DETNO: A Diffusion-Enhanced Transformer Neural Operator for Long-Term Traffic Forecasting

Owais Ahmad[1], **Milad Ramezankhani**[1], **Anirudh Deodhar**[1]

[1]Applied Research, Quantiphi
Marlborough, MA 01752, USA
{owais.ahmad, milad.ramezankhani, anirudh.deodhar}@quantiphi.com

Abstract

Accurate long-term traffic forecasting remains a critical challenge in intelligent transportation systems, particularly when predicting high-frequency traffic phenomena such as shock waves and congestion boundaries over extended rollout horizons. Neural operators have recently gained attention as promising tools for modeling traffic flow. While effective at learning function space mappings, they inherently produce smooth predictions that fail to reconstruct high-frequency features such as sharp density gradients which results in rapid error accumulation during multi-step rollout predictions essential for real-time traffic management. To address these fundamental limitations, we introduce a unified Diffusion-Enhanced Transformer Neural Operator (DETNO) architecture. DETNO leverages a transformer neural operator with cross-attention mechanisms, providing model expressivity and super-resolution, coupled with a diffusion-based refinement component that iteratively reconstructs high-frequency traffic details through progressive denoising. This overcomes the inherent smoothing limitations and rollout instability of standard neural operators. Through comprehensive evaluation on chaotic traffic datasets, our method demonstrates superior performance in extended rollout predictions compared to traditional and transformer-based neural operators, preserving high-frequency components and improving stability over long prediction horizons.

Introduction

Precise traffic forecasting is essential for effective transportation system management, particularly as urbanization accelerates and mobility needs continue to grow (Lana et al. 2018). Traffic modeling, however, faces significant challenges due to sparse data availability and the chaotic, nonlinear dynamics of traffic flow that include sudden transitions, congestion formation, and shockwave propagation, making accurate prediction exceptionally difficult (Alghamdi et al. 2022; Smith and Demetsky 1997). Conventional traffic data collection methods (e.g., loop detectors, cameras and probe vehicles) offer valuable insights within their coverage areas. However, real-world data has inherent limitations for inverse problems and optimization tasks that require controlled conditions, systematic parameter variation, or counterfactual analysis (Jain, Sharma, and Subramanian 2012; Shafik and Rakha 2025). Many critical traffic engineering problems therefore rely on using numerical solvers to simulate traffic flow and generate synthetic data, providing the reproducible environments needed for advanced analysis. Models based on partial differential equations, such as the Lighthill-Whitham-Richards (LWR) model (Lighthill and Whitham 1955), capture key dynamics like shockwave propagation and congestion, but require computationally intensive schemes (e.g., Godunov) with strict discretization and stability constraints. Second-order models like Aw-Rascle-Zhang (ARZ) (Aw and Rascle 2000) handle non-equilibrium conditions but further increase complexity, limiting their use for real-time city-wide traffic management.

To address these computational and data sparsity challenges, machine learning (ML) models have emerged as a promising solution. These approaches can learn complex traffic patterns from available data while being more computationally efficient than traditional numerical solvers for real-time applications. Importantly, ML-based solutions offer significant advantages for large-scale deployment, including the ability to process multiple traffic scenarios in parallel and adapt to varying urban infrastructure without requiring extensive recalibration for each new deployment site. ML-based approaches such as Graph Neural Networks (GNNs) model traffic networks as graphs, with nodes as road segments and edges as connectivity (Yu, Yin, and Zhu 2017; Peng et al. 2020). Advanced variants like Graph Attention Networks (GATs) (Zhang, James, and Liu 2019; Kong et al. 2020; Wang et al. 2022) have demonstrated superior performance in capturing non-linear spatial correlations and temporal dynamics through attention mechanisms. However, these models heavily rely on data availability and can suffer from poor generalization in new traffic scenarios. Scientific ML approaches such as Physics-Informed Neural Networks (PINNs) (Raissi, Perdikaris, and Karniadakis 2019) have recently shown significant promise in learning traffic flow dynamics (Shi, Mo, and Di 2021; Usama et al. 2022). They embed traffic flow physics into the learning process, enabling robust data-agnostic modeling. However, their poor domain generalization in new initial/boundary conditions and high computational cost for enforcing physics constraints hinder deployment in real-time, city-scale traffic management. Neural operators have been introduced to address the generalization limitations

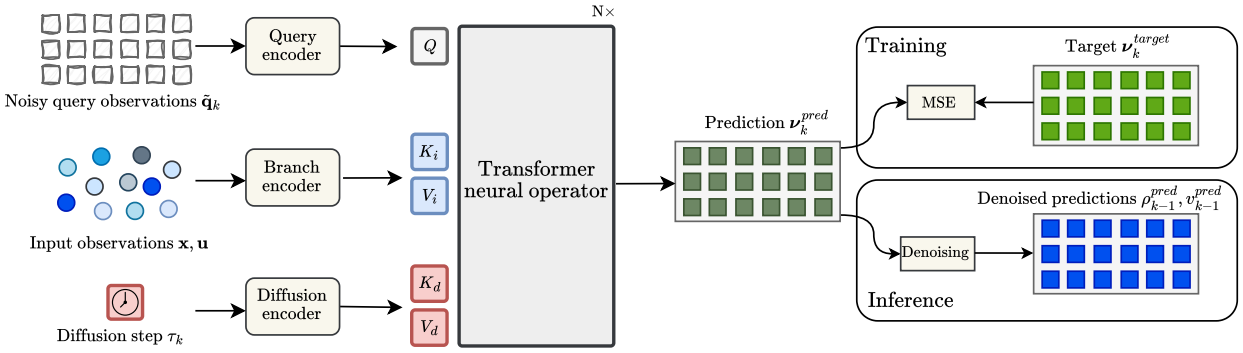

Figure 1: DETNO architecture for traffic forecasting. Noisy query tokens $\tilde{\mathbf{q}}_k = [x, t, \tilde{\rho}_k, \tilde{v}_k]$ are processed by the *Query encoder*; sensor observations $[\mathbf{x}, \mathbf{u}]$ (coordinates and measurements) are processed by the *Branch encoder*; and the diffusion timestep τ_k (Fourier-embedded) is processed by the *Diffusion encoder*. Their outputs form a query stream $\mathbf{Q}$ and two context streams, operator stream $(\mathbf{K}_i, \mathbf{V}_i)$ and diffusion stream $(\mathbf{K}_d, \mathbf{V}_d)$, which a transformer neural operator processes via heterogeneous cross-attention (followed by self-attention). At step k, the model predicts the diffusion velocity $\boldsymbol{\nu}_k^{pred}$. During training, a v-parameterization loss minimizes MSE to the target $\boldsymbol{\nu}_k^{target}$; during inference, $\boldsymbol{\nu}_k^{pred}$ drives a DDIM update to recover $(\rho_{k-1}^{pred}, v_{k-1}^{pred})$ in a $k \to k-1$ denoising schedule, yielding the final predictions.

of PINNs (Lu, Jin, and Karniadakis 2019; Li et al. 2020). Models such as DeepONets (Rap and Das 2025), which learn mappings between function spaces, Fourier Neural Operators (FNOs) (Thodi, Ambadipudi, and Jabari 2024), which capture traffic dependencies in the frequency domain, and Variable-Input Deep Operator Networks (VIDON) (Prasthofer, De Ryck, and Mishra 2022), which handle irregular and heterogeneous network layouts, have been successfully applied to traffic forecasting. These methods generalize across diverse road conditions, sensor configurations, and network densities without retraining by learning fundamental traffic flow governing laws. However, neural operators suffer from spectral bias (Moseley, Markham, and Nissen-Meyer 2023; Ramezankhani et al. 2025a), favoring low-frequency components and producing overly smooth predictions that fail to capture high-frequency phenomena such as shockwaves, and abrupt congestion transitions critical for effective traffic management. This limitation can be detrimental in long temporal rollouts, leading to error accumulation and significance divergence from ground truth.

To mitigate the spectral bias of neural operators, two diffusion-based families have emerged: two-stage and single-stage strategies. The former trains a neural operator first, and then a score-based diffusion model is conditioned on the neural operator's outputs to restore high-frequency detail and improve spectral alignment (Oommen et al. 2025; Perrone et al. 2025; Guo et al. 2025). While it is able to match the ground-truth spectrum, its added fine-scale detail can be partially *hallucinated* and *non-physical*, leading to limited gains in pointwise errors like mean squared error (MSE) and extra inference cost from multiple denoising steps. The second appoach leverages diffusion-inspired multistep denoising within the neural operator to iteratively reweight non-dominant (i.e., high-frequency) components and improve long-horizon stability (Lippe et al. 2023; Serrano et al. 2024). However, current realizations typically rely on conventional backbones (e.g., UNet and FNO) operating on fixed and regular grids, which can constrain long-range and multi-scale spatiotemporal modeling and limit portability across real-world heterogeneous geometries.

To address the aforementioned limitations, we introduce the Diffusion-Enhanced Transformer Neural Operator (DETNO), an end-to-end architecture that couples a transformer neural operator with a diffusion refiner *within one unified model*. DETNO leverages a heterogeneous cross-attention module that maintains two distinct information streams: (i) an *operator stream* whose keys/values encode input functions (e.g., sensor fields and boundary/initial conditions), and (ii) a *diffusion stream* whose keys/values encode the diffusion noise level/timestep. Each query derived from spatiotemporal coordinates is *conditioned by* both K/V sets, ensuring both the input functions and diffusion schedule influence the predictions in a controlled manner. The DETNO architecture also enables super-resolution, allowing to query at any arbiterory resolution in the traffic spatiotemporal domain. The integrated diffusion refiner performs a small number of iterative denoising steps to reconstruct fine-grained structure, recovering high-frequency phenomena with a modest computational overhead. The main contributions of this paper are twofold: (1) we introduce a novel diffusion-enhanced neural operator architecture that explicitly addresses error accumulation over long temporal rollouts in traffic forecasting; and (2) we demonstrate that the method significantly outperforms neural operator baselines by effectively capturing high-frequency traffic dynamics and sharp transitions that conventional approaches typi-

cally smooth out.

Methodology

Transformer Neural Operator

As illustrated in Figure 1, our transformer neural operator comprises a heterogeneous cross-attention block followed by self-attention. The cross-attention exposes two context streams as keys/values: an operator stream derived from input functions and a diffusion stream derived from the diffusion timestep; queries contain spatiotemporal coordinates at which we predict traffic states. All three inputs are first mapped by dedicated encoders: a Query encoder, a Branch encoder (operator stream), and a Diffusion encoder. Each encoder is an multi-layer perceptron (MLP) that projects its input into a d-dimensional latent; the Diffusion encoder additionally applies a Fourier embedding to the timestep before the MLP. For each query, we compute linear cross-attention (Hao et al. 2023) separately against the operator and diffusion streams to produce two context vectors that are then fused (summation and projection with a residual) into an updated query representation; a subsequent self-attention layer lets queries exchange information and enforce spatial–temporal coherence. We use a Mixture-of-Experts (MoEs) in the transformer blocks, with a gating network conditioned on each query's spatiotemporal coordinates to realize a soft domain decomposition that routes tokens to specialized experts (Hao et al. 2023; Ramezankhani et al. 2025b).

The training data consist of two sets with different cardinalities: sensors $\{(\mathbf{x}^i, \mathbf{u}^i)\}_{i=1}^{N_{\text{sensor}}}$ and queries $\{(\mathbf{q}^j, \mathbf{y}^j)\}_{j=1}^{N_{\text{pred}}}$, where $\mathbf{x}^i \in \mathbb{R}^2$ are space–time coordinates (x, t), $\mathbf{u}^i \in \mathbb{R}^2$ are traffic sensor states (ρ, v), $\mathbf{q}^j \in \mathbb{R}^4$ are query tokens $[x_q, t_q, \rho, v]$, and $\mathbf{y}^j \in \mathbb{R}^2$ are ground-truth states at the same query locations. The encoders map inputs to width d as follows: the Query encoder $\phi_q : \mathbb{R}^4 \to \mathbb{R}^d$ applies an MLP to yield $\mathbf{Q} \in \mathbb{R}^{N_{\text{pred}} \times d}$; the Branch encoder $\phi_b : \mathbb{R}^4 \to \mathbb{R}^d$ (MLP) acts on $[\mathbf{x}^i, \mathbf{u}^i]$ to produce operator-stream keys/-values $(\mathbf{K}_i, \mathbf{V}_i) \in \mathbb{R}^{N_{\text{sensor}} \times d}$; the Diffusion encoder maps the diffusion timestep $\tau \in \mathbb{R}$ through a Fourier embedding $\gamma(\tau) \in \mathbb{R}^{d_\tau}$ and an MLP to $z_d \in \mathbb{R}^d$, which is broadcast across queries to form $(\mathbf{K}_d, \mathbf{V}_d) \in \mathbb{R}^{N_{\text{sensor}} \times d}$. Linear cross-attention is computed separately against the operator and diffusion streams,

$$\mathbf{C}_i = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}_i, \mathbf{V}_i), \quad \mathbf{C}_d = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}_d, \mathbf{V}_d). \quad (1)$$

The outputs are then fused and passed through linear self-attention and a feed-forward block to produce a diffusion velocity fields $\hat{\boldsymbol{\nu}} \in \mathbb{R}^{N_{\text{pred}} \times 2}$ (for both traffic density and velocity) at the query coordinates. The composition of $\mathbf{q}^j$ differs between training and inference: in training, its (ρ, v) entries are noise-corrupted versions of $\mathbf{y}^j$; at inference, they are initialized with pure noise and refined by the diffusion process, allowing a single query format for both supervised learning and test-time denoising.

Diffusion-Based Refinement

The diffusion refiner learns to remove injected noise from traffic states at query locations using a velocity-parameterization objective. During training, for a noise level $k \in \{0, \dots, K\}$, we draw $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and form the corrupted targets $\tilde{\mathbf{y}}_k = \sqrt{\bar{\alpha}_k}\,\mathbf{y} + \sqrt{1 - \bar{\alpha}_k}\,\boldsymbol{\epsilon}$, where $\mathbf{y}$ is the clean state (density, velocity) at the query locations and $\bar{\alpha}_k$ is the cumulative noise schedule (Song, Meng, and Ermon 2020). The noisy query tokens are generated as $\tilde{\mathbf{q}}_k = [x_q, t_q, \tilde{\rho}_k, \tilde{v}_k]$, with $(\tilde{\rho}_k, \tilde{v}_k)$ taken *directly* from the two channels of $\tilde{\mathbf{y}}_k$ at the same query coordinates. Conditioned on $(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{q}}_k, \tau_k)$, the DETNO model $\mathcal{G}_\theta$ predicts the diffusion velocity $\boldsymbol{\nu}_k$. The supervision target uses the standard v-parameterization

$$\boldsymbol{\nu}_k^* = \sqrt{\bar{\alpha}_k}\,\boldsymbol{\epsilon} - \sqrt{1 - \bar{\alpha}_k}\,\mathbf{y}, \quad (2)$$

and the diffusion loss for a single diffusion process is

$$\mathcal{L}_{\text{diffusion}} = \sum_{k=0}^{K} \mathbb{E}_{k,\boldsymbol{\epsilon}} \left\| \mathcal{G}_\theta(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{q}}_k, \tau_k) - \boldsymbol{\nu}_k^* \right\|_2^2. \quad (3)$$

At inference, we initialize the traffic state entries with pure noise (i.e., ρ_K and v_K), $\mathbf{q}^{(K)} = [x_q, t_q, \rho_K, v_K]$, and iteratively denoise over $k = K, \dots, 0$. At step k, the model consumes $(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{q}}_k, \tau_k)$, outputs $\hat{\boldsymbol{\nu}}^{(k)}$, and a DDIM update produces ρ_{k-1}^{pred} and v_{k-1}^{pred}. DDIM is preferred over DDPM for its deterministic updates that permit larger step sizes and fewer evaluations while preserving quality, yielding faster denoising without changing the training loss (Song, Meng, and Ermon 2020). It is crucial to distinguish the physical time t_q in the query coordinates from the denoising timestep τ used by the diffusion stream. This timestep tells the model how much noise is present in the current prediction, acting as a progress indicator during the refinement process. At each refinement step, the current diffusion timestep is computed as $\tau_k = \text{scheduler\_timestep}(k) \cdot \frac{1000}{K}$. The resulting embedding $\gamma(\tau_k)$ (Fourier features followed by an MLP, as defined previously) is used to form the diffusion-stream keys/values $\mathbf{K}_d, \mathbf{V}_d$.

Case Study: Long-term Traffic Forecasting

To evaluate the proposed DETNO approach for traffic prediction, we establish a controlled simulation environment that models traffic flow dynamics over a spatiotemporal domain. The setup represents a highway segment with fixed sensors providing sparse observations of density and velocity at discrete spatiotemporal locations, mimicking real-world monitoring. The goal is to predict traffic states $\boldsymbol{u}(x, t) = [\rho(x, t), v(x, t)]^\top$ over $X := [x_{\min}, x_{\max}] \subset \mathbb{R}$ and $T \subset \mathbb{R}^+$, with current time $t_c \in T$ and windows $\Delta_{\text{past}}, \Delta_{\text{pred}} > 0$ such that $t_c - \Delta_{\text{past}}, t_c + \Delta_{\text{pred}} \in T$. For data generation, traffic density evolves according to the Lighthill–Whitham–Richards (LWR) model,

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial}{\partial x}\big[\rho(x,t)\, v(\rho(x,t))\big] = 0, \quad (x,t) \in X \times T, \quad (4)$$

where ρ is the density field and $v(\rho)$ is the fundamental diagram (velocity–density relation). Synthetic trajectories used for training and evaluation are produced with a first-order Godunov finite-volume scheme. Implementation details are provided in the Supplementary Information.

Results and Discussion

Experiment settings

Training Procedure and Temporal Rollout DETNO is trained in a *single-step* prediction procedure. It learns a supervised mapping from sparse sensor measurements in a past window $[t_c - \Delta_{\text{past}}, t_c]$ to traffic states at arbitrary query locations within the immediate future window $[t_c, t_c + \Delta_{\text{pred}}]$. In this work, both Δ_{past} and Δ_{pred} are set to 1 minute. During training, ground-truth fields are available over the full domain, so the network learns to reconstruct complete traffic states from the past sensor history for one horizon. Long-term forecasting is performed *only at inference* via an autoregressive rollout. The first prediction window uses the real sensor data. For each subsequent window, we form pseudo-sensor inputs by sampling the model's previous predictions at the fixed sensor coordinates (shifted forward in time by Δ_{pred}) and combining them with boundary-condition data; these inputs are then fed back into the model to predict the next window. Iterating this procedure extends forecasts over the desired temporal horizon. In this work, we examined the models performance for 8 rollout steps. We generated 1300 traffic simulations by varying initial and boundary conditions (e.g., initial vechicle density and velocity, and traffic-light settings at the end of the road); 1000 samples are used for training and 300 for testing. Details about the DETNO architecture are provided in Supplementary Information. The diffusion mechanism wraps the entire transformer neural operator within a denoising framework using DDIM scheduler. The wrapper manages noise scheduling through $K = 10$ refinement steps with a minimum noise standard deviation of 9×10^{-2}.

Baseline Models We compare DETNO against two primary baseline approaches. ON-Traffic (Rap and Das 2025) utilized an advanced DeepONet architecture that directly learns mappings between sensor measurements and traffic predictions. General Neural Operator Transformer (GNOT) (Hao et al. 2023) processes traffic data by leveraging transformer blocks and MoEs to model complex spatio-temporal traffic patterns. These baselines allow us to evaluate the effectiveness of our unified transformer-diffusion architecture against standard neural operator approaches and assess the contribution of the diffusion refinement mechanism in capturing hig-frequency features and minimizing the error accumulation over long rollouts.

DETNO Performance Analysis on Chaotic Traffic Data

Table 1 compares the models performance for a single step (step 1) and rollout (step 8) predictions. To evaluate the long rollout performance, the models' predictions at the 8th rollout step are compared. Our proposed DETNO approach achieves optimal rollout performance, demonstrating a 96.0% improvement in MSE and a 26.3% improvement in MAE compared to GNOT, the second best model. ONTraffic exhibits significant error accumulation, with MSE and MAE increasing by 30.53× and 7.10×, respectively, compared to a single-step prediction (step 1). GNOT shows improved stability, with 6.32× growth in MSE and 2.11× in MAE. In contrast, our proposed DETNO method achieves the most robust long-term performance, with only 4.39× increase in MSE and 1.34× in MAE going from step 1 to step 8 prediction. This superior rollout stability highlights DETNO's capacity to preserve fine-scale spatiotemporal features over extended prediction horizons. While the benefits of its diffusion-based refinement mechanism are modest in the initial rollout steps, its advantage becomes increasingly pronounced over time. As errors accumulate, models that fail to capture high-frequency components and sharp density gradients (such as ONTraffic and GNOT) experience rapid performance degradation. DETNO, by contrast, effectively reconstructs these high-frequency features, maintaining temporal consistency and predictive accuracy even in complex and chaotic traffic scenarios.

To further demonstrate these rollout stability advantages, we visualized GNOT and DETNO models performance across extended prediction horizons, as illustrated in Figure 2. The models predictions are visualized through multiple perspectives:(1) predicted density distributions, (2) ground truth density distributions, (3) absolute error heatmaps, (4) spatial density profiles at $t = 5$ minutes, and (5) frequency spectrum comparisons. The absolute-error heatmaps show that DETNO captures high-frequency structure more faithfully, yielding smaller errors along sharp transition regions (e.g., congestion fronts). In contrast, GNOT exhibits error growth over time: as the rollout progresses, both the magnitude and the spatial regions of its errors increase, consistent with missing high-frequency content early on and compounding mismatch in later steps. The spatial density profiles echo this trend: DETNO remains closely aligned with ground truth around highly nonlinear segments, whereas GNOT produces visibly smoothed transitions and local biases near discontinuities. Most critically, the frequency-spectrum comparison reveals that GNOT underestimates energy at higher wavenumbers (deviating from the ground-truth slope and amplitude), while DETNO tracks the spectrum across scales, including the high wavenumber regime. This alignment indicates that DETNO learns and preserves high-frequency features, which in turn stabilizes long-horizon rollouts and reduces error accumulation.

Impact of Diffusion-Based Refinement

Figure 3 demonstrates why DETNO sustains realistic traffic dynamics over long horizons. In panel (a), the frequency spectra averaged over 300 test rollouts show that DETNO closely follows the ground-truth curve across scales, including the high-wavenumber regime that encodes sharp density fronts and rapidly varying congestion. By contrast, GNOT and ONTraffic exhibit a premature roll-off as wavenumber increases, indicating systematic underestimation of fine-scale energy and explaining their softened transition zones. Panel (b) reports MSE by rollout step: errors increase for all methods with horizon length, but DETNO maintains both the lowest magnitude and the shallowest growth rate; ONTraffic is worst at all steps, and GNOT lies between ONTraffic and DETNO yet diverges more quickly than DETNO. The widening gap over time is consistent with the spectral finding: missing high-frequency content at early steps

Table 1: Single step and rollout performance comparison of our proposed DETNO model against ONTraffic (Rap and Das 2025) and GNOT (Hao et al. 2023) on the Godunov dataset. Mean squared error (MSE) and mean absolute error (MAE) are used as comparison metrics.

Rollout Stage	Metric	ONTraffic (Rap and Das 2025)	GNOT (Hao et al. 2023)	**DETNO (ours)**
Step 1	Avg. MSE	0.009	0.003	0.002
	Avg. MAE	0.038	0.018	0.022
Step 8	Avg. MSE	0.279	0.019	0.008
	Avg. MAE	0.272	0.038	0.030
Model Size		1.40M	1.13M	1.16M

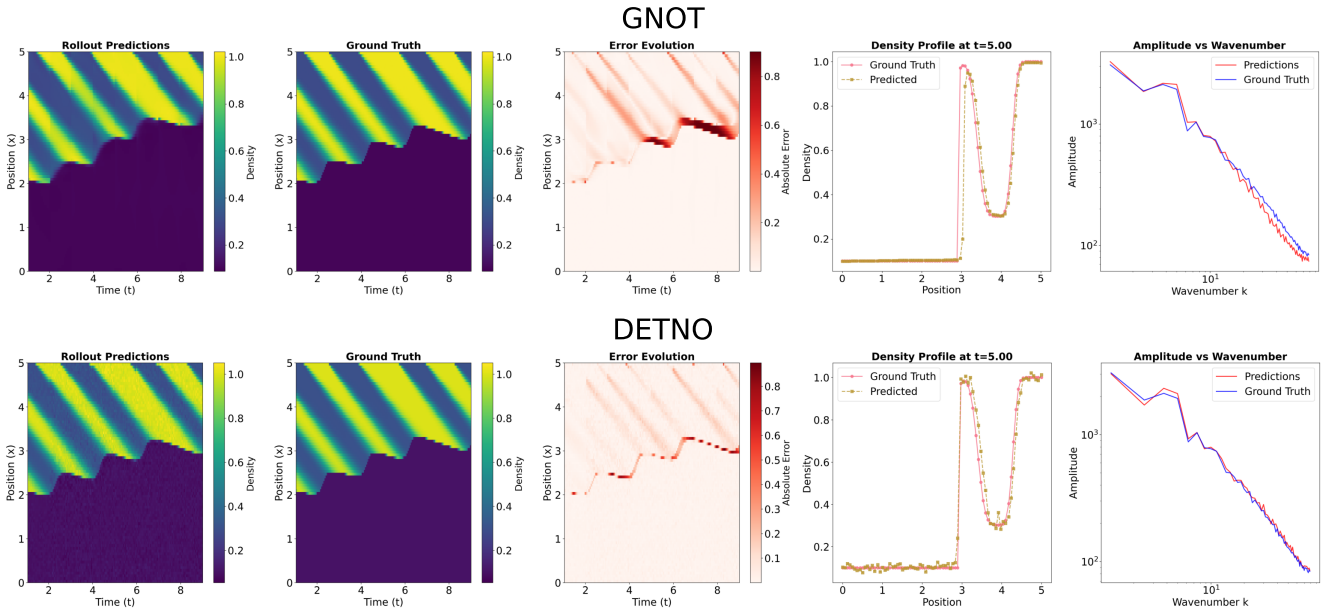

Figure 2: Comparative analysis of GNOT and DETNO predictions on chaotic traffic scenarios. The visualization shows ground truth density fields, predicted density distributions, absolute error maps, spatial density profiles at t=5.00, and frequency spectrum comparisons for two representative samples, demonstrating the refinement mechanism's superior reconstruction of sharp density transitions and localized traffic phenomena.

compounds under autoregressive reuse, whereas DETNO's fidelity at high wavenumbers slows error accumulation and preserves coherent traffic patterns deeper into the rollout.

Ablation Studies

We conducted comprehensive ablation studies to determine the optimal hyperparameter setting for our proposed DETNO architecture as elaborated below (Figure 4).

Hidden dimension. At 64 hidden units the network achieves the lowest error (MSE 0.0080), whereas reducing capacity to 32 underfits the dynamics (MSE 0.0142) and increasing to 128 degrades further (MSE 0.0215). The latter suggests over-parameterization in this data regime and less stable expert routing, leading to poorer generalization.

Number of experts. Three experts provide the strongest balance between specialization and data fragmentation (MSE 0.0050). With two experts, capacity is insufficient for learning heterogeneous traffic regimes (MSE 0.0066). Adding more experts yields diminishing or negative returns: four experts markedly worsen performance (MSE 0.0080), and five experts only partially recover (MSE 0.0060). These results are consistent with MoE load-balancing effects, where increasing the expert count reduces per-expert sample density and makes routing harder to optimize (Fedus, Zoph, and Shazeer 2022).

Minimum noise (diffusion floor). A "Goldilocks" level of corruption is required for effective denoising. The best setting is 9×10^{-2} (MSE 0.0050). Lower noise at 7×10^{-2} or 8×10^{-2} weakens the learning signal and raises error (MSE 0.0103/0.0100), while a higher floor at 1×10^{-1} overcorrupts targets and again increases error (MSE 0.0099). The chosen level provides enough perturbation to teach robust corrections without losing key structure.

Refinement steps (DDIM). We observed that multi-step refinement is essential. A single step is inadequate (MSE 0.0131), five steps capture most of the gains (MSE 0.0054),

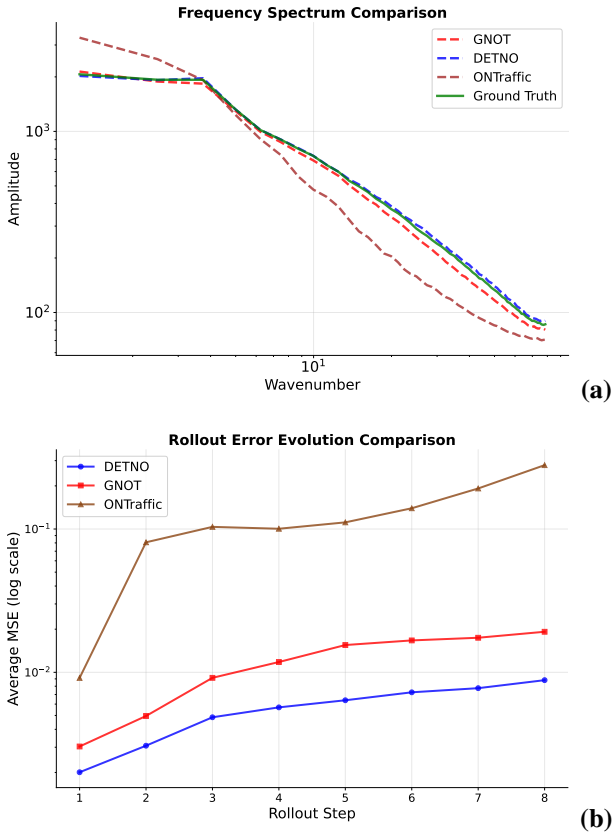

Figure 3: Performance comparison between DETNO, GNOT and ONTraffic: (a) Averaged frequency spectrum across 300 samples demonstrating superior high-frequency preservation by DETNO, with enhanced amplitude retention across all wavenumbers indicating effective recovery of sharp density gradients and discontinuous traffic patterns. (b) Step-wise rollout error evolution showing DETNO's improved stability and accuracy over extended prediction horizons compared to GNOT and ONTraffic.

and ten steps deliver the best overall accuracy (MSE 0.0050). The improvement from five to ten steps is modest, reflecting gradual restoration of fine-scale structure; beyond ten steps, additional latency is unlikely to be justified by further gains.

Cross-attention design. Using *two streams* for keys/values—an operator stream for input functions and a diffusion stream for the denoising timestep—outperforms concatenating the temporal embedding with input functions into a *single* stream. The two-stream design proposed in DETNO achieves the MSE of $= 0.0050$ versus 0.0053 for the concatenated variant. Separating streams helps the model disentangle complementary roles (sensor-driven context vs. denoising progress) and conditions each query on both without conflating sources, yielding more stable refinement and better fidelity at prediction query points.

Based on the above ablations, we chose the following configuarion for our DETNO architecture: 64 hidden units, 3 experts, a minimum noise of 9×10^{-2}, 10 refinement steps and a two-stream cross attention. This led to a better preservation of high-frequency traffic features while maintaining stable long-horizon rollouts.

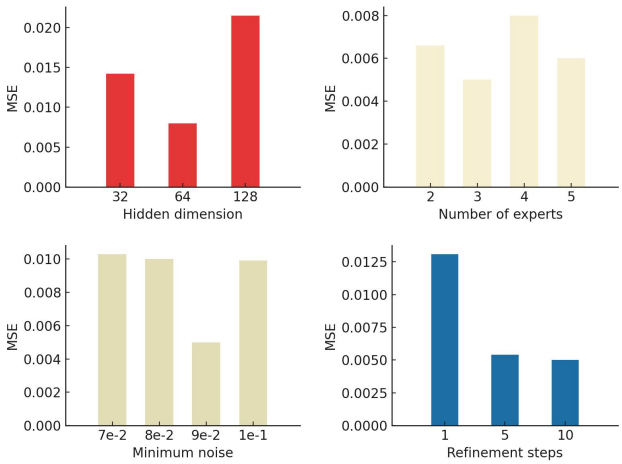

Figure 4: Ablation results for hidden dimension, number of experts, minimum noise, and refinement steps. Bars report MSE for each setting; lower is better.

These ablation results confirm our final configuration: $d_{\text{model}} = 64$, 3 experts, min_noise_std $= 9 \times 10^{-2}$, and $K = 10$ refinement steps.

Conclusion

This work introduced DETNO, a diffusion-enhanced transformer neural operator that addresses two persistent challenges in scientific traffic forecasting: spectral bias against high-frequency features and error accumulation in long rollouts. DETNO unifies operator learning and denoising in a single stage via a heterogeneous cross-attention module that conditions queries on two distinct streams, sensor-driven input functions (operator stream) and the denoising timestep (diffusion stream), and augments capacity with a mixture-of-experts backbone and linear attention for scalability. A v-parameterized diffusion objective with DDIM sampling enables efficient, few-step refinement without changing the training loss, while the model's resolution-free formulation supports super-resolution queries at arbitrary space–time coordinates. In a controlled LWR–Godunov setting, DETNO consistently outperformed neural operator baselines such as DeepONet and GNOT across qualitative and quantitative analyses.

References

Alghamdi, T.; et al. 2022. A comparative study on traffic modeling techniques for predicting and simulating traffic behavior. *Future Internet*, 14(10): 294.

Aw, A.; and Rascle, M. 2000. Resurrection of" second order" models of traffic flow. *SIAM journal on applied mathematics*, 60(3): 916–938.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Guo, Y.; Song, J.; Cao, X.; Zhao, C.; and Leng, H. 2025. Physics Field Super-resolution Reconstruction via Enhanced Diffusion Model and Fourier Neural Operator. *Theoretical and Applied Mechanics Letters*, 100604.

Hao, Z.; Wang, Z.; Su, H.; Ying, C.; Dong, Y.; Liu, S.; Cheng, Z.; Song, J.; and Zhu, J. 2023. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, 12556–12569. PMLR.

Jain, V.; Sharma, A.; and Subramanian, L. 2012. Road traffic congestion in the developing world. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, 1–10.

Kong, X.; Xing, W.; Wei, X.; Bao, P.; Zhang, J.; and Lu, W. 2020. STGAT: Spatial-temporal graph attention networks for traffic flow forecasting. *IEEE Access*, 8: 134363–134372.

Lana, I.; Del Ser, J.; Velez, M.; and Vlahogianni, E. I. 2018. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2): 93–109.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

Lighthill, M. J.; and Whitham, G. B. 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the royal society of london. series a. mathematical and physical sciences*, 229(1178): 317–345.

Lippe, P.; Veeling, B.; Perdikaris, P.; Turner, R.; and Brandstetter, J. 2023. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36: 67398–67433.

Lu, L.; Jin, P.; and Karniadakis, G. E. 2019. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*.

Moseley, B.; Markham, A.; and Nissen-Meyer, T. 2023. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4): 62.

Oommen, V.; Bora, A.; Zhang, Z.; and Karniadakis, G. E. 2025. Integrating Neural Operators with Diffusion Models Improves Spectral Representation in Turbulence Modeling. arXiv:2409.08477.

Peng, H.; Wang, H.; Du, B.; Bhuiyan, M. Z. A.; Ma, H.; Liu, J.; Wang, L.; Yang, Z.; Du, L.; Wang, S.; et al. 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521: 277–290.

Perrone, N.; Lehmann, F.; Gabrielidis, H.; Fresca, S.; and Gatti, F. 2025. Integrating fourier neural operators with diffusion models to improve spectral representation of synthetic earthquake ground motion response. *arXiv preprint arXiv:2504.00757*.

Prasthofer, M.; De Ryck, T.; and Mishra, S. 2022. Variable-input deep operator networks. *arXiv preprint arXiv:2205.11404*.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707.

Ramezankhani, M.; Deodhar, A.; Parekh, R. Y.; and Birru, D. 2025a. An advanced physics-informed neural operator for comprehensive design optimization of highly-nonlinear systems: An aerospace composites processing case study. *Engineering Applications of Artificial Intelligence*, 142: 109886.

Ramezankhani, M.; Patel, J. M.; Deodhar, A.; and Birru, D. 2025b. GITO: Graph-Informed Transformer Operator for Learning Complex Partial Differential Equations. *arXiv preprint arXiv:2506.13906*.

Rap, J.; and Das, A. 2025. ON-Traffic: An Operator Learning Framework for Online Traffic Flow Estimation and Uncertainty Quantification from Lagrangian Sensors. *arXiv preprint arXiv:2503.14053*.

Serrano, L.; Wang, T. X.; Le Naour, E.; Vittaut, J.-N.; and Gallinari, P. 2024. AROMA: Preserving spatial structure for latent PDE modeling with local neural fields. *Advances in Neural Information Processing Systems*, 37: 13489–13521.

Shafik, A. K.; and Rakha, H. A. 2025. Real-Time Turning Movement, Queue Length, and Traffic Density Estimation and Prediction Using Vehicle Trajectory and Stationary Sensor Data. *Sensors*, 25(3): 830.

Shi, R.; Mo, Z.; and Di, X. 2021. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 540–547.

Smith, B. L.; and Demetsky, M. J. 1997. Traffic flow forecasting: comparison of modeling approaches. *Journal of Transportation Engineering*, 123(4): 261–266.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Thodi, B. T.; Ambadipudi, S. V. R.; and Jabari, S. E. 2024. Fourier neural operator for learning solutions to macroscopic traffic flow models: Application to the forward and inverse problems. *Transportation research part C: emerging technologies*, 160: 104500.

Usama, M.; Ma, R.; Hart, J.; and Wojcik, M. 2022. Physics-informed neural networks (PINNs)-based traffic state estimation: An application to traffic network. *Algorithms*, 15(12): 447.

Wang, Y.; Jing, C.; Xu, S.; and Guo, T. 2022. Attention based spatiotemporal graph attention networks for traffic flow forecasting. *Information Sciences*, 607: 869–883.

Yu, B.; Yin, H.; and Zhu, Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Zhang, C.; James, J.; and Liu, Y. 2019. Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *Ieee Access*, 7: 166246–166256.