, and Figure 6 one window reconstructed from both.

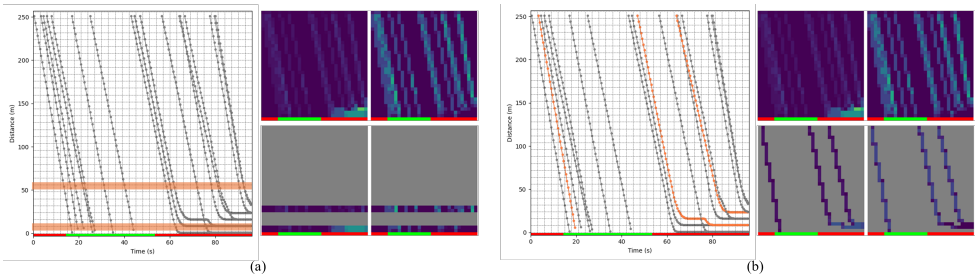

Figure 5: One time-space diagram under two sensors. (a) Gray dotted lines are all trajectories. The orange bands are loop detectors, which give exact flow and density in their rows. (b) The orange dotted line is one probe trajectory. Several vehicles can share a cell, so the state built from probes alone is a lower bound on the truth in the cells it touches. Gray marks unobserved cells.

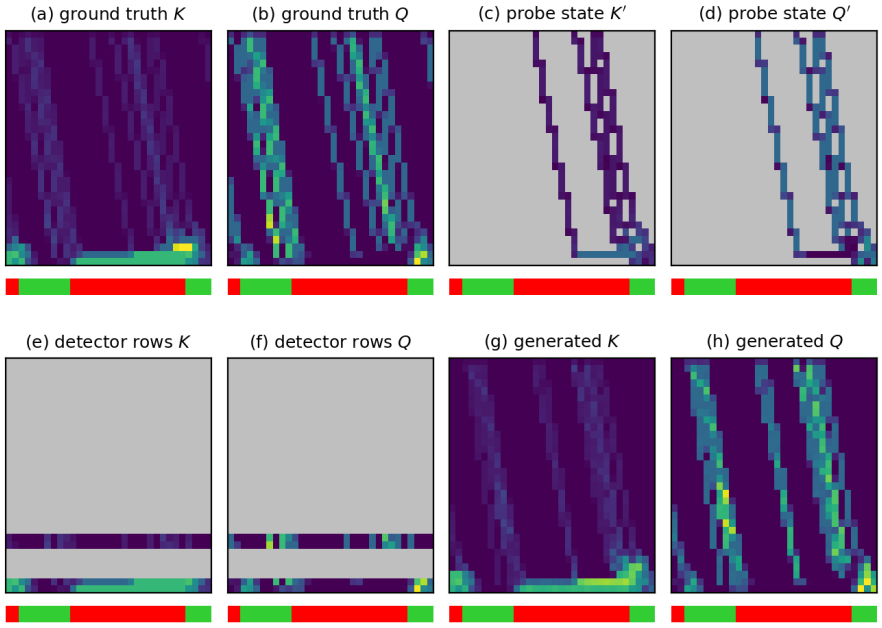

Figure 6: Reconstruction from two sensors. (a, b) Ground-truth density K and flow Q. (c, d) The partial state built from probe trajectories. (e, f) The rows observed by loop detectors. (g, h) The state generated with both sensors as guidance. Gray marks unobserved cells in (c)–(f).

H PHYSICS PROBES

Figure 7 shows the joint distributions behind Table 1, and Figure 8 the three probes discussed in Section 4.2.

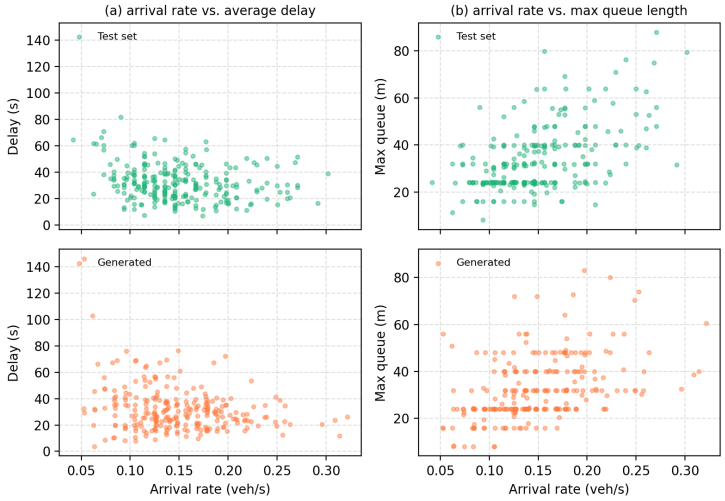

Figure 7: Metric distributions. (a) Joint distribution of arrival rate and average delay. (b) Joint distribution of arrival rate and maximum queue length. In each panel the top plot is the test set and the bottom plot is the set of generated states.

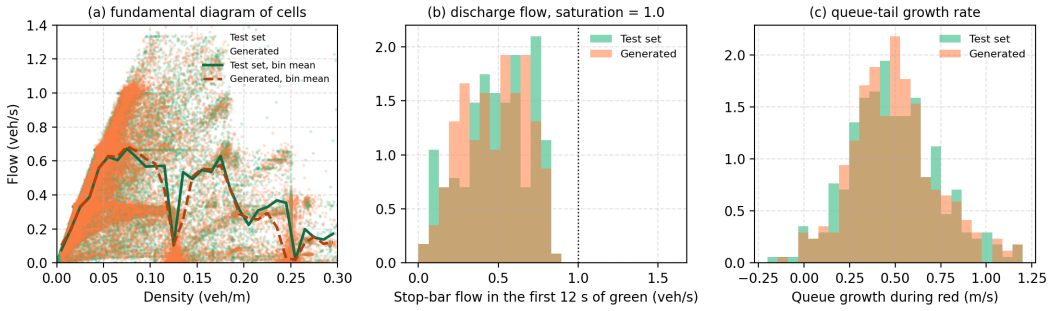

Figure 8: Physics probes on states generated from the signal alone. (a) Flow against density for every occupied cell of the 256 test windows and of the generated states, with binned means. (b) Stop-bar flow in the first 12 s of green after a red of at least 15 s. The dotted line is the two-lane saturation flow. (c) Growth rate of the queue tail during red phases of at least 24 s.