STGRAPHFM: PRIOR-ANCHORED FLOW MATCHING FOR JOINT SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatiotemporal forecasting requires accurate central predictions, informative uncertainty estimates, and coherent future scenarios across interacting locations. These goals call for a model that captures predictable structure while organizing variation across nodes and forecast leads. We propose STGraphFM, a prior-anchored flow-matching framework that unifies deterministic and probabilistic forecasting. A node-wise temporal prior provides a future-shaped predictive anchor, graph-correlated perturbations initialize structured alternatives around it, and a sample-dependent graph-time velocity field transports each alternative across nodes and forecast leads. Bounded velocity regression trains this field along sampled source–target paths without unrolling generation. The same learned transport maps the anchor to a central forecast and perturbed sources to complete future trajectories. Across traffic, air-quality, and gridded-temperature benchmarks, STGraphFM achieves strong point accuracy and improves probabilistic forecasts, including 18.4–45.8% lower continuous ranked probability score (CRPS) than the strongest evaluated probabilistic baselines on traffic. Matched air-quality comparisons retain joint-score gains at fixed coordinate-wise empirical marginals, providing evidence beyond marginal accuracy.

1 INTRODUCTION

Forecasting spatially distributed systems supports traffic management and environmental risk assessment (Liu et al., 2023d; Wang et al., 2020; Rasp et al., 2024). These applications need accurate central predictions, uncertainty estimates, and scenarios of changes across a region and over time. Point forecasts provide a reference, while marginal distributions describe uncertainty at each location and lead. Yet identical node-wise event probabilities can imply different probabilities of simultaneous exceedance (Figure 1(c)). Joint scenarios represented by complete future samples additionally preserve associations across locations and leads.

Graph forecasters exploit temporal regularities and spatial interactions (Li et al., 2018; Wu et al., 2019; Shao et al., 2022b; Liu et al., 2023a). Diffusion models sample complete future fields (Wen et al., 2023; Lin et al., 2025), and structured flow sources incorporate history (Kollovieh et al., 2025). These approaches expose a modeling choice between information specified before generation and dependence learned during transport. We use a historical prediction to specify node–lead reference levels, reserving sample-dependent cross-node and cross-lead variation for the transport.

We separate predictive anchoring from sample-dependent graph-time transport. A node-wise predictor expresses history as a reference on the future node–lead grid. Yet an accurate reference does not determine how alternative outcomes vary together. Independent perturbations omit spatial dependence; a prescribed graph filter supplies an initial pattern but cannot adapt it to each evolving future. This motivates a structured source followed by learned updates that respond to the current sample across nodes and leads.

STGraphFM implements this separation with a temporal prior, graph-correlated perturbations, and a conditional velocity network. Temporal and graph attention let each coordinate's update depend on other coordinates of the same sample. A bounded flow-matching objective trains these updates along source–target paths without unrolling generation (Lipman et al., 2023; Tong et al., 2024). Transporting the anchor yields a deterministic forecast; transporting its perturbations yields joint samples.

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.

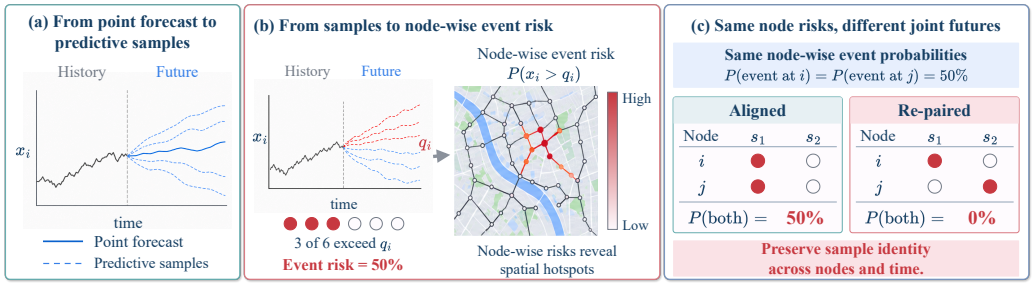

Figure 1: Why complete futures matter. (a) A point forecast specifies one future, while predictive sampling represents multiple plausible futures. (b) Complete samples induce node-wise event probabilities and spatial risk summaries. (c) Identical node-wise event probabilities can imply different probabilities of simultaneous exceedance because event pairing across nodes differs.

We evaluate central forecasts, predictive distributions, and joint scenarios in sequence. Traffic, air-quality, and weather benchmarks test forecasting quality; matched air-quality controls then isolate sample dependence. Our contributions are:

- **Predictive anchoring for forecasting and sampling.** A learned prediction anchors a shared transport that produces both a central forecast and complete future samples.
- **Coupled sample updates.** Graph-correlated initialization and state-dependent graph-time attention organize variation before and during transport.
- **Forecasting quality and dependence evidence.** Sensor and weather benchmarks show strong MAE and CRPS results; matched air-quality controls retain joint-score gains at identical marginals.

2 Related Work

Forecasts and joint future samples. Graph diffusion, adaptive connectivity, and attention support point forecasting (Li et al., 2018; Yu et al., 2018; Wu et al., 2019; Bai et al., 2020; Shao et al., 2022b; Jiang et al., 2023; Liu et al., 2023a); STID and GraphSparseNet address computation (Shao et al., 2022a; Kong et al., 2025). DeepAR, Temporal Fusion Transformer, and DeepSTUQ estimate distributions or quantiles (Salinas et al., 2020; Lim et al., 2021; Qian et al., 2023). DiffSTG and SpecSTG generate graph futures by diffusion (Wen et al., 2023; Lin et al., 2025); DYffusion combines interpolation and forecasting (Rühling Cachay et al., 2023). CSDI and PriSTI originate in imputation and adapt to forecasting by masking future entries (Tashiro et al., 2021; Liu et al., 2023b). Calibration addresses marginal reliability (Kuleshov et al., 2018).

Sources, conditioning, and transport. Flow matching, rectified flow, and stochastic interpolants learn distributional transport (Lipman et al., 2023; Tong et al., 2024; Liu et al., 2023c; Albergo & Vanden-Eijnden, 2023). Conditional TSFlow uses a history-conditioned Gaussian-process source and supplies history and a mask to its velocity network (Kollovieh et al., 2025). GiFlow uses graph-filtered observations, space–time attention, and message passing for imputation (Zhang et al., 2026). STGraphFM uses a predictive field as its history interface, graph-correlated sources, and sample-dependent transport of complete futures.

3 Problem Formulation and Preliminaries

Given H historical observations X_h on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes, we predict a joint distribution over the next T steps, $Y \in \mathbb{R}^{N \times T \times d_y}$, where d_y is the number of output channels. Conditions $\Omega = \{C_h, C_f, \mathcal{G}, C_s, E\}$ comprise historical and future calendar features, graph structure, optional static context, and protocol-specified exogenous inputs. The model returns a deterministic forecast $\hat{Y}_{\text{cs}}$ and an ensemble in which each member covers all nodes and forecast leads:

$$\hat{Y}^{(m)} \sim p_\theta(Y \mid X_h, \Omega), \qquad m = 1, \ldots, N_s. \tag{1}$$

Here, *joint* refers to all node–lead coordinates and output channels of Y together. A marginal describes one coordinate; dependence determines which values co-occur within a complete future. An event $\mathcal{A}$ is a condition on this field, such as simultaneous exceedance within a region. Its predictive probability is $P_\theta(Y \in \mathcal{A} \mid X_h, \Omega)$, estimated from generated members by

$$\widehat{P}(\mathcal{A} \mid X_h, \Omega) = \frac{1}{N_s} \sum_{m=1}^{N_s} \mathbf{1}\left\{\hat{Y}^{(m)} \in \mathcal{A}\right\}.$$

We use *event risk* as shorthand for this probability, not expected decision loss. Node-wise event probabilities alone do not determine a multi-node event probability.

Conditional flow matching for joint generation. CFM learns a transformation from a source distribution to a data distribution (Lipman et al., 2023; Tong et al., 2024). For context C, generation starts from $z_0 \sim p_0(\cdot \mid C)$ and solves

$$\frac{\mathrm{d}z_s}{\mathrm{d}s} = v_\theta(z_s, s \mid C), \qquad s \in [0, 1].$$

The terminal map transforms the source distribution into the model distribution: each evolution is deterministic, but different source draws produce an ensemble. Training avoids solving this ODE. Draw $(x_0, x_1) \sim \pi_C$, whose marginals are the source and data distributions, and $s \sim \mathcal{U}(0, 1)$. The linear path $x_s = (1 - s)x_0 + sx_1$ has pair-specific velocity $x_1 - x_0$, giving

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{C,(x_0,x_1)\sim\pi_C,s} \left\| v_\theta(x_s, s \mid C) - (x_1 - x_0) \right\|_2^2.$$

Its population minimizer is $v^\star(x, s \mid C) = \mathbb{E}[X_1 - X_0 \mid X_s = x, s, C]$: it averages path directions through the current state, not all futures given history (Liu et al., 2023c). Under standard regularity assumptions, this field transports the interpolating distributions; sampling paths need not follow individual training lines. Here every state is a complete future field. Coupling its coordinates allows joint variation within a sample. Transport time s differs from forecast lead h. The next section specifies our source, coupled field, and bounded regression surrogate.

4 STGRAPHFM

STGraphFM follows three stages: a predictive anchor, structured alternatives, and sample-dependent transport (Figure 2). The anchor uses history to set future reference levels. Graph-correlated perturbations create alternatives around it; a graph-time velocity field then evolves each complete future across nodes and leads. The same transport maps the anchor to a central forecast and perturbed sources to samples of the predictive distribution. Each complete sample is a joint scenario for the event queries in Section 3.

4.1 PREDICTIVE ANCHOR AND STRUCTURED ALTERNATIVES

The temporal prior is restricted to node-wise prediction. For each node, a lightweight multilayer perceptron (MLP) combines historical values with calendar and identity embeddings to produce a predictive reference and scale:

$$\mu_p, \ell_p = f_\phi(X_h, C_h, C_f, C_s). \tag{2}$$

Here μ_p is the predictive anchor and ℓ_p is clipped log-variance. This node-independent parameterization performs no cross-node interaction; a similar forecasting design is used by Shao et al. (2022a). Layer structure and optional static inputs are in Appendix B.

To create structured alternatives around the anchor, we mix independent noise with its graph-smoothed counterpart. The local component retains node-level variation, while smoothing correlates perturbations across graph neighborhoods. For Gaussian noise ϵ and a row-normalized graph operator $\mathcal{S}_A$ acting on the node axis, we use

$$\begin{aligned} \eta_G &= \mathrm{Renorm}((1 - \rho)\epsilon + \rho\, \mathrm{Renorm}(\mathcal{S}_A(\epsilon))), \quad \epsilon \sim \mathcal{N}(0, I), \\ x_0^{\mathrm{train}} &= \mathrm{sg}(\mu_p) + a\, \mathrm{sg}(\sigma_s) \odot \eta_G, \qquad z_0^{(m)} = \mu_p + \tau\sigma_s \odot \eta_G^{(m)}. \end{aligned} \tag{3}$$

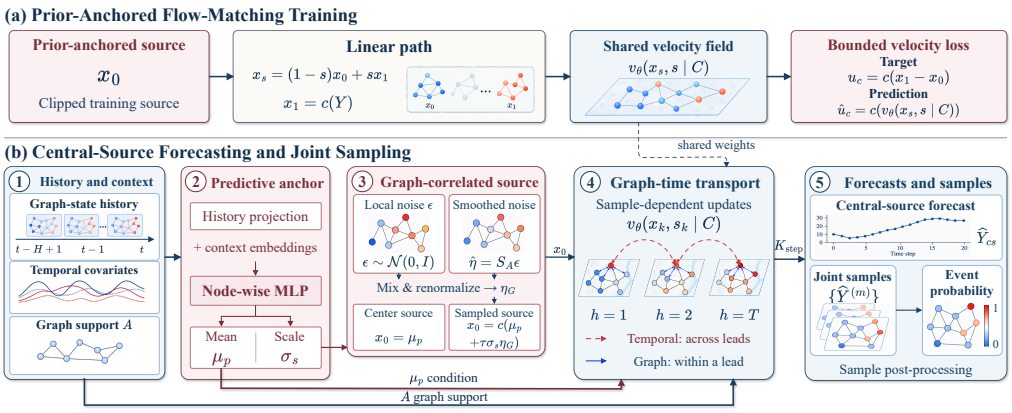

Figure 2: (a) Bounded velocity regression (Eq. 6) learns a shared field along source–target paths. (b) A predictive anchor and graph-correlated perturbations initialize central and sample paths. Temporal updates couple forecast leads; graph updates couple nodes within a lead. Shared transport produces a central forecast and joint samples; post-processed sample frequencies yield event probabilities.

Renorm matches the overall empirical noise scale to that of ϵ without subtracting the mean. The source components have distinct roles: μ_p sets the reference location, $\sigma_s = \sigma_0 s_p$ sets the perturbation amplitude, and ρ determines the graph-noise mixture. The factor s_p optionally uses a clipped heteroscedastic scale from ℓ_p. Training multiplier a varies the radius seen by the velocity network; inference temperature τ controls the radius of new sources. Stop-gradient, sg, leaves prior learning to its own supervision, as described in Section 4.3.

The source specifies only the initial graph-correlated variation. It does not determine how dependence changes with the sampled state, forecast lead, or context. The velocity field therefore learns the state-dependent evolution of this initially structured ensemble.

4.2 Sample-dependent graph-time transport

To make this evolution respond to the current realization, we represent each node–lead coordinate as a token. Temporal attention links leads at a node; graph attention links nodes at a lead. The anchor supplies a coordinate-aligned historical reference, without passing the prior's hidden representation. We project the state, future calendar features, and detached anchor into width D, then add embeddings:

$$\begin{aligned} z_{i,h}^{\text{state}} &= \gamma_x \,\text{GELU}(W_{\text{in}}[x_{s,i,h}; C_{f,i,h}; \text{sg}(\mu_{p,i,h})]), \\ Z_{i,h}^{(0)} &= F_{\text{fuse}}\left(z_{i,h}^{\text{state}} + e_s + e_i + e_{i,h}^{\text{rhythm}} + e_i^{\text{static}} + e_{i,h}^{\text{ext}}\right). \end{aligned} \tag{4}$$

Here γ_x scales the projected state, GELU is the Gaussian error linear unit, and F_{fuse} is a small fusion network. The terms $e_s, e_i, e^{\text{rhythm}}$ encode transport time, node identity, and calendar rhythm; static and external embeddings are used when their inputs are available. With layer normalization (LN) and a feed-forward network (FFN), each graph-time block applies

$$\begin{aligned} U &= Z + \text{TempAttn}(\text{LN}(Z)), \\ V &= U + \text{GraphAttn}_{\mathcal{N}_{K_g}}(\text{LN}(U)), \\ Z' &= V + \text{FFN}(\text{LN}(V)). \end{aligned} \tag{5}$$

Graph attention selects K_g candidates by graph-support scores and combines state-dependent query–key affinities with support-derived biases. Candidates can include zero-score nonedges when positive support is insufficient. A final projection gives one velocity vector per node and lead. Appendix B gives the fusion layers, head gates, and lag biases.

Factorization organizes temporal and spatial interactions without attending densely over all NT tokens. For batch size B, temporal and graph attention cost $O(BNT^2D)$ and $O(BTNK_gD)$, respectively; the reported implementation also stores dense graph support and selects candidates

at each velocity evaluation. The scaling experiments include these costs. Factorization concerns computation, not independence of the predictive distribution. Each velocity can depend on other coordinates of the same sample: shared parameters therefore produce different, coupled updates for different source draws.

4.3 Learning from source–target paths

The architecture specifies how updates depend on the current sample; training must supply these updates without repeatedly decoding trajectories. Source–target interpolation provides this supervision, with the observed future as endpoint. In our bounded CFM surrogate, using normalized coordinates, let $c(z) = \mathrm{clip}(z, -8, 8)$, set $x_0 = c(x_0^{\mathrm{train}})$ and $x_1 = c(Y)$, and form $x_s = (1-s)x_0 + sx_1$ for $s \sim \mathcal{U}(0,1)$. The objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\frac{1}{D_{\mathcal{M}}}\left\|\sqrt{w(a)} \odot \mathcal{M} \odot \left[c(v_\theta(x_s, s \mid C)) - \mathrm{sg}(c(x_1 - x_0))\right]\right\|_2^2\right], \tag{6}$$

where $\mathcal{M}$ marks valid target entries, $D_{\mathcal{M}} = \sum \mathcal{M} + 10^{-5}$, and C comprises the conditions in Eq. 4 and graph support. Radius augmentation uses $w(a) = \mathrm{clip}(a^{-2}, w_{\min}, w_{\max})$; otherwise $w(a) = 1$. Both path endpoints and regression velocities are bounded. At inference, we apply the raw velocity and clip the updated state; masking and clipping derivatives are detailed in Appendix B.

The prior receives masked SmoothL1 supervision for its mean and, when enabled, a Gaussian likelihood loss for its mean and scale (Kendall & Gal, 2017). Combining these with velocity regression gives

$$\mathcal{L} = \lambda_p \mathcal{L}_{\mathrm{prior}} + \lambda_{\mathrm{nll}} \mathcal{L}_{\mathrm{nll}} + \lambda_{\mathrm{FM}} \mathcal{L}_{\mathrm{FM}}. \tag{7}$$

Prior losses learn the reference and scale; the flow loss trains updates around their detached outputs. Although regression errors sum over coordinates, each velocity uses the graph-time network, so supervision acts on coupled updates rather than independent node-wise forecasting functions. Auxiliary objectives are in Eqs. 24–25.

4.4 Shared decoding for forecasts and samples

The trained field serves both outputs through their initial states: transporting μ_p yields the central-source forecast, while transporting a perturbed source yields a predictive sample. Nonlinear transport means the central-source forecast need not equal the ensemble mean:

$$\begin{aligned} \hat{Y}_{\mathrm{cs}} &= \mathrm{Transport}_{K_{\mathrm{step}}}(\mu_p), \\ \widetilde{Y}^{(m)} &= \mathrm{Transport}_{K_{\mathrm{step}}}\left(c(\mu_p + \tau\sigma_s \odot \eta_G^{(m)})\right). \end{aligned} \tag{8}$$

With K_{step} steps, transport performs the explicit update

$$x_{k+1} = c\big(x_k + K_{\mathrm{step}}^{-1} v_\theta(x_k, s_k \mid C)\big), \qquad s_k = (k + 0.5)/K_{\mathrm{step}}. \tag{9}$$

Each step uses the interval-start state and a midpoint time token, requiring one network evaluation. In the one-step setting this is $v_\theta(x_0, 0.5 \mid C)$. Sample count N_s, step count K_{step}, and source temperature τ separately specify ensemble size, transport cost, and initial radius.

For the main sensor results, output calibration mixes the sample center toward $\hat{Y}_{\mathrm{cs}}$ with coefficient $\rho_c = 0.25$ and applies a validation-selected scale a_c (Eq. 43). Because member identity is preserved across all nodes and leads, the decoded ensemble directly supports the event probabilities defined in Section 3.

The central forecast provides a point estimate. The ensemble represents the predictive distribution: coordinate-wise values characterize its marginals, while preserved member identities define joint scenarios. The experiments assess these three levels and then isolate the roles of source structure and sample-dependent transport.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We test central forecasts and predictive distributions on traffic, air-quality, and gridded temperature data, then use matched air-quality controls to examine joint scenarios. Further analyses examine event queries, sampling cost, and graph size. Sensor benchmarks are PEMS03 (358 nodes) (Song et al., 2020), LargeST-SD (716 nodes) (Liu et al., 2023d), and KnowAirV2 (228 nodes) (Wang et al., 2025). WeatherBench2 evaluates ten-day near-surface temperature forecasting (Rasp et al., 2024). Deterministic baselines cover node-wise, graph-convolutional, transformer, and large-graph architectures; probabilistic baselines include diffusion, flow, and uncertainty-based predictors. Appendix C details their interfaces, forecasting adaptations, and training settings.

Sensor data use chronological splits, training-only normalization, and inverse-normalized evaluation. Traffic models use no external token. Unless specified, $N_s = 16$, $K_{\text{step}} = 1$, and $K_g = 8$; main sensor rows report three-seed means and standard deviations.

Evaluation proceeds from the central forecast to the predictive distribution and joint scenarios. MAE, RMSE, and MAPE assess the central-source forecast; the continuous ranked probability score (CRPS) assesses coordinate-wise predictive distributions (Matheson & Winkler, 1976; Gneiting & Raftery, 2007). For the 10th–90th percentile interval, PICP_{80} is observed coverage (target 0.8), and MPIW_{80} is mean width in target units (lower is better). Width measures sharpness, while coverage measures how often the interval contains the observation; we report both to characterize their balance. For joint scenarios, Energy Score (ES) and graph Variogram Score (GVS) assess complete futures and graph-edge differences, respectively; lower is better. Since joint scores also depend on marginals, Section 5.3 fixes empirical marginals to test cross-coordinate organization. Formal definitions are in Appendix C. For matched traffic calibration, each model selects its spread scale on 512 validation windows and is evaluated on the same 512 test windows.

5.2 FORECASTING QUALITY

The central-source forecast achieves the lowest MAE on all three sensor benchmarks, while other models retain advantages in RMSE or MAPE (Table 1). The distributional results are more consistent: STGraphFM ranks first in CRPS and MPIW_{80} on all three datasets. Relative to the strongest probabilistic baseline, CRPS falls by 18.4% on PEMS03, 45.8% on LargeST-SD, and 6.6% on KnowAirV2.

The width advantage also persists with coverage close to the nominal 80% target. After validation-fixed traffic calibration, STGraphFM and TSFlow achieve PICP_{80} of $0.802/0.803$ on PEMS03 and $0.777/0.794$ on LargeST-SD. At these comparable coverages, STGraphFM intervals are 34.2% and 71.0% narrower, with lower CRPS (11.118 vs. 15.893 and 12.455 vs. 40.349). The model thus combines compact intervals with 77.7–80.2% coverage; Table 8 also reports SpecSTG under the same calibration protocol.

5.3 GRAPH COMPONENTS AND TRAINING OBJECTIVE

Source structure and graph-time transport. To connect forecasting quality to the model design, we examine structured alternatives and sample-dependent transport. Graph correlation initializes spatially related deviations; graph attention updates them during transport. Replacing the source by isotropic noise or removing the support-ranked attention branch raises ES by 19.2–21.0% and GVS by 203.5–225.4% (Table 2). The complete branch includes selected nonedge candidates, so this ablation tests support-ranked spatial updates as a whole rather than physical-edge propagation alone.

Table 1: Sensor-network forecasting (three-seed mean ± standard deviation). Bold/underline mark best/second means on each dataset. PICP_{80} is ranked by distance to 0.8; other metrics are lower-is-better. MAPE is a ratio.

(a) Deterministic forecasting

Model	MAE↓	RMSE↓	MAPE↓
PEMS03			
STAEformer	15.996±0.160	26.687±0.325	0.177±0.018
STID	16.132±0.294	26.071±0.356	0.185±0.020
GraphWaveNet	16.138±0.083	26.581±0.203	**0.162**±0.004
AGCRN	20.763±0.261	33.620±0.650	0.223±0.021
GSNet	16.895±0.024	**25.387**±0.036	0.172±0.007
STGraphFM	**15.870**±0.017	26.699±0.525	0.173±0.001
LargeST-SD			
STAEformer	19.173±0.219	**31.024**±0.331	0.131±0.011
STID	19.351±0.095	32.123±0.271	0.133±0.004
GraphWaveNet	21.652±0.705	33.616±1.105	0.146±0.009
AGCRN	34.197±0.208	53.817±0.560	0.223±0.005
GSNet	24.321±0.977	35.451±0.948	0.161±0.009
STGraphFM	**18.925**±0.272	31.643±0.430	**0.128**±0.001
KnowAirV2			
STAEformer	13.199±0.154	22.800±0.162	0.582±0.012
STID	13.617±0.049	23.501±0.022	0.681±0.013
GraphWaveNet	13.227±0.088	22.424±0.086	**0.571**±0.028
AGCRN	14.263±0.118	23.917±0.038	0.747±0.025
GSNet	13.454±0.088	**21.075**±0.134	0.638±0.004
STGraphFM	**12.849**±0.170	22.267±0.146	0.659±0.071

(b) Probabilistic forecasting

Model	CRPS↓	PICP_{80}	MPIW_{80} ↓
PEMS03			
SpecSTG	15.060±0.276	0.772±0.009	59.396±2.272
TSFlow	15.135±0.106	**0.799**±0.003	64.066±0.672
PriSTI-F	15.362±0.047	0.830±0.004	70.010±1.045
CSDI	15.707±0.064	0.858±0.004	77.047±1.181
DeepSTUQ	15.813±0.027	0.837±0.001	74.375±0.213
STGraphFM	**12.282**±0.023	0.735±0.001	**43.637**±0.095
LargeST-SD			
SpecSTG	26.919±0.423	0.760±0.009	103.318±2.632
TSFlow	27.943±0.421	**0.798**±0.003	117.611±2.143
PriSTI-F	27.672±0.084	0.826±0.006	124.442±3.202
CSDI	28.250±0.129	0.854±0.005	136.886±3.522
DeepSTUQ	29.428±0.131	0.838±0.002	136.156±0.579
STGraphFM	**14.594**±0.173	0.725±0.010	**49.929**±1.979
KnowAirV2			
SpecSTG	10.811±0.068	0.784±0.001	43.899±0.332
TSFlow	11.238±0.046	**0.813**±0.006	48.403±0.606
PriSTI-F	10.928±0.050	0.836±0.003	50.512±0.104
CSDI	11.335±0.048	0.868±0.002	56.163±0.115
DeepSTUQ	11.427±0.051	0.827±0.004	51.997±0.490
STGraphFM	**10.096**±0.068	0.648±0.064	**27.621**±3.906

Table 2: Components and generator alternatives on LargeST-SD (three-seed means); full results in Appendix Table 12.

Variant / alternative	CRPS↓	ES↓	GVS↓
STGraphFM	**14.594**	**18.592**	**3.041**
Isotropic source	15.559	22.153	9.228
No graph attention	15.627	22.489	9.895
Prior + calibrated Gaussian samples	16.123	25.854	12.429
Prior + graph residual (no flow)	15.626	24.618	8.857

The Gaussian and graph-residual alternatives generate samples around the prior without flow transport. Their higher CRPS and joint scores motivate the matched objective and state-input tests below. Affine calibration has a smaller role: relative to raw samples, it changes CRPS/PICP_{80}/MPIW_{80} from $14.882/.698/47.253$ to $14.594/.725/49.929$; the raw generator already retains the structural advantage (Table 12).

Training objective and sample-state dependence. An accurate anchor could improve forecasts even if every source received the same correction. We test the additional role of the current sample state, alongside the choice of training objective. The matched KnowAirV2 comparison fixes the backbone, data, optimizer, and five-epoch budget for two seeds. Score16 trains the same generator with a 16-member endpoint CRPS–ES objective; sample-shared correction retains velocity regression but feeds the detached anchor in place of each current state. Appendix E gives the aligned decoding and selection protocol.

At this training budget, velocity regression reduces CRPS/ES/GVS by $2.97\%/3.15\%/21.76\%$ relative to Score16 and $3.35\%/3.56\%/21.48\%$ relative to shared correction (Appendix E). For each trained pair, 95% paired block-bootstrap intervals over evaluation windows exclude zero at both 24-hour and 168-hour block lengths. The larger relative change in GVS than CRPS points to an effect on spatial

Table 3: Matched KnowAirV2 (16 samples, two seeds). (a) Calibrated means over all 14,003 windows. (b) STGraphFM-minus-control ranges across two seeds and both fixed-marginal directions; negative values favor STGraphFM. Shared denotes sample-shared correction.

(a) Event probability and retrieval

Method	Brier↓	Tie-AP↑	Region IoU↑
SpecSTG	.0362	.7753	.2187
STGraphFM	**.0329**	**.8013**	**.2403**

(b) Fixed-marginal dependence

Control	ΔES	ΔGVS	$10^3\Delta\text{Brier}_{\text{macro}}$
Score16	$[-.0226, -.0122]$	$[-.488, -.094]$	$[-.884, -.532]$
Shared	$[-.0235, -.0127]$	$[-.488, -.093]$	$[-.883, -.536]$

Brier: squared event-probability error; Tie-AP: tie-aware average precision; Region IoU: connected-region intersection-over-union; $\text{Brier}_{\text{macro}}$: equal-weight mean over 11 regions.

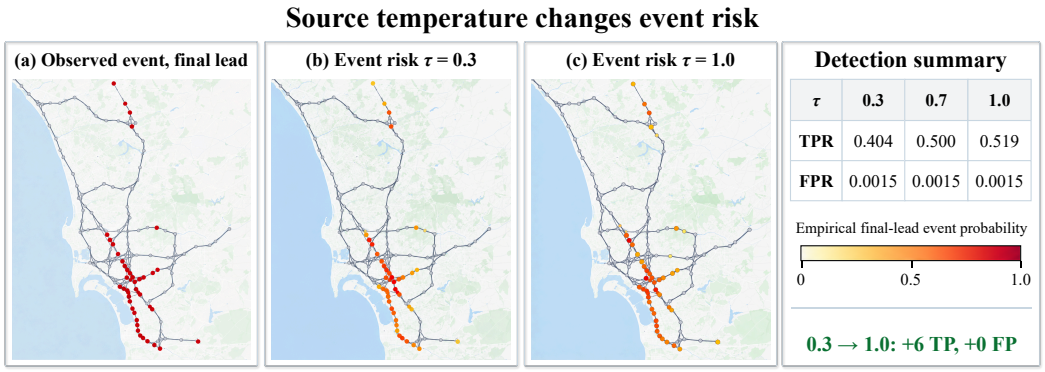

Figure 3: Source-temperature effects on LargeST-SD. (a) Observed final-lead events. (b,c) Empirical event probabilities at $\tau = 0.3$ and 1.0. The summary also includes $\tau = 0.7$. Events exceed training-derived node thresholds; detection uses probability ≥ 0.5. TPR/FPR denote true-/false-positive rates.

relationships. Table 3 connects the fixed-marginal test to the event-risk interface produced by the same complete future samples.

Dependence beyond marginal accuracy. Joint scores can improve through marginals alone. To isolate coupling, we fix either model's 16 empirical values per coordinate and reassemble them using the other model's member ranks. In both directions, STGraphFM retains lower ES, GVS, and eleven-region macro Brier than Score16 and shared correction, with identical coordinate-wise CRPS/PICP/MPIW by construction (Table 3(b); Appendix E). On each model's calibrated outputs, trajectory CRPS also improves for temporal increments, graph differences, regional means and peaks, exceedance fractions, and duration (Appendix Table 18).

5.4 Sampling Cost and Scenario Queries

Computational cost. We measure KnowAirV2 on an RTX 5090 D with training batch 8 and inference batch 2, generating 16 samples per window. STGraphFM takes 195 ms per window, including its central forecast, giving $7.60\times$ / $6.94\times$ speedups over CSDI/SpecSTG (Table 4). Its 71 ms training update falls between their 54 and 315 ms.

Table 4: Training and inference cost.

Model	Train↓ ms/update	Infer↓ ms/window
CSDI	**54.43**	1482.72
SpecSTG	314.68	1353.35
STGraphFM	71.46	**195.06**

The main saving is therefore at sampling: a one-step velocity update avoids repeated denoising, without making training uniformly cheapest. The LargeST-SD step sweep likewise favors one step at nearly unchanged CRPS (Appendix C.4). At the timed checkpoints, CSDI has lower CRPS and GVS, so this comparison reflects a quality–latency trade-off.

Table 5: WeatherBench2 (three-seed means). STAEformer has the lowest baseline MAE on the three sensor datasets. MAE, CRPS, and width are in K. Coverage targets 0.8. Full results: Appendix C (Figure 4, Table 11).

(a) Lead-time MAE (K)

Model	1d	3d	5d	7d	10d
STGraphFM	**0.936**	**1.294**	**1.360**	**1.387**	**1.403**
STAEformer	0.976	1.378	1.468	1.522	1.584

(b) Calibrated forecasts

Model	CRPS↓	PICP_{80}	MPIW_{80} ↓
STGraphFM	**0.990**	0.773	**3.501**
TSFlow	1.387	**0.787**	6.087

Source temperature and event risks. An aligned 512-window sweep fixes the checkpoint, draws, 16 samples, and one step. Raising τ from 0.3 to 1.0 increases coverage, width, and mean event probability on both traffic datasets, while sample-mean MAE stays within 2.1% of its $\tau = 1$ value. CRPS is lowest at $\tau = 0.7$: temperature mainly controls spread around a stable center (Appendix C.4).

Figure 3 illustrates the final-lead response. In this window, raising τ from 0.3 to 1.0 increases true positives from 21 to 27 with one false positive at both settings; the central-source forecast is unchanged. Across all 512 windows, TPR/FPR rise by 1.07/0.39 percentage points (Appendix C.6), showing an operating-point change at a fixed detection threshold.

On all 14,003 KnowAirV2 test windows, STGraphFM improves calibrated Brier, Tie-AP, and region IoU over SpecSTG in both seeds (Table 3(a)); raw outputs agree. Brier falls by 9.1% and region IoU rises by 9.9%, linking better event probabilities to better localization of connected hotspots. This query uses exceedance at any lead (Eq. 16); thresholds and tie handling are in Appendix C.

5.5 Gridded temperature forecasting

Ten-day temperature forecasting. On WeatherBench2 T2M (Rasp et al., 2024), eight six-hour 64×32 fields predict 40 future steps. Against STAEformer, the strongest sensor-network baseline by MAE, STGraphFM reduces MAE by 4.2/6.1/7.3/8.9/11.4% at one/three/five/seven/ten days. Relative to calibrated TSFlow, CRPS falls by 28.6% and intervals narrow by 42.5% at similar 77.3/78.7% coverage (Table 5). The additional point-forecasting baselines are reported in Appendix C.

5.6 Graph scale

LargeST-GBA (2,359 nodes) yields MAE/CRPS of 21.28/16.40 (Appendix Table 9). Across five LargeST-CA graphs of 1,000–5,000 nodes (three seeds, 15 epochs), MAE/CRPS ranges are 6.1/6.4%. Each graph defines a different prediction task. Center/sample latency rises from 5.283/46.384 to 21.274/230.506 ms/window and peak training memory from 7.172 to 31.092 GiB. Over this 1,000–5,000-node range, sample latency increases about fivefold as node count increases fivefold; dense support and candidate selection still constrain larger graphs.

6 Limitations

A node-wise predictive anchor compresses history and can discard information needed for rare or abrupt changes. Marginal calibration does not ensure regional coverage, and unconstrained transport can produce negative values for nonnegative targets. The model also imposes no physical conservation law. Finally, dense graph support and candidate selection limit scaling; more local computation is needed for substantially larger networks.

7 Conclusion

STGraphFM combines a predictive anchor, structured alternatives, and sample-dependent transport for central forecasts and a predictive distribution represented by joint scenarios. It achieves the lowest sensor MAE and improves weather MAE over STAEformer. Calibrated traffic and temperature forecasts have lower CRPS and narrower intervals than TSFlow at comparable coverage; matched air-quality controls retain joint-score gains at fixed marginals.

AI USE STATEMENT

Generative AI tools assisted with bilingual editing, mathematical exposition, code review, supplementary experiment execution and analysis, and document preparation. Revision checks included tracing equations to implementation, checking cited metadata against primary sources, inspecting saved predictions and plotting code, and compiling and visually checking both manuscript versions. Authors remain responsible for the text, code, results, citations, and submission.

ETHICS STATEMENT

This work uses benchmark traffic, air-quality, and weather observations and does not involve human-subject experiments. Operational use of predictive scenarios requires application-specific validation and monitoring.

REPRODUCIBILITY STATEMENT

Code, configurations, and STGraphFM checkpoints are available at `https://anonymous.4open.science/r/ICLR_STgraphFM-E61E/`. Original data releases: PEMS03, LargeST (SD/GBA/CA), KnowAirV2, WeatherBench2.

Appendix C specifies dataset splits, evaluation protocols, model settings, and compute. Implementation details and supplementary experiment protocols appear in Appendices D–G.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=li7qeBbCR1t.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, Red Hook, NY, USA, 2020. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/hash/ce1aad92b939420fc17005e5461e6f48-Abstract.html.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4365–4373, Palo Alto, CA, USA, 2023. Association for the Advancement of Artificial Intelligence. doi: 10.1609/aaai.v37i4.25556. URL https://ojs.aaai.org/index.php/AAAI/article/view/25556.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, pp. 5574–5584, Red Hook, NY, USA, 2017. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *The Thirteenth International Conference on Learning Representations*, Singapore, 2025. OpenReview.net. URL https://openreview.net/forum?id=uxVBbSlKQ4.

Weiyang Kong, Kaiqi Wu, Sen Zhang, and Yubao Liu. GraphSparseNet: a novel method for large scale traffic flow prediction. *Proceedings of the VLDB Endowment*, 18(7):2295–2307, 2025. doi: 10.14778/3734839.3734862. URL https://www.vldb.org/pvldb/vol18/p2295-liu.pdf.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2796–2804, Stockholmsmässan, Stockholm, Sweden, 2018. PMLR. URL https://proceedings.mlr.press/v80/kuleshov18a.html.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, Vancouver, BC, Canada, 2018. OpenReview.net. URL https://openreview.net/forum?id=SJiHXGWAZ.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, Oct 2021. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2021.03.012. URL http://dx.doi.org/10.1016/j.ijforecast.2021.03.012.

Lequan Lin, Dai Shi, Andi Han, and Junbin Gao. SpecSTG: A fast spectral diffusion framework for probabilistic spatio-temporal traffic forecasting. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, Piscataway, NJ, USA, 2025. IEEE. doi: 10.1109/IJCNN64981.2025.11227295. URL https://ieeexplore.ieee.org/document/11227295.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4125–4129, New York, NY, USA, 2023a. ACM. doi: 10.1145/3583780.3615160. URL https://doi.org/10.1145/3583780.3615160.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1927–1939, Piscataway, NJ, USA, 2023b. IEEE. doi: 10.1109/ICDE55515.2023.00150. URL https://doi.org/10.1109/ICDE55515.2023.00150.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, Kigali, Rwanda, 2023c. OpenReview.net. URL https://openreview.net/forum?id=XVjTT1nw5z.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. LargeST: A benchmark dataset for large-scale traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 75354–75371, Red Hook, NY, USA, 2023d. Curran Associates, Inc. doi: 10.52202/075280-3293. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ee57cd73a76bd927ffca3dda1dc3b9d4-Abstract-Datasets_and_Benchmarks.html. Datasets and Benchmarks Track.

James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, June 1976. ISSN 1526-5501. doi: 10.1287/mnsc.22.10.1087. URL http://dx.doi.org/10.1287/mnsc.22.10.1087.

Weizhu Qian, Dalin Zhang, Yan Zhao, Kai Zheng, and James J. Q. Yu. Uncertainty quantification for traffic forecasting: A unified approach. In *Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 992–1004, Piscataway, NJ, USA, 2023. IEEE. doi: 10.1109/ICDE55515.2023.00081. URL https://ieeexplore.ieee.org/document/10184566.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben

Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019. URL https://doi.org/10.1029/2023MS004019.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. DYffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45259–45287, 2023. doi: 10.52202/075280-1961. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8df90a1440ce782d1f5607b7a38f2531-Abstract-Conference.html.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, July 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2019.07.001. URL http://dx.doi.org/10.1016/j.ijforecast.2019.07.001.

Michael Scheuerer and Thomas M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015. doi: 10.1175/MWR-D-14-00269.1. URL https://journals.ametsoc.org/view/journals/mwre/143/4/mwr-d-14-00269.1.xml.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4454–4458, New York, NY, USA, 2022a. ACM. doi: 10.1145/3511808.3557702. URL https://doi.org/10.1145/3511808.3557702.

Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11):2733–2746, 2022b. doi: 10.14778/3551793.3551827. URL https://proceedings.vldb.org/pvldb/vol15/p2733-shao.pdf.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, Palo Alto, CA, USA, 2020. Association for the Advancement of Artificial Intelligence. doi: 10.1609/aaai.v34i01.5438. URL https://ojs.aaai.org/index.php/AAAI/article/view/5438.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, Red Hook, NY, USA, 2021. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2021/hash/cfe8504bda37b575c70ee1a8276f3486-Abstract.html.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, Red Hook, NY, USA, 2017. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, Vancouver, BC, Canada, 2018. OpenReview.net. URL https://openreview.net/forum?id=rJXMpikCZ.

S. Wang, Y. Cheng, Q. Meng, O. Saukh, J. Zhang, J. Fan, Y. Zhang, X. Yuan, and L. Thiele. KnowAir-V2: A benchmark dataset for air quality forecasting with PCDCNet. Zenodo, 2025. URL https://doi.org/10.5281/zenodo.15614907. Data set.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. PM2.5-GNN: A domain knowledge enhanced graph neural network for PM2.5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '20, pp. 163–166, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3397536.3422208. URL https://doi.org/10.1145/3397536.3422208.

Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–12, New York, NY, USA, 2023. ACM. doi: 10.1145/3589132.3625614. URL https://doi.org/10.1145/3589132.3625614.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1907–1913, Macao, China, 2019. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2019/264. URL https://www.ijcai.org/proceedings/2019/264.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640, Stockholm, Sweden, 2018. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2018/505. URL https://www.ijcai.org/proceedings/2018/505.

Zepeng Zhang, Aref Einizade, Jhony H. Giraldo, and Olga Fink. Spatiotemporal imputation with graph-informed flow matching. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.06682.

Kai Zhao, Chenjuan Guo, Yunyao Cheng, Peng Han, Miao Zhang, and Bin Yang. Multiple time series forecasting with dynamic graph modeling. *Proceedings of the VLDB Endowment*, 17(4): 753–765, 2023. doi: 10.14778/3636218.3636230. URL https://www.vldb.org/pvldb/vol17/p753-zhao.pdf.

A Full Problem Formulation and Query Semantics

Figure 1 summarizes the forecasting outputs and graph-level query interface.

Spatiotemporal graph-state generation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a spatiotemporal graph with $N = |\mathcal{V}|$ nodes. Node states may represent traffic speed or flow, air-pollutant concentration, meteorological variables, or other urban sensor observations. Historical observations are

$$X_h \in \mathbb{R}^{B \times N \times H \times d_x}, \tag{10}$$

and target future states are

$$Y \in \mathbb{R}^{B \times N \times T \times d_y}, \tag{11}$$

where B is batch size, H is history length, and T is forecast horizon. We collect optional temporal, structural, and external conditions into

$$\Omega = \{C_h, C_f, \mathcal{G}, C_s, E\}, \tag{12}$$

where C_h and C_f are historical and future time features, C_s denotes optional static or node-level contextual features, and E denotes exogenous variables supplied by the benchmark protocol. Appendix D specifies the KnowAirV2 information setting. Additional distance or similarity information can be included in Ω when available, whereas A^{sup} is constructed from the provided adjacency support.

Traditional forecasting learns a deterministic mapping

$$\hat{Y} = f_\theta(X_h, \Omega). \tag{13}$$

We consider a more general conditional state generation problem:

$$Y \sim p_\theta(Y \mid X_h, \Omega). \tag{14}$$

The model should produce not only an accurate operational deterministic trajectory, but also future samples

$$\{\hat{Y}^{(m)}\}_{m=1}^{N_s} \tag{15}$$

that respect graph structure, temporal rhythm, and external conditions. Controllability in this formulation denotes inference-time control of scenario strength: with X_h, Ω, and model parameters fixed, source temperature τ changes the initial source radius, while N_s and K_{step} separately control estimation quality and transport cost.

Application-facing scenario operators. For downstream analysis, we evaluate three lightweight post-generation operators on one selected target variable (channel zero in the experiments). Each sample is a complete node–horizon trajectory; an ensemble contains N_s such samples. For a node-specific threshold q_i, an exceedance query estimates

$$\hat{p}_{b,i}(q_i) = \frac{1}{N_s} \sum_{m=1}^{N_s} \mathbf{1}\left\{ \max_{1 \le h \le T} \hat{Y}_{b,i,h}^{(m)} > q_i \right\}. \tag{16}$$

A rank query $\mathrm{TopK}(\hat{\boldsymbol{p}}_b, K)$ returns the K nodes with the largest exceedance probabilities. A region query first retains the top $\max\{50, \mathrm{round}(0.15N)\}$ risk-ranked nodes as its fixed-budget candidate set (Appendix C), induces a subgraph from the supplied adjacency support, and returns its largest connected component, breaking size ties by aggregate risk. These operators expose generated states as application-facing probability, ranked-node, and connected-region views without changing the forecasting model. Thresholds and query budgets must be fixed independently of test labels; Appendix C specifies their values.

Flow matching preliminaries. Flow matching gives one way to realize this conditional generation problem as a continuous transport process (Lipman et al., 2023; Tong et al., 2024); rectified flow provides a related short-path formulation (Liu et al., 2023c). We use h to index the observed and forecast sequence: $h \in \{-H+1, \ldots, 0\}$ denotes history, whereas $h \in \{1, \ldots, T\}$ denotes forecast horizons encoded by C_f. We reserve $s \in [0, 1]$ for the Flow Matching path. Each path state x_s therefore represents a complete future graph field over all horizons h, rather than the system state at a particular forecast horizon. In the unconditional setting, one specifies a source distribution p_0 and a target data distribution p_1. Given a source sample $x_0 \sim p_0$ and a data sample $x_1 \sim p_1$, a simple probability path can be written as

$$x_s = (1-s)x_0 + sx_1, \qquad s \in [0, 1], \tag{17}$$

with pair-specific target velocity

$$u_s(x_s \mid x_0, x_1) = x_1 - x_0. \tag{18}$$

Flow matching trains a velocity field $v_\theta(x_s, s)$ to match this target velocity,

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{s, x_0, x_1} \left\| v_\theta(x_s, s) - u_s(x_s \mid x_0, x_1) \right\|_2^2. \tag{19}$$

The pair-conditioned target is distinct from the population field. With context C, the standard squared-loss minimizer is

$$v^\star(x, s \mid C) = \mathbb{E}[X_1 - X_0 \mid X_s = x, s, C]. \tag{20}$$

It averages directions of paths passing through the current state. Under the usual regularity assumptions, its ODE follows the interpolating distributions, not necessarily the individual straight training paths. This statement concerns standard CFM; the bounded surrogate below changes the supervised velocities. After training, generation starts from a new source sample and solves the ordinary differential equation (ODE) $dx_s/ds = v_\theta(x_s, s)$ from $s = 0$ to $s = 1$. For spatiotemporal graph forecasting, the target distribution is not a single unconditional data distribution but the conditional future distribution $p(Y \mid X_h, \Omega)$. Accordingly, the source distribution and velocity field are conditioned on history, graph support, temporal covariates, and external variables; Appendix B specifies the prior-anchored graph source and conditional graph-time velocity field.

Forecasting requirements. The output contract has three parts: a deterministic trajectory for standard point metrics, a sample set for distributional and spatial-risk evaluation, and graph-conditioned scenarios whose strength is controlled by τ. We evaluate these parts with point errors, CRPS and interval metrics, structural scores, and event or region queries.

B Method Details

B.1 Overview: Predictive Anchoring and Graph-Time Transport

STGraphFM separates the reference level of a future field from the way its alternatives vary together. Given history X_h and conditions Ω, a temporal prior supplies the predictive anchor, graph-correlated sources initialize alternative states around it, and graph-time transport updates each state into a complete future. The same transport maps the anchor itself to the deterministic central-source forecast:

$$\mu_p, \ell_p = f_\phi(X_h, C_h, C_f, C_s), \tag{21}$$

$$x_0 = c(\mu_p + \sigma_s \odot \eta_G), \qquad \frac{dx_s}{ds} = v_\theta(x_s, s \mid C_f, A^{\text{sup}}, C_s, E, \mu_p). \tag{22}$$

Here c clips normalized states to $[-8, 8]$, μ_p is the predictive anchor produced by the temporal prior, ℓ_p is the prior network's heteroscedastic log-variance (Kendall & Gal, 2017), σ_s is the effective source-scale tensor defined below, and η_G is a graph-correlated perturbation. The predictive anchor locates the source ensemble and provides a reference at every node and lead. The velocity field combines it with the current sample state, future time covariates, graph support, and static or external conditions. The historical window conditions the flow through μ_p rather than being re-encoded by the velocity network. The predictive field expresses history in the same node–lead coordinates as the transported state, providing a compact conditioning interface without a second history encoder. The temporal prior predicts node-level regularity; the velocity field learns how alternative states evolve relative to that anchor.

Figure 2 summarizes this execution order and the shared deterministic and sampled outputs.

B.2 Temporal Prior and Predictive Anchor

The temporal prior estimates the reference value for each node and forecast lead. It uses history and time features to model node scale, recent-state inertia, periodic rhythm, and horizon-dependent bias. Its output, the predictive anchor, expresses this history in the same coordinates as the future states to be transported.

Our temporal prior extends STID's node-wise design with node identity and time embeddings (Shao et al., 2022a) to predict a mean and a log-variance. Historical observations and historical time features are projected by a 1×1 temporal projection, then fused with node identity, time-of-day, day-of-week, horizon embeddings, and optional static node features. The prior produces

$$\mu_p, \ell_p = f_\phi(X_h, C_h, C_f, C_s), \tag{23}$$

where $\mu_p \in \mathbb{R}^{B \times N \times T \times d_y}$ is the predictive anchor and ℓ_p is the clipped log-variance. The prior inputs define a node-wise forecasting branch before graph-time generation. The prior receives no adjacency-based message passing or external-condition token E, and its hidden representation is not routed to the velocity network; only μ_p and, when enabled, ℓ_p are exposed. This information split reserves graph propagation, external response, and stochastic scenario variation for the flow, while ℓ_p affects only the source amplitude described below.

For each node, the prior flattens the history and its time features into a hidden state, broadcasts it across horizons, and fuses node, periodic, horizon, and optional static embeddings. A shallow multilayer perceptron (MLP) produces the predictive anchor and log-variance at every node-horizon pair; graph propagation remains a responsibility of the flow.

Let $\mathcal{M} \in \{0, 1\}^{B \times N \times T \times d_y}$ denote the validity mask for target entries; for fully observed benchmarks, $\mathcal{M}$ is all ones. The supervised sums include only valid entries and use $D_{\mathcal{M}} = \sum \mathcal{M} + 10^{-5}$ as the numerical denominator. This excludes direct losses at missing positions, not their influence through attention. Path construction assumes finite inputs; Y denotes the normalized preprocessing

output: PEMS zero sentinels retain their standardized values; LargeST and KnowAir replace nonfinite normalized values with zero, while finite masked values retain their standardized values. The flow branch clips this tensor without additional imputation. Missing-position tokens remain in temporal and graph attention and can affect valid positions. An all-invalid example contributes zero to these supervised sums when its tensors are finite; the implementation does not explicitly skip an all-invalid batch or its optimizer step. We supervise the prior mean with

$$\mathcal{L}_{\text{prior}} = \frac{1}{D_{\mathcal{M}}} \sum \mathcal{M} \odot \text{SmoothL1}(\mu_p, Y), \tag{24}$$

and a heteroscedastic likelihood following input-dependent aleatoric uncertainty modeling (Kendall & Gal, 2017)

$$\mathcal{L}_{\text{nll}} = \frac{1}{2D_{\mathcal{M}}} \sum \mathcal{M} \odot \left(\ell_p + (Y - \mu_p)^2 \exp(-\ell_p)\right). \tag{25}$$

The point loss is applied to this prior mean during training. The deterministic central-source forecast used at inference is obtained by applying the learned transport to μ_p, so the prior loss stabilizes the anchor while the flow loss learns full-state transport. If the prior is used alone, STGraphFM reduces to a deterministic point forecaster without transported samples. The component analysis in Appendix C tests what the stochastic path adds beyond this strong anchor.

B.3 Graph-Correlated Source Around the Predictive Anchor

Flow matching permits conditional source choices beyond a fixed isotropic Gaussian in many settings (Tong et al., 2024; Kollovieh et al., 2025). We construct graph-correlated source states around the predictive anchor, while separating the training source scale from the inference-time source temperature. The mixed graph perturbation is

$$\eta_G = \text{Renorm}((1-\rho)\epsilon + \rho\ \text{Renorm}(\mathcal{S}_A(\epsilon))), \tag{26}$$

where $\epsilon \sim \mathcal{N}(0, I)$, $\mathcal{S}_A$ is a row-normalized graph smoothing operator, and ρ controls graph-source mixing. For each batch example b, Renorm rescales over all node, horizon, and output-channel entries:

$$\text{Renorm}(z)_b = z_b \frac{\max\{\text{std}_{N,T,d_y}(\epsilon_b), 10^{-6}\}}{\max\{\text{std}_{N,T,d_y}(z_b), 10^{-6}\}}. \tag{27}$$

The empirical standard deviation uses denominator $NTd_y - 1$ (correction one), without reducing across the batch. The operation rescales z_b without subtracting its mean. The smoothed component is normalized before mixing, and the complete mixed perturbation is normalized again against the same ϵ_b. Let s_p be either an all-ones tensor or a clipped scale derived from ℓ_p: Concretely,

$$s_p = \begin{cases} \text{clip}\left(\exp(0.5\,\text{sg}(\ell_p)), s_{\min}, s_{\max}\right), & \text{with prior scale,} \\ 1, & \text{otherwise,} \end{cases} \tag{28}$$

The effective source scale is $\sigma_s = \sigma_0 s_p$, where σ_0 is the configured global source-noise scale and $s_{\min}, s_{\max}$ bound its heteroscedastic modulation. During training, the predictive anchor is detached from the flow branch and the source state is

$$x_0 = c(\text{sg}(\mu_p) + a\ \text{sg}(\sigma_s) \odot \eta_G), \tag{29}$$

where $c(z) = \text{clip}(z, -8, 8)$ and a is an optional per-example training-time source-scale augmentation. This stop-gradient keeps the division of labor explicit: the prior parameters are trained by Eqs. 24–25, while the flow loss trains transport around the resulting predictive anchor. During inference, source temperature τ multiplies this source radius:

$$x_0^{(m)} = c\Big(\mu_p + \tau\sigma_s \odot \eta_G^{(m)}\Big). \tag{30}$$

The implementation supports isotropic, graph-only, and mixed modes, and re-normalizes smoothed and mixed perturbations to avoid variance collapse. The support matrix is built from the provided adjacency in raw or symmetrized mode, then row-normalized with a self-loop. All experiments use this row-normalized support unless otherwise stated. For the main sensor configuration, $\rho \sim \mathcal{U}[0.1, 0.5]$ during training and is fixed to 0.3 during inference.

Graph smoothing introduces spatial correlation before any velocity evaluation, while the independent component retains local variation. This supplies a structured initial ensemble that the velocity field can then transform using the path state and conditions; optional s_p modulates its amplitude across node-horizon pairs. The source analysis in Appendix C contrasts the mixed graph-correlated source with an isotropic source.

This source distribution encodes spatial structure in the initial variation around the predictive anchor. Source temperature multiplies the perturbation amplitude before state clipping and transport. The temperature effect is therefore applied inside source construction; any separate affine sample calibration defined below is fixed across τ.

B.4 Conditional Flow Matching

Following conditional flow matching (CFM) (Lipman et al., 2023; Tong et al., 2024), let $x_1 = c(Y)$ denote the clipped target future graph state. For $s \sim \mathcal{U}(0,1)$, STGraphFM uses the linear conditional path

$$x_s = (1-s)x_0 + sx_1, \tag{31}$$

with target velocity

$$u^\star \equiv u_s(x_s \mid x_0, x_1, C) = x_1 - x_0. \tag{32}$$

With source-scale augmentation, each example is weighted by $w(a) = \min\{w_{\max}, \max\{w_{\min}, a^{-2}\}\}$; otherwise $w(a) = 1$. Let C collect future covariates, graph support, static or external conditions, and the detached predictive anchor. Define $c(z) = \mathrm{clip}(z, -8, 8)$, $u_c = c(u^\star)$, and $\bar{v}_\theta = c(v_\theta)$. The reported graph-time field uses a bounded velocity-regression surrogate:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\frac{1}{D_{\mathcal{M}}}\left\|\sqrt{w(a)} \odot \mathcal{M} \odot (\bar{v}_\theta(x_s, s \mid C) - \mathrm{sg}(u_c))\right\|_2^2\right], \tag{33}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The path derivative is $x_1 - x_0$, while the supervised velocity is $c(x_1 - x_0)$; the loss is therefore a bounded CFM surrogate. The clips bound normalized states and residuals used in training. They also change optimization: for one valid coordinate with $|v| > 8$, the direct output derivative of $(c(v) - u_c)^2$ is zero, even if the clipped prediction disagrees with the target. Other coordinates can still update shared parameters. Decoding uses the raw velocity and clips the updated state, so this training clip does not bound the inference displacement. The comparisons fix these clipping choices. A fixed-checkpoint probe quantifies activation rates for the matched models (Appendix F.1); it does not retrain models with alternative clipping thresholds. The full objective is

$$\mathcal{L} = \lambda_p \mathcal{L}_{\mathrm{prior}} + \lambda_{\mathrm{nll}} \mathcal{L}_{\mathrm{nll}} + \lambda_{\mathrm{FM}} \mathcal{L}_{\mathrm{FM}}. \tag{34}$$

When the likelihood head is disabled, $\mathcal{L}_{\mathrm{nll}}$ is set to zero. The implementation supports a prior warm-up stage, but the prior and flow are trained jointly unless otherwise stated. During flow training, the source remains centered on the predictive anchor, while its perturbations expose the velocity field to alternative states.

Because x_0 is anchored near μ_p, the full-state velocity describes transport from the anchor's neighborhood to the realized future. The same field maps μ_p to a deterministic trajectory and graph-perturbed sources to future samples. Source and graph-branch ablations assess the initialization and message-passing components. Appendix E separately compares sample-state-dependent transport with a sample-shared correction $g(C)$ while matching conditions and capacity.

B.5 Graph-Time Conditional Velocity Field

Token construction. STGraphFM represents each node-horizon pair (i, h) as a graph-time token. Each token fuses the projected path state, future time covariates, detached predictive anchor, and condition embeddings:

$$\begin{aligned} z_{i,h}^{\mathrm{state}} &= \gamma_x \,\mathrm{GELU}(W_{\mathrm{in}}[x_{s,i,h}; C_{f,i,h}; \mathrm{sg}(\mu_{p,i,h})]), \\ z_{i,h} &= F_{\mathrm{fuse}}\left(z_{i,h}^{\mathrm{state}} + \gamma_s e_s + \sum_{q \in \{\mathrm{node,rhythm,static,ext}\}} \gamma_q e_{i,h}^q\right). \end{aligned} \tag{35}$$

Here e_s embeds path time, and the other embeddings encode node identity, calendar rhythm, static graph coordinates, and external inputs. Scalars γ set token scales. The fusion network F_{fuse} is layer normalization (LN), a linear projection, SiLU, and a second linear projection. Static and external tokens are included when enabled. Condition dropout can replace optional static or weather tokens with learned null tokens during training. The graph branch is instead attenuated: with probability 0.1, its support scale is reduced from 1.0 to 0.1 rather than removed, so candidate ranking and a residual graph bias remain. The default inference path uses the full conditions with one velocity evaluation per flow step.

Condition routing. The condition routing follows the roles of the variables rather than concatenating all information into one opaque vector. The current path state tells the velocity field where the sample is along the transport path. The predictive anchor gives the reference level at each node and lead. Future time covariates and rhythm embeddings tell it which horizon phase the update belongs to. Node and static tokens provide persistent heterogeneity, while external tokens represent weather or other boundary drivers when they exist in the dataset. These tensors are fused into node-horizon tokens before attention, whereas graph support is passed separately to the neighborhood-restricted graph branch. This keeps token features and message-passing support distinct: conditions determine what a token means, while graph support scores rank and bias candidate messages.

Factorized attention. The velocity backbone uses self-attention primitives (Vaswani et al., 2017), but avoids dense attention over all NT node-horizon tokens. Each block first applies temporal attention independently within each node's forecast horizon, then applies top-K_g graph attention independently at each horizon, followed by refinement through a feed-forward network (FFN). Suppressing pre-normalization and residual notation:

$$Z' = \mathrm{FFN}\left(\mathrm{GraphAttn}_{\mathcal{N}_{K_g}}\left(\mathrm{TempAttn}(Z)\right)\right). \tag{36}$$

Temporal-attention logits include the fixed lag bias $d_{hh'} = -0.15|h - h'|$. Before each temporal or graph update U is added to its residual query state Q, the enabled query-dependent head gate uses LN:

$$g_r(Q) = \mathrm{clip}\left(\frac{\mathrm{sigmoid}(\mathrm{MLP}_r(\mathrm{LN}(Q)))}{\mathrm{sigmoid}(2)}, 0.1, 1.5\right), \qquad U_r \leftarrow g_r(Q)U_r. \tag{37}$$

The final gate layer is initialized with zero weights and bias 2, so the initial normalized gate is one; the reported strength is one. Each block uses pre-norm residual temporal, graph, and FFN updates, FFN ratio 2, and attention dropout 0.05.

Top-K_g support-ranked candidates. Support-ranked attention is related to the Graph Attention Network (GAT) (Veličković et al., 2018). The provided adjacency is represented as a dense tensor, optionally symmetrized, and row-normalized after adding self-loops. Let $\bar{A} = \mathrm{RowNorm}(A + I)$. In the reported implementation the up/down cache entries are both $\bar{A}$, so the candidate score is

$$s_{ij} = 2\bar{A}_{ij} + 0.1\mathbf{1}[i = j], \tag{38}$$

and graph-bias attenuation multiplies the $2\bar{A}_{ij}$ term by 0.1. Each node attends to the top-K_g entries over the dense score row. Support therefore ranks and biases candidates rather than hard-masking nonedges; if a row has fewer than K_g positive entries, zero-score fillers may be selected. These fillers participate in attention; hence removing this branch removes both support-edge interactions and any selected nonedge interactions. The ablation therefore evaluates the support-ranked branch, rather than isolating physical-edge propagation. The selected nonedge fraction is not recorded in the reported tables. For a selected candidate, b_{ij} in Eq. 39 is the row-max-centered score $s_{ij} - \max_{j' \in \mathcal{N}_{K_g}(i)} s_{ij'}$, clipped to $[-6, 0]$. Here r indexes an attention head, d_r is its width, and D is the total token width. Within this candidate set, attention remains state-adaptive. This design is motivated by the observation that inter-series relations evolve over time, which is central to dynamic-graph time-series forecasting (Zhao et al., 2023):

$$\begin{aligned} \alpha_{ij}^{(h,r)} &= \mathrm{softmax}_{j \in \mathcal{N}_{K_g}(i)}\left(\frac{q_{i,h,r}^{\top} k_{j,h,r}}{\sqrt{d_r}} + b_{ij}\right), \\ z'_{i,h,r} &= \sum_{j \in \mathcal{N}_{K_g}(i)} \alpha_{ij}^{(h,r)} v_{j,h,r}. \end{aligned} \tag{39}$$

Support scores rank and bias the top-K_g candidate set, while dot-product attention remains state-adaptive inside that set. This is Transformer-style support-ranked propagation rather than additive GAT: samples share structural candidates, but their attention weights can differ with path state and conditions. The reported implementation materializes the $N \times N$ support score and runs row-wise `topk` once per velocity evaluation; candidates are shared across blocks but not cached across samples or flow steps. Thus, attention uses NK_g candidate slots after an N^2 selection stage. The component analysis in Appendix C tests this branch through the no-graph-attention and isotropic-source variants.

A dense all-token layer costs $O(BN^2T^2D)$. STGraphFM uses $O(BNT^2D)$ temporal attention, $O(BTNK_gD)$ neighborhood attention, and approximately $O(N^2 \log K_g)$ candidate selection with $O(N^2)$ support storage. It avoids dense NT-token attention but is not end-to-end sparse or linear in N. Appendix C measures flow-step, sample-count, large-graph predictive quality, and fixed-hardware behavior under this implementation.

B.6 Deterministic and Stochastic Generation

At inference time, one trained model exposes both a deterministic path and a stochastic generation path. The implementation uses few-step explicit transport rather than a black-box adaptive ODE solver. Given an initial state x_0 and step count K_{step}, the reported update is

$$x_{k+1} = c\left(x_k + \frac{1}{K_{\text{step}}} v_\theta(x_k, s_k \mid C_f, A^{\text{sup}}, C_s, E, \mu_p)\right), \quad s_k = \frac{k+0.5}{K_{\text{step}}}, \tag{40}$$

This update evaluates the interval's starting state x_k with its midpoint time token s_k; it does not predict an intermediate state and is not the second-order midpoint method. Each step uses one velocity evaluation, giving $x_0 + v_\theta(x_0, 0.5 \mid C)$ before clipping when $K_{\text{step}} = 1$. Sample initialization and every updated state are clipped to $[-8, 8]$. The central-source initialization is the unmodified prior mean; decoding uses the raw velocity field, so the training velocity clip does not impose a hard bound of eight on the pre-clipping displacement. For deterministic forecasting, STGraphFM sets $x_0 = \mu_p$ and applies Eq. 40:

$$\hat{Y}_{\text{cs}} = \text{Transport}_{K_{\text{step}}}(\mu_p). \tag{41}$$

We call $\hat{Y}_{\text{cs}}$ the *central-source forecast* because its source is the prior mean; this deterministic path is separate from the N_s random samples. We use this trajectory as the deterministic forecast for Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) evaluation. For probabilistic generation, STGraphFM samples multiple prior-anchored source states and transports each one:

$$\widetilde{Y}^{(m)} = \text{Transport}_{K_{\text{step}}}\left(x_0^{(m)}\right), \quad x_0^{(m)} = c\left(\mu_p + \tau\sigma_s \odot \eta_G^{(m)}\right). \tag{42}$$

When enabled, a pre-test-frozen affine calibration maps raw samples to

$$\hat{Y}^{(m)} = \hat{Y}_{\text{cs}} + a_c\left[\widetilde{Y}^{(m)} + \rho_c\left(\hat{Y}_{\text{cs}} - \overline{\widetilde{Y}}\right) - \hat{Y}_{\text{cs}}\right], \quad \overline{\widetilde{Y}} = \frac{1}{N_s}\sum_m \widetilde{Y}^{(m)}. \tag{43}$$

Here $\rho_c \in [0, 1]$ controls center alignment and $a_c > 0$ controls spread. The reported configuration fixes $\rho_c = 0.25$ and selects a_c on validation data; both are frozen before testing. This affine map is centered on $\hat{Y}_{\text{cs}}$: its ensemble mean is $\hat{Y}_{\text{cs}} + a_c(1 - \rho_c)(\overline{\widetilde{Y}} - \hat{Y}_{\text{cs}})$, so changing a_c also changes any remaining center offset. The choice $\rho_c = 0, a_c = 1$ preserves raw samples. The central-source forecast is used for MAE/RMSE/MAPE, while the sample set $\{\hat{Y}^{(m)}\}_{m=1}^{N_s}$ supports CRPS, interval metrics, uncertainty maps, tail-event probabilities, and scenario inspection. The inference knobs are explicit: N_s controls sample budget, τ controls the source radius, and K_{step} controls transport cost. The reported flow-step sweep in Appendix C selects one step as the default from the measured accuracy–latency trade-off.

B.7 Training, Inference, and Cost Model

Algorithm 1 summarizes the complete training and inference procedure. Training uses (X_h, Y, Ω) to supervise the prior and velocity field; after training, inference requires only (X_h, Ω) and the selected sample-count, source-temperature, and flow-step settings.

Algorithm 1 Prior-anchored training and inference

Require: History and conditions (X_h, Ω); graph support; source and transport settings
Training: repeat for each batch $(X_h, \Omega, Y, \mathcal{M})$
Predict (μ_p, ℓ_p) and σ_s; compute masked prior and likelihood losses
Draw coordinate-wise ϵ, batch-shared ρ, and per-example a, s
Detach μ_p, σ_s in the source and velocity conditions; form x_0, x_1, x_s
Evaluate $v_\theta(x_s, s \mid C)$ and masked, radius-weighted $\mathcal{L}_{\mathrm{FM}}$
Backpropagate Eq. 34; take one AdamW step on both parameter groups
Prior receives prior/likelihood gradients; velocity receives FM gradients
Inference
Predict (μ_p, ℓ_p); transport μ_p to obtain $\hat{Y}_{\mathrm{cs}}$
Draw N_s graph-correlated perturbations at the fixed inference mixture
Transport their clipped sources by Eqs. 40 and 42
Apply the validation-fixed output map; return $\hat{Y}_{\mathrm{cs}}$ and samples

Computational cost. Let D be the hidden dimension and K_g be the graph-neighbor budget. One graph-time velocity evaluation costs approximately

$$C_{\mathrm{flow}} = O(N^2 \log K_g) + O(BNT^2D) + O(BTNK_gD) + O(BNTD^2), \tag{44}$$

corresponding to dense candidate selection, temporal attention, top-K_g graph attention, and token projections. The deterministic path costs

$$C_{\mathrm{det}} \approx C_{\mathrm{prior}} + K_{\mathrm{step}} C_{\mathrm{flow}}, \tag{45}$$

and an N_s-sample probabilistic run costs

$$C_{\mathrm{sample}} \approx C_{\mathrm{prior}} + (1 + N_s) K_{\mathrm{step}} C_{\mathrm{flow}}, \tag{46}$$

when the central-source forecast is also decoded. The current implementation transports samples sequentially to keep peak memory bounded; vectorized sample transport can reduce wall-clock latency when memory permits. The optional affine sample calibration adds only $O(BN_sNTd_y)$ elementwise work after transport. These expressions explain the computational budget: point metrics are independent of N_s, additional transported samples refine distribution estimates, and extra flow steps increase the number of velocity evaluations. This cost model retains the dense selection overhead instead of attributing the observed runtime solely to sparse aggregation.

C Experimental Protocol and Additional Results

The evaluation separates predictive quality, mechanism, and resource use. This section gives the shared protocols, benchmark results, component analyses, and application queries. Appendix E tests the training objective and sample-state dependence, including comparisons at fixed empirical marginals; Appendix F examines output and source sensitivity. The four-member endpoint-score study in Appendix G provides a supplementary comparison under a different training and output protocol.

C.1 Experimental Setup

Datasets. We use three sensor-network benchmarks, one global environmental field, and one large-graph stress test. PEMS03 is a 358-node traffic graph (Song et al., 2020); LargeST-SD is a 716-node San Diego traffic graph (Liu et al., 2023d); and KnowAirV2 is a 228-node air-quality graph for PM2.5 forecasting (Wang et al., 2020; 2025). WeatherBench2 is a large multivariable global weather archive (Rasp et al., 2024); we evaluate its ERA5 two-meter air temperature (T2M), measured in kelvin (K), on a 64×32 latitude–longitude grid. Each grid cell is a node, and the fixed support connects its four horizontal or vertical grid neighbors, with cyclic longitude and no latitudinal wrap. This yields a 2,048-node field at six-hour resolution; eight history steps condition a 40-step (240-hour) forecast. The year-disjoint split uses 1979–2015 for training, 2016 for validation, and 2017–2018 for testing. We select T2M because it forms a dense, spatially coupled global field and supports a controlled test of cross-domain applicability, graph-size behavior, and ten-day generation without introducing model-specific multivariable weather encoders. This is therefore a single-variable

WeatherBench2 evaluation rather than a complete weather-forecasting benchmark. LargeST-GBA contains 2,359 Bay Area traffic nodes and evaluates large-graph predictive quality. LargeST-CA supplies independently materialized graphs with 1,000–5,000 nodes for evaluating predictive behavior and computational cost across graph sizes. All benchmark rows use train-only normalization. Tables 6 and 7 collect the evaluation settings and dataset schedules.[1]

Baselines. We compare with deterministic spatiotemporal graph forecasting baselines and probabilistic or generative forecasting baselines. Deterministic baselines include Spatial-Temporal Identity (STID) (Shao et al., 2022a), GraphWaveNet (Wu et al., 2019), Adaptive Graph Convolutional Recurrent Network (AGCRN) (Bai et al., 2020), Spatio-Temporal Adaptive Embedding Transformer (STAEformer) (Liu et al., 2023a), and GraphSparseNet (GSNet) (Kong et al., 2025); these models test whether STGraphFM preserves a competitive deterministic-forecasting interface. Probabilistic baselines include SpecSTG (Lin et al., 2025), TSFlow (Kollovieh et al., 2025), PriSTI-F (Liu et al., 2023b), Conditional Score-based Diffusion models for Imputation (CSDI) (Tashiro et al., 2021), and Deep Spatio-Temporal Uncertainty Quantification (DeepSTUQ) (Qian et al., 2023), which produce predictive samples or intervals. The suffix F in PriSTI-F denotes our forecasting adaptation of PriSTI. Both it and CSDI condition on the observed history, mask the entire future window, and evaluate generated values only in that future window. WeatherBench2 additionally uses two non-learned references: calendar climatology predicts the train-period mean T2M at the corresponding calendar phase, whereas persistence repeats the latest observed T2M field throughout the forecast horizon. The WeatherBench2 comparison includes GraphWaveNet, STID, and STAEformer for point forecasts and TSFlow for sample forecasts. Available-material details are collected in Appendix D.

Table 6: Index of evaluation settings. Each row identifies its own window set, training replicates, and output settings; S denotes generated members. Repeated LargeST-SD means in Tables 2 and 13 count once as evidence.

Analysis	Data and checkpoint scope	Sampling and calibration
Sensor forecasting and components (Tables 1, 2)	Three training seeds per reported row; reference dataset schedules in Table 7	$S = 16$, one step; reported affine operating points
Symmetric traffic calibration	PEMS03/LargeST-SD; 512 validation and 512 test windows; fixed checkpoint per model	$S = 16$; common scale grid, validation coverage selection
Scenario queries (Table 13)	PEMS03/LargeST-SD; reported 512 aligned test windows	$S = 16$; validation selects $\tau = 1.2$ (STGraphFM) and TSFlow spread scale
Traffic temperature sweep	Same 512-window test setting and fixed random draws	$S = 16$, one step; vary source temperature, freeze output calibration
LargeST-SD cost sweeps	1,024 test windows; one checkpoint	Vary S or steps with the other settings fixed
LargeST-SD case study (Fig. 5)	6,976 windows; one checkpoint; displayed window 538	$S = 16$; $\tau = 0.3, 0.7, 1.0$; fixed output calibration
WeatherBench2 (Fig. 4)	721 test origins per seed (42, 2024, 3407); 357 validation origins at seed 42	$S = 16$; frozen scales 2.0 (STGraphFM), 1.5 (TSFlow)
LargeST-GBA / CA	GBA budgets in Table 9; CA: three seeds, independent graphs, 15 epochs	CA: $S = 16$; GBA probabilistic references use 256 test windows
Risk and aligned dependence (Table 3)	KnowAirV2; seeds 20261001/20261002; five epochs; 14,003 test windows each	$S = 16$; SpecSTG risk comparison uses validation-frozen scaling; Score16 uses exact common marginals
Supplementary Score4 (Appendix G)	Seed 20260930; 14,003 test windows; checkpoints 6/12 from 22/21 epochs	Train $S = 4$ for score control; test $S = 16$; center alignment and median scaling
Same-device cost (Table 4)	RTX5090D; 256 training windows and 64 validation windows; three passes	Train batch 8; inference batch 2, $S = 16$; synchronized wall time

[1]Resources: PEMS03, `https://github.com/Davidham3/STSGCN`; LargeST (SD/GBA/CA), `https://github.com/liuxu77/LargeST`; raw data, `https://www.kaggle.com/datasets/liuxu77/largest`; KnowAirV2, `https://doi.org/10.5281/zenodo.15614907`; WeatherBench2, `https://weatherbench2.readthedocs.io/en/latest/data-guide.html`.

Table 7: Dataset schedules and reference model configurations. Window counts are available loader windows before run-specific caps.

	PEMS03	LargeST-SD	KnowAirV2	WeatherBench2
Nodes	358	716	228	2,048
Interval	5 min	15 min	1 h	6 h
History / future steps	12 / 12	12 / 12	12 / 12	8 / 40
Train/val/test split	60/20/20%	60/20/20%	70/10/20%	Year split[†]
Available train windows	15,700	21,010	—	—
Available val windows	5,218	7,003	—	—
Available test windows	5,218	7,004	—	—
Prior hidden width	160	240	160	—
Prior projection width	64	96	64	—
Node / TOD embeddings	32 / 32	48 / 48	32 / 32	—
DOW / horizon embeddings	16 / 16	24 / 24	16 / 16	—
Prior MLP blocks	2	3	2	—
Velocity width / heads	64 / 4	64 / 4	64 / 4	—
Graph-time blocks	3	3	2	—
Graph candidates / static width	8 / 8	8 / 8	8 / 8	—

[†]WeatherBench2 uses 1979–2015/2016/2017–2018. PEMS03 spans 2018-09-01–2018-11-30 with split indices $[0, 15724)$, $[15724, 20966)$, $[20966, 26208)$. LargeST-SD uses 35,040 processed steps; issue-index ranges are 11–21,020 / 21,021–28,023 / 28,024–35,027. TOD/DOW mean time of day/day of week. Aligned controls are specified in Appendix E; supplementary Score4 uses Appendix G.

Evaluation metrics. Point accuracy is evaluated with MAE, RMSE, and MAPE on the inverse-normalized scale. Let $\mathcal{I}$ denote all valid test entries over windows, nodes, horizons, and variables, with $|\mathcal{I}|$ entries. For a deterministic prediction $\hat{y}_i$ and ground truth y_i, we compute

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}|\hat{y}_i - y_i|, \\ \mathrm{RMSE} &= \left(\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}(\hat{y}_i - y_i)^2\right)^{1/2}, \\ \mathrm{MAPE} &= \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\frac{|\hat{y}_i - y_i|}{\max(|y_i|, \epsilon)}, \end{aligned} \tag{47}$$

where ϵ is a small constant for numerical stability.

Predictive distributions are evaluated with the Continuous Ranked Probability Score (CRPS), the 80% Prediction Interval Coverage Probability (PICP_{80}), and the 80% Mean Prediction Interval Width (MPIW_{80}). Given N_s generated samples $\{\hat{y}_i^{(m)}\}_{m=1}^{N_s}$ for entry i, the empirical CRPS is

$$\begin{aligned} \mathrm{CRPS} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\Bigg[&\frac{1}{N_s}\sum_{m=1}^{N_s}|\hat{y}_i^{(m)} - y_i| \\ &- \frac{1}{2N_s^2}\sum_{m=1}^{N_s}\sum_{m'=1}^{N_s}|\hat{y}_i^{(m)} - \hat{y}_i^{(m')}|\Bigg]. \end{aligned} \tag{48}$$

CRPS is the primary proper scoring rule for generated samples (Matheson & Winkler, 1976; Gneiting & Raftery, 2007). The N_s^2 denominator computes CRPS of the empirical ensemble distribution; as a Monte Carlo estimate of the underlying predictive distribution's CRPS, it has finite-ensemble bias. Endpoint-score training in Appendix E uses denominator $N_s(N_s - 1)$ over distinct sample pairs; its center-alignment rule is specified there.

Table 8: Validation-selected calibration on aligned traffic windows. Bold and underline indicate best and second-best values; PICP_{80} is ranked by proximity to 0.8.

Dataset	Model	Scale	CRPS↓	PICP_{80}	MPIW_{80} ↓
PEMS03	STGraphFM	1.10	**11.118**	**0.802**	**47.154**
	SpecSTG	2.75	34.475	0.827	154.132
	TSFlow	1.80	<u>15.893</u>	<u>0.803</u>	<u>71.710</u>
LargeST-SD	STGraphFM	1.00	**12.455**	0.777	**50.277**
	SpecSTG	4.00	89.449	<u>0.812</u>	434.925
	TSFlow	2.75	<u>40.349</u>	**0.794**	<u>173.173</u>

Let $q_{0.1,i}$ and $q_{0.9,i}$ be the empirical 10th and 90th percentiles of the generated samples. The 80% interval metrics are

$$\begin{aligned} \text{PICP}_{80} &= \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \mathbf{1}\{q_{0.1,i} \le y_i \le q_{0.9,i}\}, \\ \text{MPIW}_{80} &= \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} (q_{0.9,i} - q_{0.1,i}). \end{aligned} \tag{49}$$

PICP_{80} measures interval coverage, whereas MPIW_{80} measures sharpness. Overly wide intervals can inflate coverage, while narrow intervals can under-cover; CRPS summarizes overall distributional accuracy. Ranked tables use bold and underlined entries for the best and second-best distinct reported means within each dataset or panel, sharing ranks for ties; PICP_{80} is ranked by absolute deviation from 0.8. MPIW_{80} is ranked in ascending order. Two-model comparisons mark only the best value. Heterogeneous-budget or different-graph rows are not ranked.

Protocol and scope. For STGraphFM, deterministic metrics are computed from the central-source forecast $\hat{Y}_{\text{cs}}$, while probabilistic metrics are computed from generated samples. Results use multi-seed means and standard deviations whenever available. The main rows use standard dataset splits, train-only normalization, and inverse-normalized evaluation; WeatherBench2 uses the year boundaries above and latitude-weighted scores in kelvin. We train STGraphFM with AdamW and select checkpoints and distribution scales on validation data. Any configured sample recentering or spread scale is fixed after validation; temperature sweeps vary only the pre-transport source radius. The sensor-network main table reports each model at its reported operating point and is not coverage matched. A separate symmetric calibration evaluation compares STGraphFM, SpecSTG, and TSFlow on 512 validation windows and 512 test windows per dataset, aligned across models, for PEMS03 and LargeST-SD; KnowAirV2 is interpreted only at the reported operating points. This supplementary evaluation uses one checkpoint per model and dataset, separately from the main table's multi-seed results. At $N_s = 16$ and calibration sampling seed 160016, each model selects one sample-spread scale from the shared grid $\{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.4, 1.6, 1.8, 2, 2.25, 2.5, 2.75, 3, 3.5, 4\}$ by minimizing validation $|\text{PICP}_{80} - 0.8|$; MPIW_{80} and then scale break exact ties. Each scale is frozen before testing. Table 8 reports all three models at $N_s = 16$ with test sampling seed 1616: STGraphFM combines the lowest CRPS and narrowest intervals with coverage of 77.7%–80.2%. Budgets $N_s \in \{8, 16, 32, 64\}$ use test generation seeds $\{8008, 1616, 3232, 6464\}$, respectively, shared across models. Monte Carlo standard errors use 64 nonparametric member-bootstrap replicates on a fixed 8,192-valid-point subsample. Window uncertainty uses 1,000 moving-block bootstrap replicates with block length 24 and 95% percentile intervals; neither quantity is training-seed uncertainty. WeatherBench2 uses 721 stride-4 test origins per seed. A 357-origin seed-42 validation sweep selects one scale per model (2.0 for STGraphFM and 1.5 for TSFlow), frozen for test seeds 42, 2024, and 3407, each with 721 origins. Unless a sweep states otherwise, probabilistic evaluation uses $N_s = 16$ samples, one flow step, and a top-$K_g = 8$ neighborhood budget; deterministic evaluation uses the central-source path. Implementation versions are specified in Appendix D.

Implementation details. The following settings describe the sensor-benchmark configuration. The sensor-network models use full-state transport, four attention heads, and one graph-smoothing pass for source construction. The graph-time backbone has three blocks on PEMS03 and LargeST-SD and two on KnowAirV2; every block uses the same top-$K_g = 8$ candidate budget. The mixed source combines isotropic and graph-smoothed noise, re-normalizes the smoothed component and

the complete mixture, and samples its mixture coefficient uniformly from $[0.1, 0.5]$ during training; inference fixes it to 0.3. The source radius is sampled log-uniformly from $[0.7, 1.5]$ on PEMS03 and $[0.7, 1.7]$ on LargeST-SD and KnowAirV2; the prior scale is clipped to $[0.05, 0.35]$, $[0.05, 0.40]$, and $[0.05, 0.45]$, respectively. The prior, likelihood, and flow-matching loss weights are 1.0, 0.05, and 1.0; source-scale weights are clipped to $[0.5, 2.0]$, and path states plus target and predicted velocities to $[-8, 8]$. The prior log variance is clipped to $[-8, 4]$, the global source scale is one, attention dropout is 0.05, and the FFN expansion ratio is two. Static-context dropout is 0.2; with probability 0.1, graph-bias attenuation uses support scale 0.1 instead of 1.0. These settings are selected with validation data and then held fixed for test evaluation.

Computational environment. Main benchmark runs use one vGPU-32GB device, 16 Intel Xeon Platinum 8352V vCPUs, and 60GB system memory. Resource measurements use batch size 16, top-$K_g = 8$, and one flow step; latency is the median per-window time and memory is peak process GPU allocation. The training peak covers the complete optimization path and therefore represents the measured device-memory requirement of this configuration.

Same-device inference cost. Table 4 uses a separate matched KnowAirV2 measurement: the same 64 validation windows, batch size 2, 16 samples, and one exclusively used NVIDIA RTX 5090 D. The main table compares CSDI, SpecSTG and STGraphFM; the FM/DDPM adapters below use the same protocol. All five measured models use PyTorch 2.7.0, CUDA 12.8, four CPU threads, FP32 tensors, deterministic execution, and no AMP or compilation. Matrix-multiplication TF32 is disabled; cuDNN retains its default policy except for the equivalent dilated kernels described below. Each model is warmed up before three passes of 32 batches, with CUDA synchronization around every timed batch. Time includes sample generation, transfer to CPU, and inverse normalization; batch preparation, disk writes, and scoring are excluded. We report the mean batch time divided by two. Each sampler retains its member-batching implementation. STGraphFM includes its central forecast and uses one velocity evaluation per sample; CSDI uses 20 reverse steps, and SpecSTG uses 10 steps at each of 12 autoregressive leads. FM uses eight Heun steps (16 velocity evaluations), and DDPM uses 20 respaced steps over its full 1,000-step training schedule. The standard deviations of the three per-window timing means are $0.27/0.08/6.26/6.85/0.36$ ms for FM, DDPM, CSDI, SpecSTG, and STGraphFM. The frozen seed-20261001 checkpoints come from five-epoch runs except for CSDI (20 epochs). These are implementation-level sampling costs at the specified operating points; shared-device training times are not used in this comparison.

FM and DDPM adapters. The two additional baselines share a graph-free temporal convolutional backbone with 114,049 parameters: six residual blocks of width 64, 12 historical targets, and known calendar features. Both have complete 14,003-window KnowAirV2 evaluations with 16 samples. Their calibrated CRPS values are 10.363 (FM) and 10.309 (DDPM), compared with 10.210 for the five-epoch STGraphFM checkpoint. They are algorithm adapters, rather than the native spatial baselines in Table 1. On this GPU/software combination, deterministic dilated convolutions fall back to a slow execution path. For timing, we express the same convolution using zero-inserted dense kernels, with TF32 disabled for these kernels. Trainable tensors, receptive fields, objectives, and samplers remain unchanged. Checks against the original implementation on real inputs and frozen checkpoints give maximum gradient differences below 2.2×10^{-6} and relative sample differences below 3.2×10^{-5}; repeated optimized draws are bitwise reproducible. The unoptimized fallback is excluded from efficiency claims. FM and DDPM take 21.29 and 25.39 ms per window, with training updates of 22.15 and 20.48 ms, respectively. These smaller graph-free implementations are faster than STGraphFM; the main-table speedups specifically concern CSDI and SpecSTG.

Same-device training cost. Training costs in Table 4 use the same device and software environment, batch size 8, and 256 evenly spaced training windows. Each of three profiling passes resets to the same seeded initialization, then runs five warm-up updates followed by 32 timed updates. CUDA-synchronized timing covers the forward pass, training loss, backward pass, gradient clipping, and optimizer step; batch preparation and validation are excluded. Standard deviations across the three pass means are $1.91/3.09/1.42/3.16/6.01$ ms for FM, DDPM, CSDI, SpecSTG, and STGraphFM. These disposable updates retain each model's training objective and do not change the prediction checkpoints. The measure is cost per update, not total time to convergence.

Scenario and structural protocol. The traffic comparison uses 512 aligned test windows and $S = N_s = 16$ samples. Queries use target channel zero. Node thresholds q_i are its training-set 90th percentiles; the preparation script takes the first 60% of the full chronological series, which is the training split. A target event is $e_{bi} = \mathbf{1}\{\exists h : \mathcal{M}_{bih} = 1,\ Y_{bih} > q_i\}$, where $\mathcal{M}$ requires a finite target and missing-mask value below 0.5. Only nodes with at least one valid lead enter risk scoring. The historical traffic results use valid leads for target events and all leads for sample maxima in Eq. 16. The revised evaluator applies the same valid-lead mask to both events. The main event comparison uses the complete KnowAirV2 windows described below.

Complete KnowAirV2 risk comparison. Table 3(a) uses the matched-input setting, two five-epoch training seeds, all 14,003 test windows, and $S = 16$. The STGraphFM rows score retained full-test samples; SpecSTG uses its frozen checkpoints and recorded ten-step respaced ancestral sampler. Each model's affine scale is selected on all 3,471 validation windows and frozen before testing. Window IDs, target events, and train-derived nodewise 90th-percentile thresholds are identical across models. All 12 leads of every evaluated node-window are finite in the prepared test data; both target and sampled events use these same leads. Expected-uniform-tie AP is the expected standard AP under a uniform random permutation within each of the 17 event-count groups induced by $S = 16$. Top-K precision and recall use $K = 50$; Region IoU uses the connected-region construction below. The two-seed mean P@50/R@50 is $0.3925/0.5420$ for SpecSTG and $0.4019/0.5677$ for STGraphFM.

Rank average precision and region retrieval. The evaluator concatenates all valid window–node pairs in evaluation-window order and, within each window, input-node order, then sorts this pooled list. Thus Rank AP is a pooled statistic, not a mean of per-window AP. Let $e_{(r)}$ be labels after stable descending sorting of risk, and let $P = \sum_r e_{(r)}$. The reported stable-order rank average precision (Rank AP) is

$$\mathrm{RankAP} = \frac{1}{P} \sum_{r:e_{(r)}=1} \frac{\sum_{j \le r} e_{(j)}}{r}, \qquad \mathrm{RankAP} = 0 \text{ if } P = 0. \tag{50}$$

Equal risks retain their input order. With $S = 16$, event probabilities occupy at most 17 values, so within-tie ordering can affect the score under a node relabeling or a different window order. This rank statistic differs from threshold-grouped average precision. Appendix F.1 reports grouped-tie and within-tie diagnostics on the completed KnowAirV2 predictions; the traffic results use their stated convention. Top-K uses $K = 50$; region candidates number $\max\{50, \mathrm{round}(0.15N)\}$. Both selections use `argpartition`, whose boundary-tie choice follows its index partition rather than a risk-based secondary key, so a node relabeling can also change the selected boundary candidates. The region is the largest component among positive-risk candidates on symmetrized nonzero support without self-loops, with equal-size components ranked by total risk. If all selected risks vanish, it falls back to a maximum-risk node. Region IoU is averaged only over windows with a nonempty observed-event region, after excluding invalid nodes from the predicted set. Rank AP and Region IoU are unchanged by independent cross-node permutation of whole-trajectory member IDs; they measure risk retrieval rather than cross-node member dependence.

Energy and graph variogram scores. For window b, vectorize nodes, leads, and output channels and define the masked root-mean-square (RMS) norm

$$\|z\|_{b,\mathrm{RMS}} = \left(\frac{\sum_j \mathcal{M}_{bj} z_j^2}{\max(1, \sum_j \mathcal{M}_{bj})} \right)^{1/2}. \tag{51}$$

The empirical Energy Score is averaged over windows:

$$\mathrm{ES}_b = \frac{1}{S} \sum_m \|\hat{Y}_b^{(m)} - Y_b\|_{b,\mathrm{RMS}} - \frac{1}{2S^2} \sum_{m,m'} \|\hat{Y}_b^{(m)} - \hat{Y}_b^{(m')}\|_{b,\mathrm{RMS}}. \tag{52}$$

The graph variogram score (GVS) uses exponent $p = 1/2$, equal edge weights, and both-endpoint validity $M_{bijh} = \mathcal{M}_{bih}\mathcal{M}_{bjh}$:

$$\mathrm{GVS} = \frac{\sum_{b,(i,j)\in\mathcal{E}_*,h} M_{bijh} \left[|Y_{bih} - Y_{bjh}|^{1/2} - \frac{1}{S} \sum_m |\hat{Y}_{bih}^{(m)} - \hat{Y}_{bjh}^{(m)}|^{1/2} \right]^2}{\max(1, \sum_{b,(i,j)\in\mathcal{E}_*,h} M_{bijh})}. \tag{53}$$

The sensor evaluator selects nonzero off-diagonal adjacency entries in row-major order and, when there are more than 2,048, retains 2,048 evenly spaced entries; reverse edges remain separate entries; multiple output channels, when present, are included in the valid-entry average. The matched comparison instead uses the first 4,096 upper-triangular nonzero edges, or all if fewer, and computes the masked GVS separately per window before averaging. These edge and averaging rules are fixed within each comparison. Energy and variogram scores assess joint samples (Gneiting & Raftery, 2007; Scheuerer & Hamill, 2015).

Dominant-lag agreement. The statistic denoted Lag agr. uses the ensemble mean $\bar{Y}$, channel zero, and the sensor evaluator's fixed edge set. For $Z \in \{Y, \bar{Y}\}$, define $u^Z_{bih} = \mathbf{1}\{|Z_{bih}| \geq q^Z\}$, where q^Z is the 90th percentile of absolute valid test values of Z, computed separately for the target and the prediction. For $r = 1, \ldots, L$ with $L = \min(3, T-1)$, compute

$$P^Z_{ij}(r) = \frac{\sum_{b,h \leq T-r} u^Z_{bih} \mathcal{M}_{bih} u^Z_{bj,h+r} \mathcal{M}_{bj,h+r}}{\max(1, \sum_{b,h \leq T-r} u^Z_{bih} \mathcal{M}_{bih})},$$
$$\text{LagAgr} = \frac{1}{|\mathcal{E}_*|} \sum_{(i,j) \in \mathcal{E}_*} \mathbf{1}\{\arg\max_r P^Y_{ij}(r) = \arg\max_r P^{\bar{Y}}_{ij}(r)\}. \tag{54}$$

Ties, including all-zero vectors, select lag one. An edge whose target and predicted lag vectors are both zero therefore counts as agreement despite containing no observed lag evidence. The reported score averages over the entire selected edge set; it does not subtract this contribution or provide an always-lag-one reference. It describes dominant positive-lag bins of mean forecasts. Energy and GVS, rather than this diagnostic, assess individual generated members jointly.

C.2 Main Results

The deterministic results verify that STGraphFM retains a competitive point-forecasting interface: its reported MAE is competitive across the sensor benchmarks, while specialized deterministic models retain the strongest RMSE and several MAPE results. Compared with the best reported three-seed probabilistic result, STGraphFM reduces CRPS by 18.4% on PEMS03 and 45.8% on LargeST-SD. The reduction on KnowAirV2 is 6.6%. STGraphFM also has the lowest MPIW_{80} on all three datasets. At the reported operating points, PICP_{80} is 0.735 versus 0.772, 0.725 versus 0.760, and 0.648 versus 0.784, respectively. With validation-fixed traffic calibration, it retains narrower intervals at coverage 0.802 and 0.777 (Appendix C.4), combining compact ranges with coverage close to the nominal 0.8. The temperature sweep below shows how coverage changes with scenario spread.

The complete deterministic and probabilistic sensor-network tables appear in Table 1 of the main paper.

WeatherBench2 tests whether this behavior extends beyond sensor networks to a 40-step global environmental field. STGraphFM achieves the best overall latitude-weighted deterministic scores among the compared models in Table 11. GraphWaveNet is better at 24 hours and nearly tied at 72 hours, but STGraphFM reduces its MAE by 2.6%, 4.5%, and 6.4% at 120, 168, and 240 hours. STGraphFM and validation-calibrated TSFlow obtain CRPS/PICP_{80}/MPIW_{80} of $0.990/0.773/3.501$ and $1.387/0.787/6.087$, respectively.

Table 5 summarizes the key deterministic and probabilistic comparisons. Figure 4 retains all six point-forecasting methods and the calibration diagnostics.

C.3 Forecast Quality and Resource Use on Larger Graphs

LargeST-GBA quality. LargeST-GBA evaluates deterministic and probabilistic forecasting quality on a 2,359-node road graph under the heterogeneous budgets listed in Table 9. STGraphFM has the lowest observed MAE and CRPS in that stress test; its PICP_{80}/MPIW_{80} pair is $0.736/56.95$, compared with $0.773/106.27$ for SpecSTG and $0.801/125.13$ for TSFlow. The observed row combines strong deterministic accuracy with lower CRPS and narrower intervals, while PICP_{80} remains below nominal 0.8; it does not establish a same-budget or multi-seed baseline ranking.

Independent graph-scale evaluation. We train independent models at 1,000, 2,000, 3,000, 4,000, and 5,000 nodes for each of three paired seeds. Each run uses a separately materialized induced graph,

Table 9: Heterogeneous-budget LargeST-GBA stress test. Budgets are STGraphFM: 3 seeds/8 epochs (4+4); STID: 1/15; STAEformer: 1/5 with a 4,096-window cap; GraphWaveNet/TSFlow: 1/5; SpecSTG: 1/15. Probabilistic references use 256 test windows. Values are descriptive, not a controlled budget comparison; PICP_{80} targets 0.8.

(a) Deterministic forecasting

Model	MAE↓	RMSE↓	MAPE↓
STGraphFM	21.28	34.89	0.174
STID	21.84	35.52	0.192
STAEformer	27.45	41.48	0.246
GraphWaveNet	28.14	42.95	0.257

(b) Probabilistic generation

Model	CRPS↓	PICP_{80}	MPIW_{80} ↓
STGraphFM	16.40	0.736	56.95
SpecSTG	26.35	0.773	106.27
TSFlow	29.59	0.801	125.13

Table 10: Forecasting quality and measured cost across independently trained LargeST-CA graphs. Node counts define different prediction tasks, so values are not ranked across rows. Runs converge within 1% of their best validation CRPS at epochs 13–14; resource measurements are hardware-specific.

(a) Forecasting performance

N	MAE↓	CRPS↓	PICP_{80}	MPIW_{80} ↓
1,000	22.080±.682	17.119±.529	.764±.006	64.060±1.081
2,000	23.156±.321	17.878±.263	.759±.001	66.393±1.120
3,000	23.399±.434	18.155±.442	.711±.071	59.324±10.264
4,000	22.112±.313	17.150±.241	.751±.002	61.942±1.284
5,000	22.033±.058	17.038±.046	.754±.001	62.887±.209

(b) Inference, memory, and convergence cost

N	Center ms/window	N_s=16 ms/window	Inference memory (GiB)	Training memory (GiB)	Conv. epoch	Conv. time (min)
1,000	5.283	46.384	1.400	7.172	14	11.55
2,000	10.735	94.443	2.492	13.172	13	22.62
3,000	12.918	138.891	3.182	19.084	13	34.66
4,000	18.478	185.246	4.625	25.703	14	50.31
5,000	21.274	230.506	5.701	31.092	14	64.03

random initialization, 15 epochs, 4,096 training windows per epoch, and batch size 16; checkpoint transfer and graph cropping are not used. The full CA graph's largest weakly connected component has 3,655 nodes, so the 1,000–3,000-node prefixes are connected while the 4,000- and 5,000-node graphs contain two weak components after the largest component is exhausted. All runs share the split, train-only scaler, optimization schedule, and model configuration, and quality is evaluated on the same 512 test windows with $N_s = 16$. Across this fivefold node-count range, mean MAE spans 22.033–23.399 and mean CRPS spans 17.038–18.155, corresponding to relative ranges of 6.1% and 6.4%. These ranges describe different node sets and graph prediction tasks; changing task difficulty can affect them. The runs demonstrate execution and measured quality at each size, rather than error invariance under a controlled increase in graph size. PICP_{80} is below 0.8 at every size, and the 3,000-node row has elevated seed variation.

Computational scaling. Panel (b) reports center and $N_s = 16$ inference for the independently trained models using batch size 16 and one flow step. At each graph size, latency and convergence quantities are medians over three seeds, and memory is the maximum observed allocation. From 1,000 to 5,000 nodes, center/sample latency increases from $5.283/46.384$ to $21.274/230.506$ ms/window, and sample throughput decreases from 21.559 to 4.338 windows/s. Peak inference/training memory rises from $1.400/7.172$ to $5.701/31.092$ GiB. Runs reach within 1% of their best validation CRPS by epochs 13–14; the median time grows from 11.55 to 64.03 minutes. These measurements are implementation- and hardware-dependent; dense $N \times N$ support storage and candidate selection remain the main bottlenecks.

Table 11: Overall latitude-weighted WeatherBench2 T2M error (K). Learned models: three-seed mean±standard deviation; references are fixed.

Model	MAE	RMSE
STGraphFM	1.272 ± 0.004	2.164 ± 0.005
GraphWaveNet	1.298 ± 0.010	2.205 ± 0.015
STID	1.368 ± 0.011	2.333 ± 0.007
STAEformer	1.394 ± 0.072	2.356 ± 0.119
Climatology	1.480	2.413
Persistence	2.122	3.539

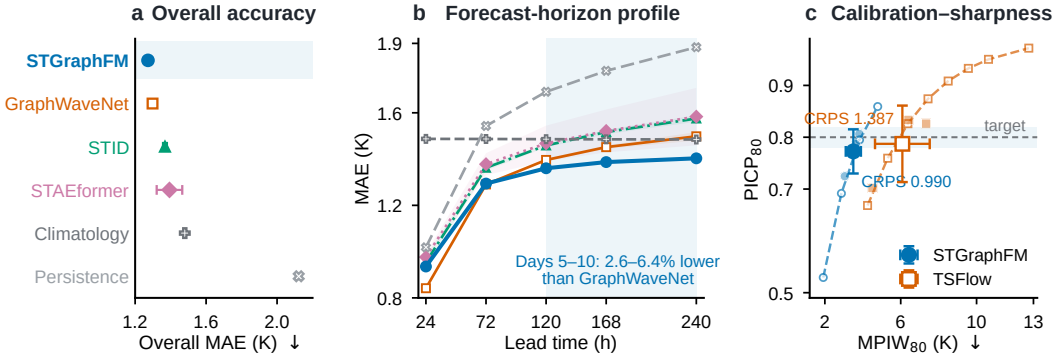

Figure 4: WeatherBench2 diagnostic views. (a) Overall MAE; (b) lead-time MAE, with shading identifying days 5–10; (c) validation spread-scale sweeps (dashed) and frozen-scale test results (marks). Error bars/bands show one standard deviation over three seeds for learned models.

C.4 Inference Controls

STGraphFM separates a deterministic central-source path from stochastic sample paths. On 1,024 LargeST-SD test windows, increasing N_s from 4 to 32 leaves the deterministic path unchanged, improves CRPS/PICP$_{80}$ from $14.485/0.578$ to $12.371/0.804$, and widens MPIW$_{80}$ from 36.330 to 53.918, while latency rises from 15.327 to 128.469 ms/window. We use $N_s = 16$ as the default calibration–sharpness–cost balance. At $N_s = 16$, one, two, and three transport steps give nearly identical CRPS ($12.6744/12.6745/12.6747$) but increase latency from 55.107 to $90.303/123.359$ ms/window; one step is therefore the default quality–cost operating point.

Control semantics. N_s changes the Monte Carlo estimation budget, K_{step} changes numerical transport cost, and τ changes the pre-transport source distribution. The first two select a quality–cost operating point; sweeping τ with a fixed checkpoint changes scenario spread without changing the independent central-source forecast.

An aligned two-dataset sweep fixes 512 windows, random draws, $N_s = 16$, and one transport step. From $\tau = 0.3$ to 1.0, PICP$_{80}$/MPIW$_{80}$/mean tail probability increase strictly from $0.324/12.858/0.175$ to $0.767/42.858/0.284$ on PEMS03 and from $0.330/15.083/0.180$ to $0.776/50.277/0.245$ on LargeST-SD. Sample-mean MAE stays within 2.1% of its $\tau = 1$ value, and CRPS is best at $\tau = 0.7$ on both datasets (11.029 and 12.346). Increasing the source radius expands predictive intervals and changes tail-event probabilities while the sample-mean forecast remains comparatively stable. The independent central-source path does not depend on τ by construction.

Symmetric probability evaluation. With the same validation-frozen scale grid and $N_s = 16$ budget, STGraphFM obtains CRPS/PICP$_{80}$/MPIW$_{80}$ of $11.118/0.802/47.154$ on PEMS03 versus $15.893/0.803/71.710$ for TSFlow. On LargeST-SD, the corresponding values are $12.455/0.777/50.277$ versus $40.349/0.794/173.173$. Across nominal levels from 50% to 95%, its maximum absolute coverage gap is $0.050/0.066$, and horizon-wise PICP$_{80}$ stays within 0.799–$0.807/0.775$–0.778 on the two datasets. As N_s rises from 8 to 64, CRPS Monte Carlo stan-

Table 12: LargeST-SD configurations (three seeds). Both panels use the same runs. Groups distinguish output calibration, alternative generators, and graph components. PICP_{80} targets 0.8.

(a) Point and marginal-distribution metrics

Variant / alternative	MAE↓	CRPS↓	PICP_{80}	MPIW_{80} ↓
STGraphFM	$18.925 \pm .272$	$14.594 \pm .173$	$.725 \pm .010$	49.929 ± 1.979
Raw samples (no affine calib.)	$18.925 \pm .272$	$14.882 \pm .168$	$.698 \pm .019$	47.253 ± 1.526
Prior only	$19.314 \pm .017$	–	–	–
Prior + calibrated Gaussian	$19.314 \pm .017$	$16.123 \pm .195$	$.613 \pm .015$	$\mathbf{42.157} \pm .881$
Prior + graph residual (no flow)	$19.314 \pm .017$	$15.626 \pm .182$	$.642 \pm .012$	$\underline{44.305} \pm .912$
Alternative one-step generator	$19.421 \pm .115$	$15.384 \pm .192$	$.675 \pm .011$	46.829 ± 1.483
Isotropic source	$19.194 \pm .666$	$15.559 \pm .676$	$.692 \pm .068$	46.368 ± 7.110
No graph attention	$19.059 \pm .318$	$15.627 \pm .314$	$.681 \pm .056$	45.124 ± 6.216

(b) Multivariate scores and region retrieval

Variant / alternative	ES↓	GVS↓	Region IoU↑
STGraphFM	$18.592 \pm .121$	$3.041 \pm .114$	$.254 \pm .008$
Raw samples (no affine calib.)	$19.327 \pm .145$	$3.481 \pm .132$	$.238 \pm .010$
Prior only	–	–	–
Prior + calibrated Gaussian	$25.854 \pm .213$	$12.429 \pm .195$	$.112 \pm .006$
Prior + graph residual (no flow)	$24.618 \pm .185$	$8.857 \pm .164$	$.158 \pm .009$
Alternative one-step generator	$23.946 \pm .161$	$7.912 \pm .147$	$.185 \pm .008$
Isotropic source	$22.153 \pm .275$	$9.228 \pm .281$	$.172 \pm .011$
No graph attention	$22.489 \pm .265$	$9.895 \pm .272$	$.165 \pm .010$

dard error falls from $0.541/0.506$ to $0.034/0.044$; the $N_s = 16$ moving-block 95% intervals are $[9.333, 12.535]/[10.975, 13.797]$.

Latency repeatability. Across three synchronized LargeST-SD repeats, max–min latency is at most 3.5% of the median over the sample-budget sweep and at most 2.6% over the flow-step sweep.

C.5 Component Analysis

The three-seed LargeST-SD analysis separates three comparisons: source/graph component ablations, raw versus calibrated outputs, and complete generator alternatives. Every sample-producing variant uses the same windows, sample count, event threshold, and region query.

Table 12 reports every configuration and metric; Table 2 in the main paper highlights the source, attention, and complete-generator comparisons.

The prior-only anchor obtains MAE 19.314 ± 0.017, whereas the full model reaches 18.925 ± 0.272 and also produces a predictive distribution. Affine calibration leaves deterministic MAE unchanged while improving raw-sample CRPS/Energy/GVS/Region IoU from $14.882/19.327/3.481/0.238$ to $14.594/18.592/3.041/0.254$. Raw samples outperform the reported non-flow and direct residual alternatives on these criteria before affine calibration. These rows compare complete model configurations; Appendix E reports the same-backbone training and state-input comparisons. Gaussian residual, graph residual, and direct one-step alternatives yield CRPS 16.123, 15.626, and 15.384, respectively, with worse joint, edge-structure, and region scores. Isotropic source and no-graph-attention variants increase GVS to $9.228/9.895$ and reduce Region IoU to $0.172/0.165$, while their MAE changes are smaller. These rows support the tested source and support-ranked attention components. The matched comparison of sample-shared correction and a state-dependent field is given in Appendix E. The full model obtains the lowest CRPS and strongest structural scores with a wider interval. GVS measures graph-edge sample structure, while Region IoU measures the retrieval region induced by nodewise risks.

C.6 Case Study: Controllable and Graph-Conditioned Traffic Scenarios

Table 13 compares STGraphFM and TSFlow on event ranking, graph-region retrieval, and structural sample scores.

Final-lead event-detection case. Figure 3 uses a 15-epoch checkpoint, one transport step, 16 samples, common random draws across $\tau \in \{0.3, 0.7, 1.0\}$, and identity output calibration. Events exceed training-derived nodewise 90th percentiles at the final lead; detection uses probability at least 0.5. The illustrated window contains 702 valid nodes and 52 events; TP is 21/26/27 and FP is one at every temperature, hence FPR is $1/650 = 0.001538$. Across all 512 windows, TPR/FPR change from $0.7204/0.0199$ to $0.7311/0.0238$; these aggregates describe the overall change in detection operating point at a fixed threshold.

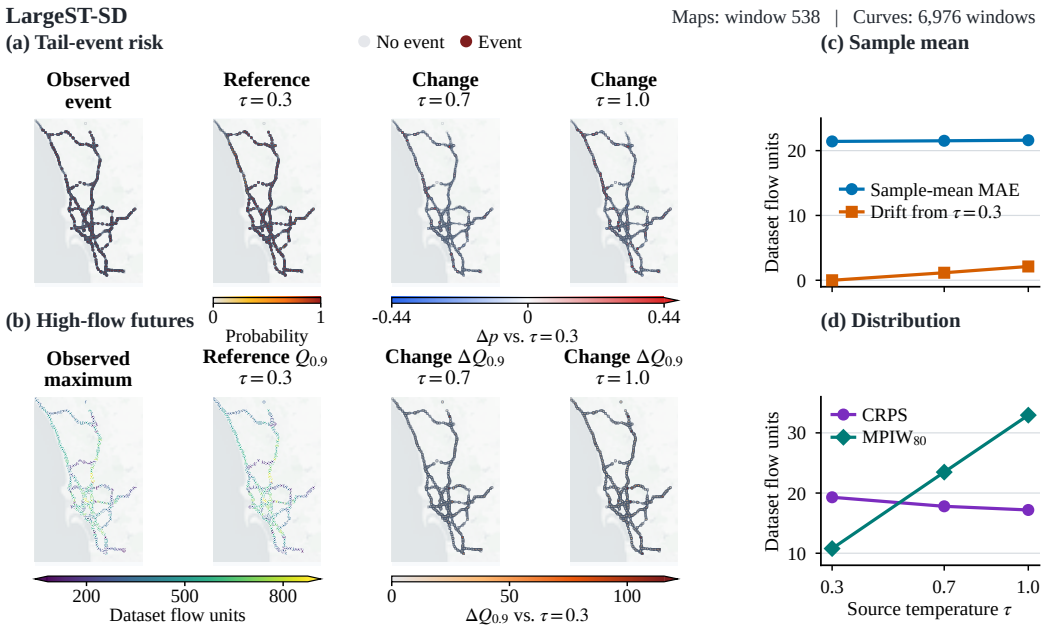

Figure 5: LargeST-SD source-temperature responses. (a,b) Observed and generated fields at window 538; changes are relative to $\tau = 0.3$ and share color scales. Colorbar arrows indicate values beyond the displayed limits. (c,d) Descriptive means over 6,976 windows; center metrics use the sample mean. $Q_{0.9}$ is the quantile across memberwise maxima over 12 forecast steps. Window selection and aggregation are detailed in Appendix C.

Table 13: Scenario queries and multivariate scores on 512 aligned windows with $N_s = 16$. Validation selects $\tau = 1.2$ for STGraphFM and a spread scale for TSFlow. Rank AP uses stable-order rank average precision; Region IoU evaluates retrieved connected regions; Lag agr. measures dominant positive-lag agreement. ES and GVS are multivariate scores; exact definitions are in Appendix C. LargeST-SD STGraphFM ES, GVS, and Region IoU repeat the reported means in Table 12.

Dataset	Model	Rank AP↑	Region IoU↑	Lag agr.↑	ES↓	GVS↓
PEMS03	STGraphFM	**0.893**	**0.165**	**0.897**	**16.280**	**2.738**
	TSFlow	0.852	0.125	0.857	22.694	3.576
LargeST-SD	STGraphFM	**0.885**	**0.254**	**0.904**	**18.592**	**3.041**
	TSFlow	0.677	0.159	0.814	56.169	12.999

STGraphFM improves Rank AP/Region IoU from $0.852/0.125$ to $0.893/0.165$ on PEMS03 and from $0.677/0.159$ to $0.885/0.254$ on LargeST-SD. It also obtains lower Energy and GVS and higher dominant-lag agreement on both datasets. Energy and GVS assess joint sample quality; AP and

Region IoU assess node-risk ranking and graph-based retrieval. Eq. 50 specifies the reported rank statistic and its tie handling.

Query overhead. On 1,024 LargeST-SD windows with $N_s = 16$, end-to-end scenario generation takes 36.255 ms/window, while probability, ranking, and region retrieval add 1.818 ms/window (4.8% of this workload). The measurement includes transfer, sampling, inverse normalization, risk aggregation, and graph search and is therefore separate from the synchronized latency evaluation above.

Figure 5 uses an output-independent event-count rule to select the displayed window. Window 538 is the lowest test index at the observed-only P90 event-count boundary (419 events). Its display region of interest (ROI) is the projected bounding box of the observed-event largest connected component plus 8% padding, containing 699 of 716 nodes; this display ROI is separate from the fixed-budget Region-IoU query. Its tail-risk component expands from 80 nodes at $\tau = 0.3$ to 103 and 102 at $\tau = 0.7$ and 1.0, while observed-region IoU changes from 0.188 to 0.246 and 0.243. This illustrates controlled spread and connected-risk expansion while the deterministic path remains fixed; the aggregate temperature results above establish the general trend.

D Information Settings and Implementation Versions

Information settings. The traffic models use target history and known calendar features. The matched comparison in Appendix E gives the compared models the same available history and calendar inputs.

Static context. The optional C_s is an eight-dimensional static spectral encoding. Let $\widetilde{A} = A + I$, $D_A = \mathrm{diag}(\widetilde{A}\mathbf{1})$, and $B_A = D_A^{-1/2}\widetilde{A}D_A^{-1/2}$. LargeST explicitly symmetrizes the normalized operator as $B_{\mathrm{sym}} = (B_A + B_A^\top)/2$ before eigendecomposition; KnowAir uses a symmetric graph, so B_A is already symmetric. The PEMS preprocessing instead passes its asymmetric B_A directly to the symmetric eigensolver with the default lower-triangle convention. Its effective matrix is therefore $B_{\mathrm{low}} = \mathrm{tril}(B_A) + \mathrm{tril}(B_A, -1)^\top$, not the diffusion operator of an explicitly symmetrized adjacency. For the corresponding symmetric matrix, retain the second through ninth eigenpairs in descending order and set $C_{s,ik} = (D_A)_{ii}^{-1/2} U_{ik} \max(\lambda_k, 0)^2$. The velocity network projects these coordinates into its token space; the archived sensor prior configuration disables this static-context input.

Implementation versions. The sensor implementation described in Appendix B attenuates training graph bias from 1 to 0.1 and recomputes candidates at each velocity call. The frozen matched runtime instead drops this bias from 1 to 0 and caches graph candidates. Its KnowAir graph is symmetric, so the two support channels coincide. It uses arithmetic-equivalent dense source propagation for deterministic CUDA execution. Explicit evaluation mode disables radius/mixture augmentation, condition dropout, and network dropout while preserving random source sampling. Its checkpoint checks verify fixed-seed repeatability, independence from future targets, and restoration of network mode after evaluation.

Materials and interpretation. Table 6 maps each analysis to its sample and training budget. The anonymous repository provides code, configurations, and STGraphFM checkpoints. Source temperature controls initial random variation, while the central-source forecast supplies the point output. Joint scores, fixed-marginal comparisons, and transformed-trajectory scores evaluate sample dependence, while risk queries evaluate their use for retrieval.

E Aligned Objectives and Sample-State Conditioning

The controls address two possible explanations for better forecasts: the training objective may matter beyond the backbone, and each sample may benefit from its own state-dependent update beyond a shared correction. Common-marginal and trajectory comparisons then test whether the gains extend beyond coordinate-wise predictive quality.

Data, optimization, and output protocol. All models use the 228-node KnowAirV2 series, 12 hourly history steps, a 12-hour horizon, calendar features, node identities, and a training-derived graph. The chronological split is 70%/10%/20%, with 49,066 training windows. The first validation half supplies 64 evenly spaced checkpoint-selection windows; after a 24-hour embargo, the second half supplies 3,471 calibration windows. Testing uses all 14,003 admissible windows. Statistics, graph construction, and event thresholds use training data only; targets are the finite, publicly released imputed observations. Seeds 20261001 and 20261002 each train for five epochs with batch size eight, AdamW at learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, and gradient-norm clip 1. Eight-member validation after epochs two and four selects the checkpoint. Calibration and testing use 16 members, one-step decoding, and source temperature one. A positive scale about each coordinate's ensemble median is fitted for 80% coverage on the calibration partition and frozen for testing. The three matched controls share this five-epoch update schedule. Runtime variants are specified in Appendix D.

Training objective and sample-state controls. Score16 uses the same 298,883 parameters and inference output map as the velocity model. It trains 16 generated members with CRPS plus 0.5 times root-mean-square normalized ES, retaining the distinct-pair denominator $S(S-1)$. The transported central-source trajectory is the center-alignment anchor and receives score gradients; the temporal prior retains its smooth-L_1 and Gaussian auxiliary objectives. Common ensemble-center alignment couples members, so the distinct-pair correction does not yield an unbiased score estimate. The sample-shared control retains the same bounded velocity objective, graph-correlated source, graph, calendar and parameters, replacing the current state input by the detached predictive anchor μ_p. Both controls use the same one-step solver and center-alignment rule at test time.

Table 14: Per-seed calibrated quality. Seeds 001/002 denote 20261001/20261002; Shared is sample-shared correction. CSDI budgets are listed separately.

Model	Seed	Epochs	CRPS	ES	GVS	PICP_{80}
STGraphFM	001	5	10.2100	13.8205	3.1716	0.7958
STGraphFM	002	5	10.2862	13.9328	3.9252	0.8089
Score16	001	5	10.5336	14.3265	4.4411	0.8071
Score16	002	5	10.5907	14.3293	4.6290	0.8176
Shared	001	5	10.6565	14.4589	4.5016	0.8028
Shared	002	5	10.5499	14.3191	4.5362	0.8161
CSDI	001	20	9.7975	13.4084	2.2469	0.7916
CSDI	002	10	10.4047	14.1299	2.2578	0.7894
SpecSTG	001	5	15.0766	18.9947	3.4244	0.7999
SpecSTG	002	5	14.0955	18.0028	3.2330	0.8104

Regional co-exceedance events. For a graph region $\mathcal{R}$, the event at each lead is that at least 25% of its valid nodes exceed their training q90:

$$e^{\mathcal{R}}_{bh} = \mathbf{1}\left\{ \frac{\sum_{i\in\mathcal{R}} \mathcal{M}_{bih}\mathbf{1}\{Y_{bih} > q_i\}}{\max(1, \sum_{i\in\mathcal{R}} \mathcal{M}_{bih})} \geq 0.25 \right\}, \quad p^{\mathcal{R}}_{bh} = \frac{1}{S}\sum_m e^{\mathcal{R},(m)}_{bh}. \tag{55}$$

Here $e^{\mathcal{R},(m)}$ replaces Y by the m-th sample with the same mask. The score averages $(p^{\mathcal{R}}_{bh} - e^{\mathcal{R}}_{bh})^2$ over leads with at least one valid region node and then over windows. The mechanism comparisons average this score equally over eleven training-defined regions. Their construction is specified in Appendix F.1.

E.1 Paired Statistics and Common Marginals

All differences below are STGraphFM minus control. For each fixed trained pair, 2,000 paired circular moving-block replicates give 95% intervals at both 24-hour and 168-hour block lengths. They describe temporal-window variation, rather than uncertainty over a training population. The main-text two-seed averages are descriptive.

Common-marginal comparisons hold one model's 16 raw empirical values fixed at every coordinate and reassemble them with the other model's member ranks, without test labels. Both models supply

the common marginals in turn; coordinate CRPS, coverage and width are identical. Macro Brier is the unweighted mean of Brier scores over eleven training-defined regions, with the regional event defined in Eq. 55. The eleven-region partition is specified in Appendix F.1; the supplementary Score4 study uses a single region. Score16 and shared correction have nearly identical scores under STGraphFM marginals; these are not independent replications of a dependence effect.

Table 15: Paired differences for calibrated outputs; lower scores are better.

Control	Seed	Metric	Δ	24h 95% CI	168h 95% CI
Score16	001	CRPS	-0.32360	$[-0.36916, -0.27986]$	$[-0.36586, -0.27827]$
Score16	001	ES	-0.50600	$[-0.56272, -0.45048]$	$[-0.54966, -0.46163]$
Score16	001	GVS	-1.26956	$[-1.29380, -1.24490]$	$[-1.29495, -1.24465]$
Score16	002	CRPS	-0.30453	$[-0.32435, -0.28467]$	$[-0.32619, -0.28390]$
Score16	002	ES	-0.39646	$[-0.42540, -0.36821]$	$[-0.42942, -0.36579]$
Score16	002	GVS	-0.70384	$[-0.72295, -0.68517]$	$[-0.72030, -0.68761]$
Shared	001	CRPS	-0.44653	$[-0.50154, -0.39261]$	$[-0.50981, -0.38160]$
Shared	001	ES	-0.63833	$[-0.70644, -0.57130]$	$[-0.69939, -0.57401]$
Shared	001	GVS	-1.33006	$[-1.35472, -1.30540]$	$[-1.35652, -1.30473]$
Shared	002	CRPS	-0.26367	$[-0.29378, -0.23549]$	$[-0.29109, -0.23631]$
Shared	002	ES	-0.38628	$[-0.42685, -0.34743]$	$[-0.42484, -0.34886]$
Shared	002	GVS	-0.61101	$[-0.63162, -0.59105]$	$[-0.62862, -0.59294]$

Table 16: Both common-marginal directions: STGraphFM minus Score16.

Seed	Marginals	Metric	Δ	24h 95% CI	168h 95% CI
001	STGraphFM	ES	-0.01983	$[-0.02046, -0.01921]$	$[-0.02046, -0.01918]$
001	STGraphFM	GVS	-0.35762	$[-0.36931, -0.34637]$	$[-0.37530, -0.34115]$
001	STGraphFM	Macro Brier	-0.00088	$[-0.00107, -0.00071]$	$[-0.00118, -0.00059]$
001	Score16	ES	-0.02261	$[-0.02331, -0.02189]$	$[-0.02333, -0.02189]$
001	Score16	GVS	-0.48755	$[-0.49759, -0.47709]$	$[-0.50285, -0.47309]$
001	Score16	Macro Brier	-0.00083	$[-0.00102, -0.00065]$	$[-0.00112, -0.00056]$
002	STGraphFM	ES	-0.01286	$[-0.01329, -0.01243]$	$[-0.01351, -0.01225]$
002	STGraphFM	GVS	-0.19363	$[-0.19813, -0.18906]$	$[-0.20118, -0.18648]$
002	STGraphFM	Macro Brier	-0.00075	$[-0.00091, -0.00060]$	$[-0.00100, -0.00050]$
002	Score16	ES	-0.01221	$[-0.01264, -0.01177]$	$[-0.01289, -0.01156]$
002	Score16	GVS	-0.09374	$[-0.09772, -0.08988]$	$[-0.09938, -0.08844]$
002	Score16	Macro Brier	-0.00053	$[-0.00065, -0.00042]$	$[-0.00071, -0.00036]$

Table 17: Both common-marginal directions: STGraphFM minus Shared.

Seed	Marginals	Metric	Δ	24h 95% CI	168h 95% CI
001	STGraphFM	ES	-0.01983	$[-0.02046, -0.01921]$	$[-0.02046, -0.01918]$
001	STGraphFM	GVS	-0.35762	$[-0.36931, -0.34637]$	$[-0.37530, -0.34115]$
001	STGraphFM	Macro Brier	-0.00088	$[-0.00107, -0.00071]$	$[-0.00118, -0.00059]$
001	Shared	ES	-0.02346	$[-0.02416, -0.02274]$	$[-0.02414, -0.02277]$
001	Shared	GVS	-0.48833	$[-0.49823, -0.47799]$	$[-0.50348, -0.47378]$
001	Shared	Macro Brier	-0.00087	$[-0.00108, -0.00068]$	$[-0.00118, -0.00058]$
002	STGraphFM	ES	-0.01286	$[-0.01329, -0.01243]$	$[-0.01351, -0.01225]$
002	STGraphFM	GVS	-0.19363	$[-0.19813, -0.18906]$	$[-0.20118, -0.18648]$
002	STGraphFM	Macro Brier	-0.00075	$[-0.00091, -0.00060]$	$[-0.00100, -0.00050]$
002	Shared	ES	-0.01267	$[-0.01309, -0.01224]$	$[-0.01329, -0.01208]$
002	Shared	GVS	-0.09345	$[-0.09743, -0.08960]$	$[-0.09899, -0.08821]$
002	Shared	Macro Brier	-0.00054	$[-0.00066, -0.00042]$	$[-0.00071, -0.00036]$

E.2 Distribution Scores for Trajectory Functionals

We evaluate the predictive distributions of temporal increments, graph differences and four regional trajectory functionals, computed from each complete sample and its observed counterpart. All outputs use validation-fitted scaling. Both seeds improve over both same-backbone controls on all six scores at both block lengths.

Table 18: Per-seed calibrated CRPS of six trajectory functionals.

Functional	Seed	STGraphFM	Score16	Shared
Temporal increment	001	5.3553	7.4185	7.4949
Temporal increment	002	6.8324	7.6322	7.5224
Graph difference	001	8.2837	9.4601	9.4991
Graph difference	002	8.8944	9.6431	9.5824
Regional mean	001	8.1887	8.3925	8.5916
Regional mean	002	8.1760	8.2531	8.2457
Regional peak	001	17.6418	28.2311	29.3593
Regional peak	002	18.3036	27.6683	27.9422
Exceedance fraction	001	0.0207	0.0218	0.0219
Exceedance fraction	002	0.0205	0.0217	0.0217
Exceedance duration	001	0.2577	0.2983	0.2989
Exceedance duration	002	0.2606	0.2956	0.2951

Table 19: Paired trajectory-CRPS differences: STGraphFM minus Score16.

Functional	Seed	Δ	24h 95% CI	168h 95% CI
Temporal increment	001	-2.06318	$[-2.09466, -2.03197]$	$[-2.11600, -2.01258]$
Temporal increment	002	-0.79985	$[-0.82619, -0.77401]$	$[-0.83952, -0.76332]$
Graph difference	001	-1.17633	$[-1.19865, -1.15455]$	$[-1.20169, -1.15113]$
Graph difference	002	-0.74869	$[-0.76897, -0.72861]$	$[-0.77711, -0.72207]$
Regional mean	001	-0.20377	$[-0.26564, -0.14353]$	$[-0.25638, -0.15330]$
Regional mean	002	-0.07714	$[-0.11753, -0.03899]$	$[-0.12554, -0.02990]$
Regional peak	001	-10.58929	$[-10.97416, -10.20464]$	$[-11.35142, -9.80122]$
Regional peak	002	-9.36469	$[-9.66350, -9.06507]$	$[-9.93975, -8.77850]$
Exceedance fraction	001	-0.00104	$[-0.00136, -0.00075]$	$[-0.00137, -0.00074]$
Exceedance fraction	002	-0.00117	$[-0.00136, -0.00100]$	$[-0.00149, -0.00086]$
Exceedance duration	001	-0.04058	$[-0.04756, -0.03418]$	$[-0.05308, -0.02866]$
Exceedance duration	002	-0.03500	$[-0.04212, -0.02853]$	$[-0.04638, -0.02404]$

Table 20: Paired trajectory-CRPS differences: STGraphFM minus Shared.

Functional	Seed	Δ	24h 95% CI	168h 95% CI
Temporal increment	001	-2.13961	$[-2.17083, -2.10849]$	$[-2.19183, -2.08916]$
Temporal increment	002	-0.68999	$[-0.71684, -0.66370]$	$[-0.72851, -0.65463]$
Graph difference	001	-1.21536	$[-1.23716, -1.19354]$	$[-1.24091, -1.18916]$
Graph difference	002	-0.68798	$[-0.70994, -0.66690]$	$[-0.71856, -0.66034]$
Regional mean	001	-0.40285	$[-0.48426, -0.32517]$	$[-0.48397, -0.32217]$
Regional mean	002	-0.06976	$[-0.12326, -0.01671]$	$[-0.11942, -0.02114]$
Regional peak	001	-11.71749	$[-12.12804, -11.30583]$	$[-12.54015, -10.87914]$
Regional peak	002	-9.63855	$[-9.95349, -9.32400]$	$[-10.25243, -9.01277]$
Exceedance fraction	001	-0.00113	$[-0.00148, -0.00080]$	$[-0.00148, -0.00080]$
Exceedance fraction	002	-0.00115	$[-0.00132, -0.00098]$	$[-0.00146, -0.00085]$
Exceedance duration	001	-0.04115	$[-0.04781, -0.03506]$	$[-0.05342, -0.02947]$
Exceedance duration	002	-0.03451	$[-0.04148, -0.02822]$	$[-0.04579, -0.02377]$

E.3 Other Generators

Generic adapters. The table below reports single-seed, five-epoch FM/DDPM adapters with a shared 114,049-parameter graph-free temporal backbone. They are distinct from the 298,883-parameter same-backbone controls. Their costs and implementation are described in Appendix C.1.

Table 21: Supplementary generic-generator comparison, seed 20261001.

Model	CRPS	ES	GVS	PICP_{80}
STGraphFM	10.2100	13.8205	3.1716	0.7958
FM	10.3626	14.6774	5.7797	0.7975
DDPM	10.3086	14.4330	6.0448	0.7918

F Output Diagnostics and Source Sensitivity

These diagnostics clarify how numerical bounds, retrieval conventions, regional calibration, and source settings affect the interpretation of generated samples. They complement the controlled mechanism comparisons in Appendix E.

F.1 Clipping, Retrieval, and Regional Calibration Diagnostics

Clip activation. For two trained STGraphFM checkpoints, fixed probes of 32 training windows and 64 validation windows measure the fraction of normalized coordinates outside $[-8, 8]$ before clipping. Zero denotes no exceedance on this probe, not a guarantee over full training or testing. These activation measurements are distinct from a retrained ablation of the clipping thresholds.

Table 22: Pre-clip exceedance rates on fixed probes (%).

Quantity	Seed 001	Seed 002
Train target	0.06739	0.06739
Train source	0.00343	0.00571
Train target velocity	0.00000	0.00114
Train predicted velocity	0.00000	0.00000
Validation velocity	0.00000	0.00000
Validation updated state	0.00000	0.00000

Ties and physical output support. Over all 14,003 test windows, we compare stable-order AP, expected AP under uniform within-tie permutations, threshold-grouped AP, and best/worst within-tie bounds. Outputs are not truncated to nonnegative values; negative fractions also cover the full test set. A separate 512-window relabeling probe uses the paper's candidate rule $K = \max\{50, \text{round}(0.15N)\}$ and `argpartition`. Raw-sample mean candidate Jaccard is 0.374/0.526 for the two seeds. Lag agreement is 0.9922/0.9927; the always-lag-one reference is 0.9927.

Table 23: Tie-sensitive AP and negative outputs; the last column is a percentage of sample coordinates.

Seed	Output	Stable	Expected	Grouped	Bounds	Neg.%
001	Raw	0.7746	0.7804	0.7570	$[0.7360, 0.8511]$	0.85
001	Scaled	0.7988	0.8011	0.7837	$[0.7642, 0.8365]$	3.17
002	Raw	0.7851	0.7900	0.7707	$[0.7339, 0.8553]$	3.65
002	Scaled	0.7988	0.8015	0.7865	$[0.7488, 0.8433]$	6.53

Regional calibration. The training graph is partitioned into eleven regions by deterministic breadth-first traversal: start from the lowest unassigned node ID, visit neighbors in ascending order, and cap each group at 32 nodes. The table reports coverage of nominal 80% intervals for each regional mean. STGraphFM improves coverage over Score16 and sample-shared correction in all eleven regions for both seeds. Macro Brier for regional tail events is included in both common-marginal directions.

Table 24: Per-region PICP_{80}; column suffixes denote training seeds.

Region	STG-001	S16-001	Shared-001	STG-002	S16-002	Shared-002
0	0.6448	0.4887	0.4568	0.5360	0.4750	0.5007
1	0.6384	0.4645	0.4427	0.5575	0.4834	0.4943
2	0.6737	0.5430	0.5109	0.6006	0.5621	0.5801
3	0.7698	0.6070	0.5836	0.6971	0.6429	0.6413
4	0.6281	0.4925	0.4693	0.5447	0.5143	0.5207
5	0.7759	0.6205	0.6034	0.6560	0.6374	0.6395
6	0.8194	0.7341	0.7214	0.7693	0.7479	0.7424
7	0.6759	0.5142	0.4821	0.6047	0.5663	0.5610
8	0.6603	0.5408	0.5001	0.5874	0.5534	0.5651
9	0.8028	0.6933	0.6855	0.7561	0.7198	0.7183
10	0.7983	0.6630	0.6495	0.7397	0.6878	0.6878

F.2 Source Sensitivity and Supplementary Distribution Diagnostics

Temperature and graph-noise mixing. Source temperature changes the initial perturbation radius. In a separate three-seed study, affine output scaling is fitted on validation data to match the coverage of $\tau = 1.25$. Table 25 reports temperature-minus-affine differences in CRPS, ES, and GVS. A second study fixes $\rho = 0.5$ on validation data and compares common-marginal scores against the reference, independent ranks, and training-history ranks. Together, these comparisons quantify sensitivity to source initialization.

Table 25: Temperature minus affine control matched on validation coverage.

Seed	ΔCRPS	ΔES	ΔGVS
20260930	0.11812	0.13273	0.08859
20261001	0.10168	0.07890	0.10092
20261002	0.08488	0.06112	0.08482

Table 26: At fixed marginals, the $\rho = 0.5$ candidate minus rank controls.

Seed	Control	ΔES	ΔGVS	ΔRegion Brier
20260930	Reference	-0.00339	-0.10416	-0.00042
20260930	Independent	-0.02128	-0.56376	-0.00261
20260930	Train-history	-0.04726	0.05785	0.00089
20261001	Reference	-0.00277	-0.15535	-0.00033
20261001	Independent	-0.01832	-0.71635	-0.00247
20261001	Train-history	-0.04552	0.23271	0.00085
20261002	Reference	-0.00362	-0.13348	-0.00021
20261002	Independent	-0.01829	-0.55471	-0.00171
20261002	Train-history	-0.05827	0.17702	0.00072

Conditional C2ST and MMD. A fixed summary-feature logistic probe performs a conditional classifier two-sample test (C2ST); biased squared maximum mean discrepancy (MMD^2) is computed on the joint features. Kernel bandwidth is selected only on validation data. There are four generated samples per history, 145 validation histories, and 584 test histories. Real–real MMD between alternating days is a reference rather than an exact null distribution. The checkpoints in this diagnostic were trained separately from those in the two-seed, five-epoch mechanism study.

Table 27: Conditional distribution diagnostics for separately trained checkpoints.

Model	Seed	C2ST acc.	MMD^2	Real–real MMD^2
STGraphFM	20260930	0.7207	0.01249	0.00134
STGraphFM	20261001	0.9155	0.01690	0.00139
STGraphFM	20261002	0.7735	0.01507	0.00139
CSDI	20260930	0.6083	0.00501	0.00084
CSDI	20261001	0.5749	0.00200	0.00091
CSDI	20261002	0.6535	0.00429	0.00079
SpecSTG	20261001	0.8566	0.11302	0.00079
SpecSTG	20261002	0.8673	0.13487	0.00079

G Supplementary Four-Member Endpoint-Score Comparison

This comparison examines endpoint-score training with four training members, 16 test members, and different center anchors during training and testing. It complements the aligned Score16 study in Appendix E with a single-seed, longer-training setting.

G.1 Protocol, Objectives, and Output Maps

Data and information. Both models use the 228-node public KnowAirV2 PM2.5 series, 12 hourly history steps, a 12-hour forecast, calendar features, node identities, and the same training-derived graph. The chronological split is 70%/10%/20%, with 49,066 training windows. The first half of validation supplies 64 evenly spaced checkpoint-selection windows; after a 24-hour embargo, the second half supplies all 3,471 calibration windows. Testing uses all 14,003 admissible windows with 16 samples each. Statistics, graph construction, and nodewise event thresholds use training data only. Targets are the finite, publicly released imputed observations.

Backbone and objectives. Both models use the same 298,883-parameter prior and graph-time network: width 64, two blocks, four heads, and eight graph candidates. *Velocity regression* uses the bounded target in Eq. 33. *Score training* evaluates four generated samples and optimizes $\mathrm{CRPS}_{\mathrm{fair}} + 0.5\mathrm{ES}_{\mathrm{fair}}$ in normalized coordinates. For either score, the fair form replaces the empirical pairwise denominator S^2 with $S(S-1)$ and sums distinct pairs. Both retain the prior smooth-L_1 loss and Gaussian likelihood weight 0.05.

Generated outputs and center alignment. Let $R^{(m)} = c(x_0^{(m)} + v_\theta(x_0^{(m)}, 0.5 \mid C))$ denote a one-step sample before center alignment. The frozen implementation uses coefficient 0.25 for center alignment. The samples supplied to the Score-training loss and to test evaluation are, respectively,

$$Z_{\mathrm{train}}^{(m)} = R^{(m)} + 0.25(\mu_p - \overline{R}_4), \qquad Z_{\mathrm{test}}^{(m)} = R^{(m)} + 0.25(\hat{Y}_{\mathrm{cs}} - \overline{R}_{16}). \tag{56}$$

Both methods use the test map. *Base output* includes this center alignment; *Scaled* adds median-centered calibration. Score training uses different center anchors at training and test time. Shared ensemble centering makes its fair-denominator loss operate on dependent members.

Optimization and calibration. Training uses seed 20260930, batch size 8, AdamW with learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, and gradient-norm clip 1. Every two epochs, eight-sample base-output validation CRPS selects the checkpoint. Velocity regression stops by early stopping at epoch 22 and selects epoch 6; Score training stops at its budget of 21 epochs and selects epoch 12. Calibration applies one positive scale around each coordinate's ensemble median, chosen for 80% coverage on the calibration partition and then frozen: 1.452992 for Velocity regression and 1.211951 for Score training. This is the median-centered rule for this comparison; Eq. 43 gives the sensor-benchmark affine rule.

Execution. Runs use NVIDIA GB10, PyTorch 2.13.0 with CUDA 13.0, 32-bit floating-point (FP32) arithmetic, deterministic algorithms, and four CPU compute threads; automatic mixed precision (AMP), TensorFloat-32 (TF32), and compilation are disabled. Sampling uses batch size 2, one velocity evaluation per sample, and temperature 1. A batch beginning at window ID w uses sampling

Table 28: Same-backbone KnowAirV2 objective comparison. One training seed, 14,003 test windows and 16 samples per window; 22 completed epochs for the velocity model and 21 for Score. Bold/underline rank the two methods within each base/median-scaled setting; PICP is ranked by distance to 0.8; lower MPIW is better.

Setting	Model	CRPS↓	PICP_{80}	MPIW_{80} ↓	ES↓	GVS↓
Base	Velocity	**10.102**	0.652	**28.375**	**13.837**	**2.758**
Base	Score	10.276	**0.740**	33.994	14.079	3.660
Scaled	Velocity	**10.110**	**0.795**	41.228	**13.816**	**3.604**
Scaled	Score	10.342	0.808	**41.199**	14.120	4.357

seed $20260930 + 700000 + w$. Window IDs, resolved configurations, scales, checkpoint hashes, and generated tensors are retained. Appendix D collects runtime differences and evaluation-mode checks.

G.2 Joint Scores and Trajectory Distributions

Table 28 gives all base and median-scaled scores, including the joint scores omitted from the compact Table 32(a). Base outputs already include the center alignment in Eq. 56; Scaled adds the validation-fixed median-centered scale. All analyses below use the same 14,003 test windows.

Exact common-marginal comparison. At each window–node–lead coordinate, we sort Velocity's 16 base-output values and reassign them using Score's member ranks, without target access. This fixes empirical marginals, CRPS, coverage, and width while changing cross-coordinate ordering. Table 29 gives absolute scores; Table 30 gives 2,000-replicate paired circular moving-block intervals with 168-hour blocks, conditional on this training seed.

Table 29: Absolute joint scores with the same 16 empirical marginal values from Velocity regression. Only the cross-coordinate ordering is changed; lower is better.

Ordering	Energy	GVS	Regional Brier
Velocity regression	13.836983	2.757814	0.029517
Score training	13.853547	2.998568	0.030654

Table 30: Score differences (Velocity minus Score) and 95% paired circular moving-block intervals: 2,000 replicates and 168-hour blocks, conditional on this training seed and its predictions. Common marginals exactly match all 16 empirical values at every coordinate.

Setting	Score	Difference	95% interval
Scaled	CRPS	-0.23228	$[-0.29778, -0.17420]$
Scaled	ES	-0.30411	$[-0.40655, -0.21233]$
Scaled	GVS	-0.75283	$[-0.76494, -0.74055]$
Common marginals	ES	-0.01656	$[-0.01723, -0.01594]$
Common marginals	GVS	-0.24075	$[-0.25573, -0.22612]$
Common marginals	Regional event Brier	-0.00114	$[-0.00166, -0.00068]$

Single-region evaluation. The supplementary Score4 comparison applies Eq. 55 to one fixed region: breadth-first search from node 0 on the training-derived adjacency, stopping at 32 nodes. It evaluates this region rather than the eleven-region macro average used in Appendix E.

Trajectory and regional distributions. Table 31 scores fixed transformations of samples and observations over all 14,003 windows. Temporal increments start at the last history value. Graph differences use 1,213 evenly spaced edges of the train-derived undirected support. Fixed breadth-first graph groups define regional mean per lead, maximum over nodes and leads, and fraction above training nodewise q90. Duration is the longest consecutive exceedance run per node within 12 hours. These scores assess how marginal rescaling affects trajectory and regional distributions.

Table 31: Transformed CRPS of generated joint samples; lower is better. Bold and underline compare the two models within each base/median-scaled setting. Increment, graph-difference, regional-mean and regional-peak scores use PM2.5 concentration units; exceedance fraction is dimensionless and duration is in hours.

Statistic	Base		Scaled	
	Velocity	Score	Velocity	Score
Temporal increment	**5.4235**	6.5116	**6.5793**	7.3204
Graph difference	**8.1037**	9.0014	**8.6474**	9.4080
Regional mean	**8.1758**	8.2549	**7.9567**	8.0999
Regional peak	**15.5974**	21.5502	**20.8672**	26.8269
Regional exceedance fraction	**0.0207**	0.0216	**0.0210**	0.0218
Longest exceedance duration	**0.2604**	0.2918	**0.2618**	0.2974

Table 32: Same-backbone KnowAirV2 objective comparison: one training seed, 14,003 test windows, and 16 samples. (a) Base includes center alignment; Scaled adds validation-selected median scaling. (b) Velocity minus Score at common Velocity-Base empirical marginals. The 95% intervals use 2,000 paired circular moving-block replicates with 168-hour blocks.

(a) Marginal scores and calibration

	Base		Scaled	
Metric	Velocity	Score	Velocity	Score
CRPS↓	**10.102**	10.276	**10.110**	10.342
PICP_{80}	0.652	0.740	0.795	0.808
MPIW_{80} ↓	**28.375**	33.994	41.228	**41.199**

(b) Joint scores at common marginals

	Velocity − Score	
Metric	Difference	95% interval
ES	−0.01656	[−0.01723, −0.01594]
GVS	−0.24075	[−0.25573, −0.22612]
Regional Brier	−0.00114	[−0.00166, −0.00068]

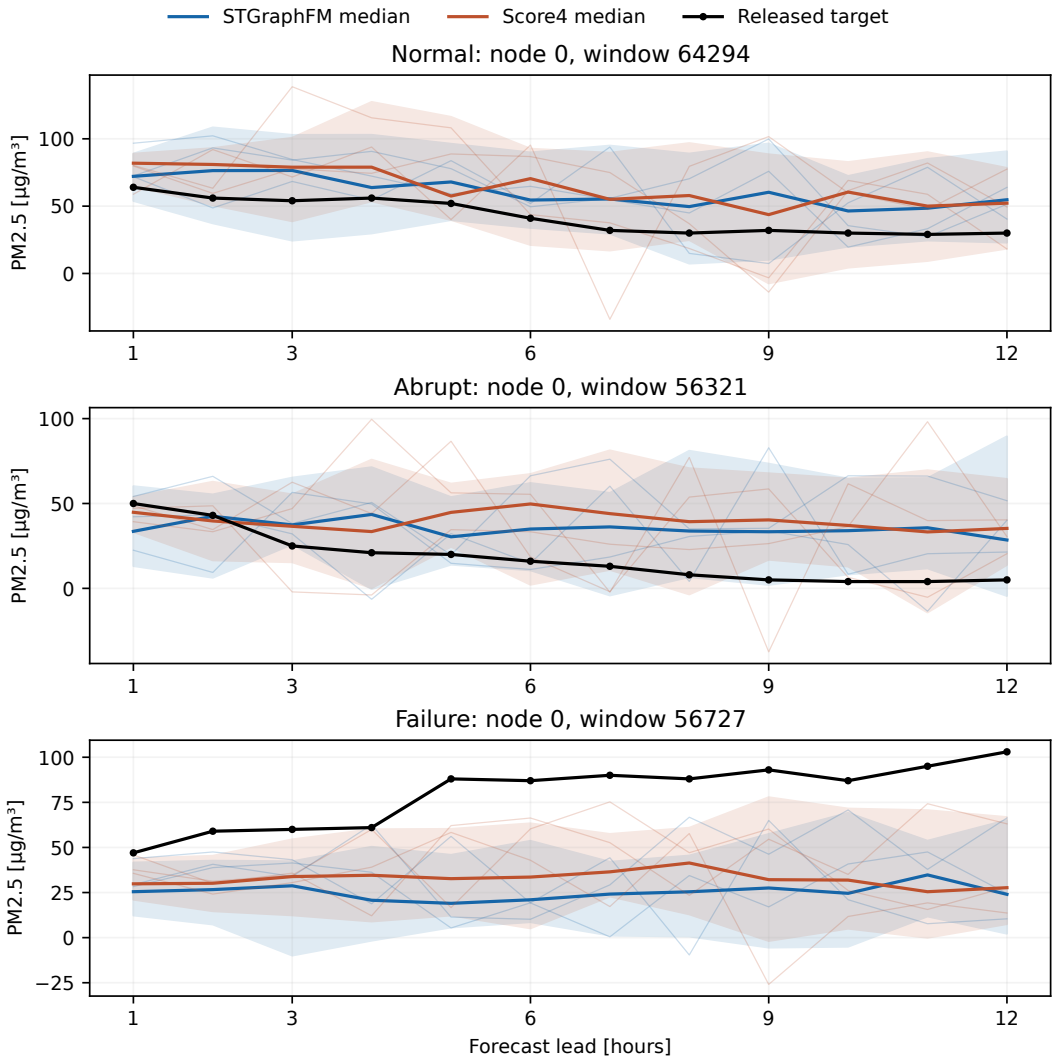

Figure 6: Observed and generated KnowAirV2 trajectories from the full-test comparison of STGraphFM and Score4. Both models use the same node, window, physical units, and forecast leads. Solid colored lines are ensemble medians, shaded bands are calibrated central 80% intervals, and thin lines are the first three saved generated trajectories. Black lines show the released target series. Normal and abrupt cases use only the last observed hourly change at fixed node 0, with the abrupt threshold fixed from training changes. The failure case is explicitly selected after evaluation by the largest fraction of interval misses for Velocity regression at node 0; it is a selected failure case rather than a representative window.