Mamba Flow Matching Neural Processes: Linear-Time Inference for Irregularly Observed Spatial Fields

Cosmo Santoni* **Giovanni Charles** **Timothy James Hitge**
Oliver John Watson **Elizaveta Semenova**
Imperial College London

Abstract

Interpolating a spatial field, such as daily maximum temperature, from sparse and irregularly located observations is a recurring task across the environmental and health sciences. The standard tool, Gaussian-process kriging, refits hyperparameters and solves an $O(N^3)$ system for every new field. When many related fields are observed, such as daily maps from one sensor network, neural processes (NPs) instead learn the interpolation map from past fields and predict a new field in a single forward pass. Joint-predictive variants, which sample whole fields rather than independent values at each location, typically rely on self-attention over context and query points, incurring $O(K^2)$ compute in the combined point count K and capping practical use at a few thousand points. State-space models offer a linear-time alternative, but a direct state-space adaptation compresses the context into a single hidden state before reading queries, reducing the information available to the decoder. We propose MambaFlowNP, a joint-predictive NP with linear compute, trained with a conditional flow-matching objective. A multi-directional Mamba-2 scan reads context and query tokens in a single shared sequence, so each query's velocity is read from its own scan position rather than from a compressed context summary, and because queries only read the scan state, predictions are consistent across query sets. MambaFlowNP is competitive with the strongest attention-based joint-predictive baselines on synthetic Matérn Gaussian-process benchmarks, and across regional and continental-scale NOAA temperature interpolation tasks it attains the lowest RMSE, the lowest CRPS and near-nominal coverage on every tier, while running 114–784$\times$ faster than the strongest joint-predictive baseline. A single forward pass scales to 10^6 points on one NVIDIA L40S GPU in under a second. This makes joint prediction computationally feasible for continental-scale, irregularly sampled sensor networks, with predictive quality evaluated up to K=5,580.

1 Introduction

Modelling continuous spatial fields from discrete, scattered sensor data remains a fundamental challenge in spatial statistics and probabilistic machine learning. Examples include inferring temperature from weather stations (Vaughan et al., 2022), disease prevalence from sentinel sites (Weiss et al., 2025), and rainfall between rain gauges (Li et al., 2023). Gaussian-process kriging is the standard approach, but it refits hyperparameters and solves an $O(N^3)$ system for every new field. When many related fields are observed, as with daily readings from a fixed sensor network, neural processes (NPs; Garnelo et al., 2018b) cast this as amortised inference, in which the network is trained on many

*Corresponding author: cosmo.santoni@imperial.ac.uk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

datasets drawn from a data-generating process and, given a new dataset, performs inference in a single forward pass without further optimisation. An episode is one dataset together with a random split into context observations and query locations at which values are to be predicted. Factorised NPs infer the parameters of an independent Gaussian distribution at each query (Garnelo et al., 2018a), whilst latent-variable NPs introduce a global latent variable (Garnelo et al., 2018b) through which all cross-query dependence passes. In contrast, a fully joint-predictive NP directly models the conditional distribution of the values at all query locations together, via autoregressive (Nguyen and Grover, 2022), diffusion (Dutordoir et al., 2023) or flow (Abu Hamad and Rosenbaum, 2025) parametrisations. Most existing fully joint-predictive instances employ self-attention over the concatenation of N context and M query locations, with compute scaling as $O(K^2)$, where $K = N + M$. Flow Matching NP (FNP) (Abu Hamad and Rosenbaum, 2025) and Transformer NP (TNPA) (Nguyen and Grover, 2022) are representative, and FNP exhausts GPU memory at 16,384 points and TNPA at 65,536 (Table 8). As an alternative to self-attention transformers, state-space models (SSMs) such as Mamba-2 (Dao and Gu, 2024) run a recurrent scan in $O(K)$ time, FLOPs, and activation memory. However, a naive adaptation of these models compresses the context into a single hidden state, a bottleneck that we show degrades accuracy on synthetic and NOAA tasks (§6.4).

MambaFlowNP (MFNP) avoids this compression by maintaining a hidden state at every token position throughout a multi-directional state-space scan (Figure 1). Context and query tokens share a single sequence, and the model predicts a flow-matching (Lipman et al., 2023; Liu et al., 2023) velocity at each query location. A single joint-predictive draw for all the query locations requires batched ODE integration of the velocity, which typically entails tens of network function evaluations (NFE) (Table 13). MFNP mitigates this cost via a single round of reflow distillation (Liu et al., 2023) to enable inference with only one NFE. As in latent-variable NPs, dependence between queries is carried by a shared latent, here a random-function initial condition transported by the learned flow (§4.3).

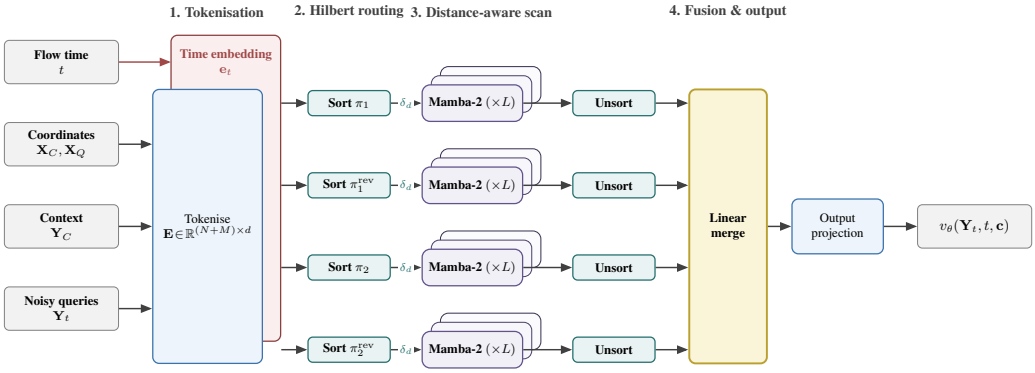

Figure 1: MFNP parametrises the velocity field $v_\theta(\mathbf{Y}_t, t, \mathbf{c})$ of §2.2 in four stages. The stages are tokenisation of context values and the noisy query slice $\mathbf{Y}_t$ with frozen Random Fourier Feature (RFF) coordinate embeddings; multi-directional Hilbert routing under $\Pi = \{\pi_1, \pi_1^{\text{rev}}, \pi_2, \pi_2^{\text{rev}}\}$; distance-aware Mamba-2 scans modulated by an adaLN-Zero flow-time embedding $\mathbf{e}_t$; and inverse-permutation fusion to per-query velocities. Trained with $\mathcal{L}_{\text{CFM}}$, reflow-distilled to a one-step Euler student, and CRPS fine-tuned. Full pipeline in Appendix B, Algorithm 1.

Contributions.

1. We introduce MambaFlowNP (MFNP), a joint-predictive NP that processes observations and prediction locations together in a shared sequence, scanned in multiple directions. Its computational cost grows linearly with the total number of points.

2. Through controlled ablations, we identify and quantify a single-state compression bottleneck in state-space NPs. A variant that compresses the context into one terminal hidden state has 10–17% higher MSE than the full model on synthetic tasks and 58–65% higher MSE on NOAA. This finding motivates MFNP's use of hidden states at individual query positions.

3. Across three NOAA temperature interpolation tiers, MFNP achieves lower RMSE and CRPS than the strongest joint-predictive baselines, with sampling speedups of 114–784×. We separately assess the contributions of reflow distillation and the linear-time scan to these speedups.
4. We demonstrate computational scalability to 10^6 points, with a single forward pass completing in under a second on one NVIDIA L40S GPU. Each forward pass requires $O(K)$ compute, where K is the total number of context and query points. All tested quadratic-time baselines exceed GPU memory at this scale.

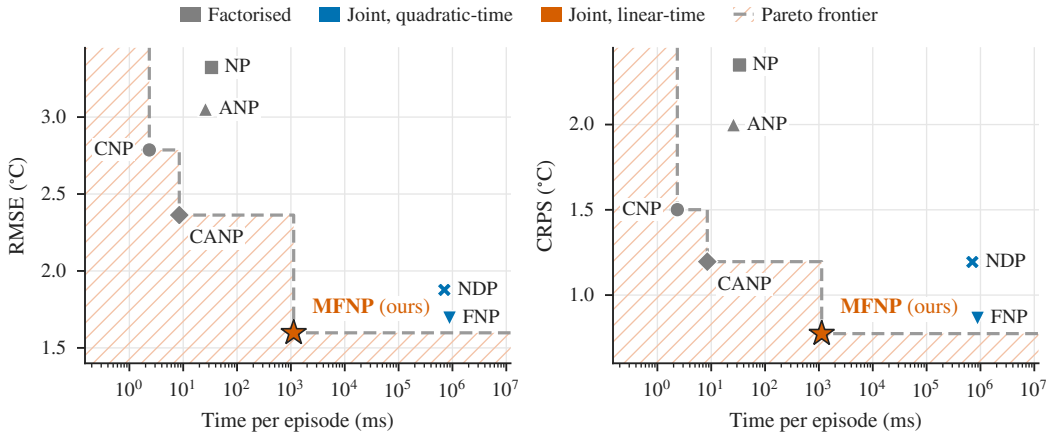

Figure 2: Per-episode latency vs. RMSE (left) and CRPS (right) on continental US NOAA (K=5,580). Means ±1 SE over 10 seeds at N_{samples}=128; the hatched staircase is the empirical Pareto frontier. TNPA exceeds the per-cell budget (§5.3); Utah and California frontiers in Appendix F.

2 Background

2.1 Neural processes

A neural process defines a conditional distribution $p(\mathbf{Y}_Q \mid \mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q)$ over target values $\mathbf{Y}_Q \in \mathbb{R}^M$ at query locations $\mathbf{X}_Q \in \mathbb{R}^{M\times D}$, given a context of N input–output pairs $\mathbf{X}_C \in \mathbb{R}^{N\times D}$, $\mathbf{Y}_C \in \mathbb{R}^N$. Let $K = N + M$ denote the total number of points. We tokenise each point separately, so K is also the sequence length. Training is episodic. Each iteration samples a dataset from a stochastic process, partitions points into context and query, and fits the predictive distribution against the ground-truth query values. At test time, the context and query locations of a new dataset are passed through the trained model in a single forward pass. NPs do not specify an explicit prior and then apply Bayes' rule; instead, they learn an implicit prior over functions from the data-generating stochastic process during episodic training, and the forward pass through $(\mathbf{X}_C, \mathbf{Y}_C)$ acts as an amortised update producing an approximation to the posterior predictive $p(\mathbf{Y}_Q \mid \mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q)$ over the query locations. We use *predictive* and *posterior predictive* interchangeably for this distribution.

2.2 Conditional flow matching and reflow distillation

Flow matching (Lipman et al., 2023; Albergo and Vanden-Eijnden, 2023; Liu et al., 2023) models approximate a velocity field u_t that transports samples from a source distribution p_0 to the conditional data distribution $p_1(\mathbf{Y}_1 \mid \mathbf{c}) \equiv p_{\text{data}}(\mathbf{Y}_1 \mid \mathbf{c})$, where $\mathbf{c} = \{\mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q\}$ denotes the neural process conditioning. Here $\mathbf{Y}_1$ equals the query values $\mathbf{Y}_Q \in \mathbb{R}^M$ targeted by the predictive, so $\mathbf{Y}_t \in \mathbb{R}^M$ is indexed by the query locations. The velocity field is determined by a choice of conditional flow path or interpolant $\mathbf{Y}_t = a_t\mathbf{Y}_1 + b_t\mathbf{Y}_0$,

$$u_t(\mathbf{Y}_t; \mathbf{c}) = \int \frac{d\mathbf{Y}_t}{dt} \frac{p_t(\mathbf{Y}_t \mid \mathbf{Y}_1, \mathbf{c}) p_1(\mathbf{Y}_1 \mid \mathbf{c})}{p_t(\mathbf{Y}_t \mid \mathbf{c})} d\mathbf{Y}_1 \tag{1}$$

where $a_t, b_t \in \mathbb{R}$, $\mathbf{Y}_1 \sim p_1(\cdot \mid \mathbf{c})$, $\mathbf{Y}_0 \sim p_0$, and $\frac{d\mathbf{Y}_t}{dt}$ is the *conditional* velocity. Directly regressing a neural parametrisation $v_\theta(\mathbf{Y}_t, t, \mathbf{c})$ to $u_t(\mathbf{Y}_t; \mathbf{c})$ is intractable due to the marginalisation in the definition of u_t. The *conditional* flow-matching loss is tractable and has the same gradient (Lipman et al., 2023),

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\mathbf{Y}_1,\mathbf{Y}_0,\mathbf{c}}\left[\left\| v_\theta(\mathbf{Y}_t, t, \mathbf{c}) - \frac{d\mathbf{Y}_t}{dt} \right\|^2\right] \tag{2}$$

with $t \sim \mathcal{U}(0,1)$. Samples are drawn by integrating $\hat{\mathbf{Y}}_1 = \mathbf{Y}_0 + \int_0^1 v_\theta(\mathbf{Y}_t, t, \mathbf{c})dt$ numerically (e.g. Euler). Drawing inspiration from work on data-dependent source distributions (Tong et al., 2024; Kollovieh et al., 2025), we use a query-geometry-dependent Gaussian source distribution $p_0(\mathbf{Y}_0 \mid \mathbf{X}_Q) = \mathcal{N}(0, \Sigma(\mathbf{X}_Q))$, obtained by evaluating a random-feature Gaussian-process draw at the query locations. This gives a spatially correlated source, of rank at most F, that shortens the transformation from source samples to smooth spatial fields (§4.2). The network also receives a per-draw latent $\mathbf{z}$ and routing offset $\mathbf{o}$ (§4), which we omit from the notation $v_\theta(\mathbf{Y}_t, t, \mathbf{c})$. The gradient equivalence above assumes a network without these inputs. With them, the minimiser of $\mathcal{L}_{\mathrm{CFM}}$ is $\mathbb{E}[\mathbf{Y}_1 - \mathbf{Y}_0 \mid \mathbf{Y}_t, t, \mathbf{c}, \mathbf{z}, \mathbf{o}]$ rather than u_t; reflow and CRPS fine-tuning (§4.2) then fit the one-step map directly. We use the linear interpolant $a_t = t$, $b_t = (1-t)$, with conditional velocity $\frac{d\mathbf{Y}_t}{dt} = \mathbf{Y}_1 - \mathbf{Y}_0$. Although this conditional flow has straight trajectories, the underlying velocity field typically induces curved trajectories, leading to inaccurate results with few solver steps (Liu et al., 2023; Geng et al., 2025). To obtain high-quality samples with fewer steps, we use reflow distillation (Liu et al., 2023). A pre-trained teacher model generates coupled $(\mathbf{Y}_0, \hat{\mathbf{Y}}_1)$ pairs, and the student model is optimised on $\mathcal{L}_{\mathrm{CFM}}$ against these couplings rather than independently drawn source and target samples. The full training pipeline is described in §5.2 and Appendix H.

3 Related work

Factorised and latent-variable NPs. Conditional Neural Processes (Garnelo et al., 2018a) encode the context with a permutation-invariant set encoder and predict independent Gaussian distributions at query locations. Latent-variable NPs (Garnelo et al., 2018b) introduce dependence across queries through a shared Gaussian latent, trained under a variational ELBO. Attentive NP (Kim et al., 2019) replaces mean-pooling with cross-attention from each query to the context. Convolutional Conditional NPs (Gordon et al., 2020) use grid-based convolutions to encode translation equivariance, an inductive bias suited to stationary spatial fields. AR-CNP (Bruinsma et al., 2023) introduces cross-query dependence by sampling from a CNP autoregressively, requiring M sequential forward passes for M queries.

Joint-predictive NPs. Gaussian NPs (Bruinsma et al., 2021) explicitly model cross-query correlations through a joint Gaussian predictive. TNPA (Nguyen and Grover, 2022) decodes queries autoregressively under causal self-attention. NDP (Dutordoir et al., 2023) and FNP (Abu Hamad and Rosenbaum, 2025) use transformer backbones trained with score-based diffusion (Song et al., 2021) and flow-matching (Lipman et al., 2023) objectives, respectively. In these transformer models, self-attention incurs $O(K^2)$ compute per network evaluation. Spatial coordinates enter through token embeddings, whereas MFNP also uses them to determine the scan order.

Mamba on point sets. Mamba (Gu and Dao, 2023) and Mamba-2 (Dao and Gu, 2024) introduced selective state-space models with input-dependent gating and computational cost linear in sequence length. VMamba (Liu et al., 2024) scans regular image grids in four directions. MFNP adapts this four-direction scheme to irregular point sets through two Hilbert orderings and their reversals. PointMamba (Liang et al., 2024) serialises 3D point clouds along Hilbert curves for supervised shape analysis, while Voxel Mamba (Zhang et al., 2024) applies Hilbert serialisation to sparse voxels for 3D detection. These methods use spatial scans for visual recognition tasks. MFNP applies multi-directional scans to amortised joint-predictive inference for continuous 2D fields, with episodic training over varying numbers of context and query points (N, M).

Scalable GPs and spatial statistics. Exact GP inference for kriging requires an $O(N^3)$ covariance factorisation for N observed locations. Sparse variational GPs (Titsias, 2009), Vecchia approximations and nearest-neighbour GPs (Vecchia, 1988; Datta et al., 2016), and structured kernel interpolation (Wilson and Nickisch, 2015) reduce this computational burden, and the SPDE approach

represents Matérn fields as sparse Gaussian Markov random fields (Lindgren et al., 2011). These approaches make inference within an individual field more tractable, but each still fits hyperparameters per field. NPs amortise inference across fields. Episodic training learns a map from context observations to a predictive distribution, which can then be applied to a new field without further optimisation.

In-context learning and prior-fitted networks. NPs learn to infer from context observations without updating their parameters at test time. Prior-fitted networks (Müller et al., 2022) similarly train transformers on datasets sampled from a prior to approximate the posterior predictive. Mamba has also been shown to learn regression tasks in context (Grazzi et al., 2024).

MFNP builds on established components, namely Mamba-2 scans (Dao and Gu, 2024), Hilbert serialisation (Liang et al., 2024; Zhang et al., 2024), conditional flow matching (Lipman et al., 2023), GP-based source distributions (Kollovieh et al., 2025), and reflow distillation (Liu et al., 2023). Our contributions are the shared context–query scan for joint-predictive neural processes and the analysis of how scan routing affects predictive performance.

4 MambaFlowNP

MFNP processes irregularly placed observations in one or two dimensions with an $O(K)$ neural recurrence. A direct SSM-based NP adaptation compresses the context into a single fixed-dimensional hidden state and conditions a query decoder on that state. MFNP avoids the compression by maintaining a hidden state at every token position throughout the scan. Context and query tokens share the sequence, so each query is decoded from its own position. The single-state ablation forces the context through one hidden vector before reading queries (§6.4).

4.1 Architecture

For each joint draw, a shared latent $\mathbf{z}$ determines the random initial values at the query locations and conditions the network. Given the current query values $\mathbf{Y}_t$, flow time t and context $\mathbf{c}$, the network computes the velocity $v_\theta(\mathbf{Y}_t, t, \mathbf{c})$ in the four stages shown in Figure 1.

(1) Tokenise. For each point, we concatenate a coordinate embedding computed using frozen Random Fourier Features (Rahimi and Recht, 2007) with its scalar value, which is the observed value at a context point or $Y_{t,i}$ at a query. Context and query tokens use separate linear projections.

(2) Route. We use Hilbert curves (Sagan, 1994; Butz, 1971) to order the combined context and query points so that nearby locations tend to remain close in the sequence. Two orderings, π_1 and π_2, use the original and swapped coordinate axes. Together with their reversals, they form

$$\Pi = \{\pi_1, \pi_1^{\mathrm{rev}}, \pi_2, \pi_2^{\mathrm{rev}}\}.$$

A shared random offset $\mathbf{o}$ shifts the routing grid to reduce artefacts at fixed curve boundaries. In one dimension, we scan points in ascending and descending coordinate order, repeating these two orderings to fill the four direction slots (Appendix B).

(3) Scan. A Mamba-2 block propagates information through a recurrent hidden state. Context tokens write to the recurrent state. Query tokens read the state at their own positions but do not write to it, since they enter the scan with zero input and zero step size; the block has no short convolution. A single L-layer stack of Mamba-2 blocks, with parameters shared across directions, processes each direction-permuted sequence. Every block is conditioned through adaLN-Zero (Peebles and Xie, 2023) on the flow time and the shared latent $\mathbf{z}$. Its state-update gate also uses the distance $\delta_{d,i}$ to the nearest preceding context point in the scan order.

(4) Fuse and read out. The per-direction hidden states are restored to the original point order and linearly fused, and a final adaLN-conditioned projection maps each query's fused state to its velocity. Implementation settings and the full forward pass are given in Appendix B, Algorithm 1.

4.2 Training and sampling

Random-function source distribution. Each draw samples a latent $\mathbf{z} \sim \mathcal{N}(0, I_F)$ and sets

$$Y_{0,i} = \boldsymbol{\psi}(\mathbf{x}_{Q,i})^\top \mathbf{z},$$

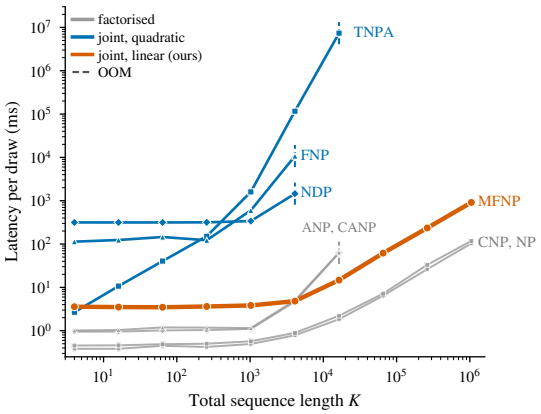

Figure 3: End-to-end sampling latency vs. K (log–log) on 2D Gaussian Matérn-$3/2$, batch$=1$, $N_{\text{samples}}=1$ (1-NFE Euler for MFNP; published step count per baseline). Colour gives the model class; a short dashed line through a curve's last point means the next K runs out of memory. Numbers in Table 8.

where ψ is a fixed F-dimensional random Fourier feature map. This is the source p_0 of §2.2. It evaluates the same random function at every query location, providing spatially dependent initial values. It replaces the standard $\mathbf{Y}_0$ draw in the CFM loss and at inference (Algorithms 2 and 3).

Flow matching and distillation. We first train a teacher using the conditional flow-matching objective in §2.2. The teacher then generates paired initial and final samples for one round of reflow distillation. The resulting student generates a joint draw with one Euler step, requiring a single network evaluation (Appendix H).

CRPS fine-tuning. On NOAA, a short final stage fine-tunes the student on the continuous ranked probability score of its own one-step samples. For each training episode, we draw S samples $\hat{y}^{(1)}, \ldots, \hat{y}^{(S)}$ at every query and minimise the fair CRPS (Gneiting and Raftery, 2007; Ferro, 2014)

$$\mathcal{L}_{\text{CRPS}} = \frac{1}{S}\sum_{s}\left|\hat{y}^{(s)} - y\right| - \frac{1}{2S(S-1)}\sum_{s\neq s'}\left|\hat{y}^{(s)} - \hat{y}^{(s')}\right|, \tag{3}$$

averaged over queries. The stage early-stops on the CRPS of held-out validation episodes (Appendix A). Baselines do not receive this stage.

4.3 Properties and computational cost

For distinct Hilbert sort keys, deterministic sorting makes MFNP invariant to context-row permutations and equivariant to query-row permutations.

It is also consistent under query-set marginalisation. Queries only read the scan state, and all randomness shared between queries is the latent $\mathbf{z}$ and offset $\mathbf{o}$ of each draw, so each draw is one random function of the context evaluated at the queries. The joint predictive over any query set therefore has at most F degrees of freedom, as in latent-variable NPs. For the same context, latent and offset, adding, removing, or chunking queries leaves the values at the other locations unchanged, and, given the context, the predictive defines a Kolmogorov-consistent stochastic process over query locations. Recent transformer NPs relax this property for efficiency (Hassan et al., 2026; Mortimer et al., 2026). Because queries do not write to the scan state, the context states can be computed once and reused across query sets for a fixed draw and flow time.

Inference complexity. The random-function source costs $O(MF)$, which is linear in the query count for fixed feature dimension F. The Mamba-2 recurrence has computational complexity $O(K)$ in sequence length by construction. The routing permutations are obtained by sorting Hilbert keys, also $O(K)$ in the operating regime (Appendix C). Each forward pass therefore costs $O(K)$ for fixed feature dimension, state size and depth.

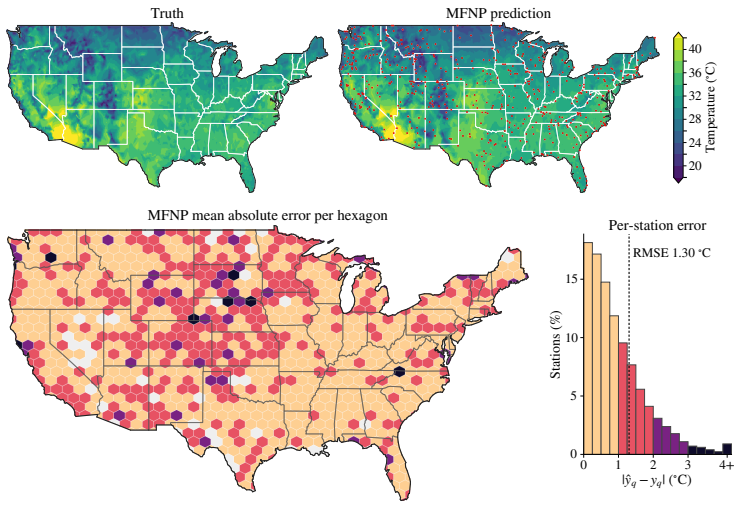

Figure 4: Continental US (K=5,580) posterior on a held-out date, $10\%/90\%$ context/query split. Top, truth and MFNP posterior mean (red dots are context stations), interpolated for display. Bottom, MFNP absolute error averaged over the query stations in each $\sim$110 km hexagon (grey hexagons hold none), with the per-station histogram as legend and the RMSE dashed. Population-averaged accuracy in Figure 5.

5 Experimental setup

5.1 Tasks

Synthetic Gaussian processes. Each episode draws from a fresh GP, following the standard settings of TNP (Nguyen and Grover, 2022). 1D tasks sample $K \leq K_{\max}$=128 points uniformly in $[-2, 2]$; 2D tasks sample $K \leq K_{\max}$=256 points uniformly in $[0, 1]^2$. The simulator draws a latent field $f \sim \mathcal{GP}(0, k_{\ell,\sigma_f^2})$ and observations $y = f + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n^2 I)$. Kernel hyperparameters are resampled per episode from $\ell \sim \mathcal{U}(0.1, 0.6)$ and $\sigma_f \sim \mathcal{U}(0.1, 1.0)$, with $\sigma_n = 0.02$, so the model must infer smoothness and amplitude from context. Context and query sizes vary per episode with $N \sim \mathcal{U}\{16, K_{\max} - 16\}$ and $M \sim \mathcal{U}\{16, K_{\max} - N\}$. We use RBF, Matérn- $3/2$, and Matérn-$5/2$ kernels in 1D, and Matérn-$3/2$ in 2D.

NOAA temperature interpolation. We use daily maximum temperature (TMAX) records from the NOAA Global Historical Climatology Network (Menne et al., 2012b,a) over 2005-01-01 to 2024-12-31, split temporally into train ($\leq$ 2016), validation (2017–2019), and test (2020–2024) sets. These cover **Utah** (K=222), **California** (K=486) and the continental **US** (K=5,580), three geographic tiers of increasing extent. The same station set is reused across splits, so generalisation is across dates rather than locations; the task is therefore infilling at stations seen during training. Each NOAA episode draws a date uniformly from the appropriate split and partitions that date's observed stations $50/50$ at random into context and query. Coordinate projection, station selection, and target scaling are in Appendix A.

5.2 Training

MFNP trains a CFM teacher for 80k steps and a reflow student for a further 20k steps. On NOAA, this is followed by the CRPS fine-tuning stage of §4.2. On Utah, California and the US, the student trains on 5,000, 5,000 and 1,000 teacher couplings. CRPS fine-tuning uses batches of 8, 8 and 2 episodes with 8, 8 and 4 samples per episode, a learning rate of 10^{-4} scaled by the square root of the batch size relative to 8; it is scored on 64 validation episodes from 2017–2019 every 1,000 steps, stops after 5 evaluations without improvement or at 40k steps, and keeps the best weights. Baselines train for 100k steps with no student or CRPS phase; on NOAA, MFNP receives up to 140k optimiser updates in total. Per-cell hyperparameters (hidden size, depth, learning rate, batch size) come from Bayesian-optimisation sweeps on validation MSE, with a single training run at the

Table 1: 1D and 2D Gaussian Matérn-$3/2$ (K=128, 256). MSE, CRPS (↓); C2ST (target 0.5); 95% coverage (target nominal); time = per-episode ms, MFNP and baselines on one NVIDIA L40S. Mean ± SE over 10 evaluation seeds; bold = best per column (within 1 SE). † our method; full kernel sweep in Appendix D.

	Predictive Quality		Calibration Quality		
Model	**MSE** ↓	**CRPS** ↓	**C2ST** → 0.5	**Cov-95** → 0.95	**Time (ms)** ↓
1D Gaussian – Matérn-$3/2$ (K=128)					
Joint predictives, quadratic-time					
TNPA	$\mathbf{0.0110 \pm 0.0003}$	$\mathbf{0.0367 \pm 0.0004}$	$\mathbf{0.499 \pm 0.003}$	$\mathbf{0.946 \pm 0.001}$	735.7 ± 0.4
NDP	0.1720 ± 0.0028	0.2508 ± 0.0014	0.688 ± 0.018	0.996 ± 0.000	$(2.174 \pm 0.004) \times 10^3$
FNP	$\mathbf{0.0107 \pm 0.0002}$	0.0398 ± 0.0004	0.532 ± 0.011	0.752 ± 0.001	$(1.253 \pm 0.001) \times 10^3$
Joint predictives, linear-time					
MFNP†	0.0111 ± 0.0003	0.0398 ± 0.0004	0.537 ± 0.006	0.938 ± 0.001	$\mathbf{29.5 \pm 0.2}$
2D Gaussian – Matérn-$3/2$ (K=256)					
Joint predictives, quadratic-time					
TNPA	$\mathbf{0.0126 \pm 0.0004}$	$\mathbf{0.0453 \pm 0.0006}$	$\mathbf{0.488 \pm 0.002}$	$\mathbf{0.938 \pm 0.001}$	$(3.480 \pm 0.002) \times 10^3$
NDP	0.0443 ± 0.0007	0.1089 ± 0.0007	0.668 ± 0.009	0.992 ± 0.000	$(3.158 \pm 0.004) \times 10^3$
FNP	0.0137 ± 0.0004	0.0503 ± 0.0007	0.573 ± 0.019	0.630 ± 0.001	$(3.526 \pm 0.002) \times 10^3$
Joint predictives, linear-time					
MFNP†	0.0167 ± 0.0005	0.0537 ± 0.0005	0.554 ± 0.005	0.971 ± 0.000	$\mathbf{103.6 \pm 0.3}$

BO-best configuration. Optimiser and schedule are shared across models (AdamW, warmup-cosine to 10^{-5}, EMA β=0.9995); per-cell values and the BO protocol are in Appendix A, and the reflow coupling and solver details are in Appendix H. All training and evaluation run on NVIDIA L40S GPUs. Each (model, task) cell is capped at 100 GPU-hours covering training and all 10 evaluation seeds; cells exceeding the cap are reported as T/O.

5.3 Evaluation protocol

Metrics. Predictive quality is scored by mean-squared error (MSE) on the predictive mean and the continuous ranked probability score (CRPS) (Gneiting and Raftery, 2007), each averaged over N_{samples}=128 draws per target shared across methods to control the $O(1/N_{\text{samples}})$ bias in the standard empirical CRPS estimator. Calibration is scored by the classifier two-sample test (C2ST) (Lopez-Paz and Oquab, 2017) against GP reference draws (synthetic only) and the empirical coverage of the central 95% interval of the 128 samples, both computed at each query location and averaged over locations. All numbers are mean ± one standard error across 10 evaluation seeds for a single trained model per cell, each covering 256 synthetic episodes or 1,024 NOAA episodes on a fresh date; evaluation sets are cached so every method sees identical test data.

Latency. Per-episode wall-clock is the median of 10 timed forward passes after 5 warm-ups on a single NVIDIA L40S, each pass a full N_{samples}=128-draw call on one held-out episode, that is, the time to generate 128 joint predictive draws over all query locations. Timings include the Hilbert sort. MFNP times use the release sampler, 8 routing offsets of 16 draws each. Each method runs its deployment configuration. MFNP uses 1-NFE Euler from the reflow student, FNP and NDP use their published ODE/diffusion step counts, and TNPA uses autoregressive decoding. TNPA at NOAA US (K=5,580) is reported as T/O because its serial-within-a-draw decoding exceeds the per-cell budget by an order of magnitude (Appendix A).

6 Results

6.1 Synthetic Gaussian processes

MFNP is competitive with the strongest joint-predictive baselines on predictive quality at matched point counts and is 25–42× faster (Table 1; full kernel sweep in Appendix D). On 1D Matérn-$3/2$, MFNP matches TNPA on MSE within one SE and FNP on CRPS, at 30 ms per episode against 736 ms for TNPA. On 2D Matérn-$3/2$, MFNP records ∼1.2× higher MSE than FNP at ∼34× lower latency. MFNP coverage stays within three points of 0.95 on the Matérn kernels and nine points on RBF, whereas FNP under-covers by 20–32 points and NDP over-covers.

6.2 NOAA temperature interpolation

MFNP attains the lowest RMSE and CRPS on every tier and runs 114–784× faster than the strongest transformer baseline (Table 2; factorised baselines and per-tier Pareto frontiers in Appendix F). Its 95% predictive coverage is closest to nominal among the evaluated joint-predictive models, reaching 0.940 on Utah, 0.945 in California and 0.955 across the continental US. The respective contributions of distillation and architecture to the speedup are examined in §6.3.

Exact, sparse and nearest-neighbour GPs fit to each episode serve as spatial references, with MSE 3.7–6.8× that of MFNP and Cov-95 of 0.92–0.94 (Table 10). A representative prediction on the continental US is shown in Figure 4.

6.3 Computational scaling

End-to-end sampling. MFNP reaches $K = 10^6$ in under a second on one NVIDIA L40S. We measure the time to generate one joint predictive draw ($N_{\text{samples}} = 1$, batch$= 1$) on 2D Gaussian Matérn-$3/2$, using each method's deployment configuration (Figure 3, Table 8). FNP and NDP exhaust memory by $K = 16{,}384$, and TNPA exhausts memory by $K = 65{,}536$. Even before reaching its memory limit, TNPA takes ∼117 s at $K = 4{,}096$ and ∼2 hours at $K = 16{,}384$.

Reflow versus architecture. To separate the effect of fewer sampling steps from the cost of each forward pass, we compare the architectures at matched 1-NFE and parameter count (Figure 10, Appendix H). MFNP, FNP, and NDP share a ∼1.5–3.5 ms dispatch floor for $K \leq 1{,}024$, so the order-of-magnitude speedup at small and moderate K is attributable to reflow distillation. At larger K, quadratic attention becomes more expensive and eventually exceeds the memory budget, while MFNP retains linear-time computation. Both effects contribute to the 628–784× speedup over NDP and FNP on NOAA US (Table 2). At the US tier, NDP and FNP fit the per-GPU memory budget only via sample-axis chunking, which preserves their predictive distribution while retaining the quadratic per-step cost in the point count.

Table 2: NOAA temperature interpolation on Utah (K=222), California (K=486), and the continental US (K=5,580); joint-predictive baselines only (full table with factorised baselines in Table 9). MSE (°C)^2, RMSE, CRPS °C (↓); Cov-95 (target 0.95); C2ST omitted (no tractable reference posterior on real fields); time = per-episode ms on one NVIDIA L40S. Mean ± SE over 10 evaluation seeds, denormalised scale; bold = best per column within tier (within 1 SE). † our method.

	Predictive Quality			Calibration	
Model	**MSE (°C)^2 ↓**	**RMSE (°C) ↓**	**CRPS (°C) ↓**	**Cov-95 → 0.95**	**Time (ms) ↓**
NOAA Utah (K=222)					
Joint predictives, quadratic-time					
TNPA	2.524 ± 0.007	1.589 ± 0.002	0.783 ± 0.001	0.846 ± 0.000	$(4.923 \pm 0.004) \times 10^3$
NDP	4.665 ± 0.012	2.160 ± 0.003	1.221 ± 0.002	0.742 ± 0.000	$(2.414 \pm 0.004) \times 10^3$
FNP	2.712 ± 0.007	1.647 ± 0.002	0.853 ± 0.001	0.730 ± 0.000	$(3.071 \pm 0.003) \times 10^3$
Joint predictives, linear-time					
MFNP†	$\mathbf{2.464 \pm 0.016}$	$\mathbf{1.570 \pm 0.005}$	$\mathbf{0.767 \pm 0.002}$	$\mathbf{0.940 \pm 0.000}$	$\mathbf{43.3 \pm 0.4}$
NOAA California (K=486)					
Joint predictives, quadratic-time					
TNPA	1.898 ± 0.005	1.378 ± 0.002	0.701 ± 0.001	0.938 ± 0.000	$(3.004 \pm 0.002) \times 10^4$
NDP	3.071 ± 0.008	1.753 ± 0.002	1.064 ± 0.001	0.590 ± 0.000	$(9.154 \pm 0.015) \times 10^3$
FNP	2.395 ± 0.008	1.548 ± 0.003	0.839 ± 0.001	0.785 ± 0.001	$(8.844 \pm 0.004) \times 10^3$
Joint predictives, linear-time					
MFNP†	$\mathbf{1.808 \pm 0.006}$	$\mathbf{1.344 \pm 0.002}$	$\mathbf{0.681 \pm 0.001}$	$\mathbf{0.945 \pm 0.000}$	$\mathbf{67.9 \pm 0.5}$
NOAA US (K=5,580)					
Joint predictives, quadratic-time					
TNPA	T/O	T/O	T/O	T/O	T/O
NDP	3.528 ± 0.087	1.878 ± 0.023	1.194 ± 0.013	0.322 ± 0.005	$(7.075 \pm 0.020) \times 10^5$
FNP	2.872 ± 0.064	1.695 ± 0.019	0.864 ± 0.009	0.786 ± 0.001	$(8.831 \pm 0.002) \times 10^5$
Joint predictives, linear-time					
MFNP†	$\mathbf{2.554 \pm 0.010}$	$\mathbf{1.598 \pm 0.003}$	$\mathbf{0.774 \pm 0.001}$	$\mathbf{0.955 \pm 0.001}$	$\mathbf{(1.126 \pm 0.000) \times 10^3}$

6.4 Ablations

Each row of Table 11 changes one model component (Appendix G).

Context compression. Compressing the context into a single state gives 10–17% higher MSE than the full model on synthetic tasks and 58–65% higher MSE on NOAA. These results support decoding queries from states at their own scan positions rather than from one terminal context state.

Spatial routing. Random ordering raises MSE by 1.2–1.7× on synthetic tasks and 2.0–5.0× on NOAA (Table 11). Reducing the 4-direction routing to one direction inflates MSE by 1.6–4.2×; using two directions increases MSE by 3–9% on the headline NOAA tiers while using 31–74% of the runtime, and matches or improves on the full model's synthetic MSE.

Morton and lexicographic curves give 4% and 9% lower MSE than the Hilbert curve on 2D synthetic data, and lexicographic ordering raises CRPS by 16% (Table 12). The boundary diagnostic gives comparable boundary-to-interior MSE ratios for MFNP and FNP, with 1.40 versus 1.54 on Utah and 1.43 versus 1.47 on California (Appendix I).

Sampling. The deployment configuration uses one-step Euler sampling from the reflow-distilled student. The teacher–student and solver comparisons are reported in Table 13 (Appendix H). On the student, multi-step solvers do not reduce MSE on any task. On 2D Matérn-$3/2$, 15-step Euler gives an MSE of 0.0166 and a CRPS of 0.0537, against 0.0167 and 0.0537 for one-step Euler, and 15- and 30-step Euler take 14 and 28 times as long. Heun-1 inflates NOAA MSE by 1.58–1.86× over Euler-1 and is not used.

7 Conclusion

MambaFlowNP combines joint prediction on irregular spatial point sets with $O(K)$ computation per forward pass. Reflow distillation enables one-pass sampling. Across the three NOAA tiers, MFNP attains the lowest RMSE and CRPS among the evaluated joint-predictive models, with near-nominal coverage and 114–784× lower latency than the strongest transformer baseline. On synthetic Gaussian processes, it matches TNPA's MSE within one SE on 1D Matérn-$3/2$ and runs 25–42× faster than TNPA and FNP on Matérn-$3/2$. A single forward pass at 10^6 points completes in under a second on one NVIDIA L40S GPU, demonstrating the computational feasibility of joint prediction over large, irregular spatial point sets.

Limitations. MFNP targets irregular point sets. On regularly gridded domains, convolutional NPs and FFT-based GP solvers exploit grid structure directly and are typically more efficient. At small K ($\lesssim 10^3$) MFNP's per-call dispatch floor leaves it slower than the simplest factorised NPs (Tables 7 and 9). Its scaling advantage over quadratic attention emerges at larger point counts. We evaluate one- and two-dimensional inputs only; space–time inputs need one extra coordinate in the Hilbert ordering, such as a three-dimensional curve over (x, y, t), and we leave their evaluation to future work.

Reflow distillation adds a coupled-pair generation pass to the training pipeline that single-stage baselines do not incur. The reported inference speedups combine the effects of distillation and the architecture.

Our real-data evaluation is limited to one dataset and one variable, NOAA daily maximum temperature. The split is temporal within a fixed station network, and transfer to held-out regions remains untested. Evaluation against scalable spatial baselines is incomplete, as structured kernel interpolation, local kriging, graph-based methods and AR-ConvCNP (Bruinsma et al., 2023) remain to be tested. Predictive quality at 10^6 points is also unestablished.

Code and reproducibility. MambaFlowNP code, trained weights, and training and evaluation scripts for the three NOAA tiers are available at `https://github.com/CosmoNaught/MambaFlowNP`.

Acknowledgments and Disclosure of Funding

CS was supported by the Wellcome Trust (226072/Z/22/Z). GC's research was funded in whole, or in part, by the Wellcome Trust (220900/Z/20/Z). TJH and OJW acknowledge support from the Academy

of Medical Sciences, the Wellcome Trust, the Government Department of Science, Innovation and Technology, the British Heart Foundation and Diabetes UK (Springboard Award SBF0010\1221). OJW is also supported by an Imperial College Research Fellowship sponsored by Schmidt Sciences. ES acknowledges support in part from the AI2050 program at Schmidt Sciences (Grant G-22-64476). All authors acknowledge funding from the Medical Research Council (MRC) Centre for Global Infectious Disease Analysis (reference MR/X020258/1), funded by the United Kingdom MRC. We thank Richard E. Turner and Cristiana-Diana Diaconu for generously sharing their time and for insightful discussions on neural processes and state-space models during the development of this work. We thank the Research Software for Infectious Disease Epidemiology (RESIDE) team at Imperial College London for access to their computing cluster. The GPU compute was funded by MRC Mid-Range Equipment Funding (UKRI2681). For the purpose of open access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Hussen Abu Hamad and Dan Rosenbaum. Flow matching neural processes. In *Advances in Neural Information Processing Systems*, 2025.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Wessel P Bruinsma, James Requeima, Andrew Y K Foong, Jonathan Gordon, and Richard E Turner. The Gaussian neural process. In *Advances in Approximate Bayesian Inference*, 2021.

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner. Autoregressive conditional neural processes. In *International Conference on Learning Representations*, 2023.

Arthur R Butz. Alternative algorithm for Hilbert's space-filling curve. *IEEE Transactions on Computers*, C-20(4):424–426, 1971.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*. PMLR, 2024.

Abhirup Datta, Sudipto Banerjee, Andrew O. Finley, and Alan E. Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. In *International Conference on Machine Learning*, pages 8990–9012. PMLR, 2023.

Christopher A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

Marta Garnelo, Dan Rosenbaum, Christopher J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and S M Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, S M Ali Eslami, and Yee Whye Teh. Neural processes. In *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018b.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

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.

Jonathan Gordon, Wessel P Bruinsma, Andrew Y K Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020.

Riccardo Grazzi, Julien Siems, Simon Schrodi, Thomas Brox, and Frank Hutter. Is Mamba capable of in-context learning? In *International Conference on Automated Machine Learning*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Chris H. Hamilton and Andrew Rau-Chaplin. Compact Hilbert indices: Space-filling curves for domains with unequal side lengths. *Information Processing Letters*, 105(5):155–163, 2008.

Conor Hassan, Nasrulloh Loka, Cen-You Li, Daolang Huang, Paul E. Chang, Yang Yang, Francesco Silvestrin, Samuel Kaski, and Luigi Acerbi. Efficient autoregressive inference for transformer probabilistic models. In *International Conference on Learning Representations*, 2026.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with Gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, 2025.

Jia Li, Yanyan Shen, Lei Chen, and Charles Wang Wai Ng. Rainfall spatial interpolation with graph neural networks. In *Database Systems for Advanced Applications (DASFAA)*, volume 13946 of *Lecture Notes in Computer Science*, pages 175–191. Springer, 2023. doi: 10.1007/978-3-031-30678-5_14.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems*, 2024.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Yaron Lipman, Ricky T Q Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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*, 2023.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, 2024.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

Matthew J Menne, Imke Durre, Bryant Korzeniewski, Shelley McNeill, Kristy Thomas, Xungang Yin, Steven Anthony, Ron Ray, Russell S Vose, Byron E Gleason, and Tamara G Houston. Global historical climatology network - daily (GHCN-daily), version 3, 2012a.

Matthew J Menne, Imke Durre, Russell S Vose, Byron E Gleason, and Tamara G Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7):897–910, 2012b. doi: 10.1175/JTECH-D-11-00103.1.

Duane Merrill. CUB: Cooperative primitives for CUDA C++. https://github.com/NVIDIA/cccl, 2016.

Bongki Moon, H. V. Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the Hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141, 2001.

Philip Mortimer, Cristiana Diaconu, Tommy Rochussen, Bruno Mlodozeniec, and Richard E. Turner. Incremental transformer neural processes. In *International Conference on Machine Learning*, 2026.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pages 16569–16594. PMLR, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, pages 4195–4205, 2023.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

Hans Sagan. *Space-Filling Curves*. Universitext. Springer, 1994.

Cosmo Santoni. Compiler-first state space duality and portable $O(1)$ autoregressive caching for inference. *arXiv preprint arXiv:2603.09555*, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

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.

Anna Vaughan, Will Tebbutt, J. Scott Hosking, and Richard E. Turner. Convolutional conditional neural processes for local climate downscaling. *Geoscientific Model Development*, 15(1):251–268, 2022. doi: 10.5194/gmd-15-251-2022.

Aldo V. Vecchia. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312, 1988.

Daniel J. Weiss, Paulina A. Dzianach, Adam Saddler, Jailos Lubinda, Annie Browne, Michael McPhail, Susan F. Rumisha, Francesca Sanna, Yalemzewod Gelaw, Juniper B. Kiss, Sarah Hafsia, Rubini Jayaseelen, Hunter S. Baggen, Punam Amratia, Amelia Bertozzi-Villa, Olivia Nesbit, Joanna Whisnant, Katherine E. Battle, Michele Nguyen, Kefyalew Addis Alene, Ewan Cameron, Melissa A. Penny, Samir Bhatt, David L. Smith, Tasmin L. Symons, Jonathan F. Mosser, Christopher J. L. Murray, Simon I. Hay, and Peter W. Gething. Mapping the global prevalence, incidence, and mortality of *Plasmodium falciparum* and *Plasmodium vivax* malaria, 2000–22: a spatial and temporal modelling study. *The Lancet*, 405(10483):979–990, 2025. doi: 10.1016/S0140-6736(25)00038-8.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Guowen Zhang, Lue Fan, Chenhang He, Zhen Lei, Zhaoxiang Zhang, and Lei Zhang. Voxel Mamba: Group-free state space models for point cloud based 3D object detection. In *Advances in Neural Information Processing Systems*, 2024.

A Hyperparameters

All models train with AdamW (weight decay 10^{-4}, the optax default; gradient clip 1.0), a warmup-cosine learning-rate schedule (linear warmup over $\min(100, T-1)$ steps, cosine decay to 10^{-5}), and EMA weights (β=0.9995). Per-model hyperparameters come from a two-phase Bayesian-optimisation sweep, with a 16-trial coarse pass plus a 16-trial narrowed pass around the coarse optimum, giving 32 trials per (model, task) cell, each optimising validation MSE. On the synthetic tasks, MFNP is tuned by one Bayesian-optimisation sweep of 41 to 50 trials per task over hidden width, state size, depth, RFF frequencies and bandwidth, learning rate and batch size, each trial trained for 20k flow-matching and 5k reflow steps and scored by validation MSE on 64 fixed episodes drawn from the task prior, and the best configuration is retrained with the full budget.

Latency methodology (§5.3). Baselines retain their authors' attention implementations; the additive context–query mask routes PyTorch SDPA to the memory-efficient kernel rather than FlashAttention, so reported OOM thresholds reflect $O(K^2)$-memory attention. TNPA's autoregressive decoder runs M sequential self-attention passes per draw and is serial within a draw, so sample chunking cannot reduce its cost; on NOAA US the projected per-seed runtime exceeds the per-cell budget by an order of magnitude.

NOAA preprocessing (§5.1). The Utah tier spans 37–42°N, 109–114°W; California spans 32–42°N, 114–124.5°W. Stations are kept if they report on at least 80% of train-window days, with top-coverage fallback to the per-tier cap. Station coordinates are equirectangular-projected to kilometres and axis-wise min-max scaled to $[0,1]^2$; observations pass through a Gaussian quantile transform fit on the train window, with metrics reported on the denormalised °C scale.

B Detailed algorithms

We first specify the feature maps, routing settings and scan implementation, then give the forward pass, CFM training step and sampling procedure. The coordinate embedding ϕ and the random-function feature map $\boldsymbol{\psi}$ serve different roles, as ϕ encodes each input location, while $\boldsymbol{\psi}$ defines the initial query values.

Embedding (§4). The frozen Random Fourier Features map (Rahimi and Recht, 2007) referenced as ϕ in Algorithm 1 is

$$\begin{aligned} \phi(\mathbf{x}) &= \big[\cos(2\pi B\mathbf{x}),\ \sin(2\pi B\mathbf{x})\big], \\ B \in \mathbb{R}^{F_\phi \times 2}, &\qquad B_{jk} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\phi^2), \\ \sigma_\phi &= 0.25/d_{\min}, \end{aligned} \tag{4}$$

with $d_{\min} = 0.5/\sqrt{K_{\max}}$ and F_ϕ the number of coordinate-embedding frequencies. This is separate from the source feature dimension $F = 256$ below.

Random-function source distribution (§4). The source of each draw is $Y_{0,i} = \boldsymbol{\psi}(\mathbf{x}_{Q,i})^\top \mathbf{z}$ with $\mathbf{z} \sim \mathcal{N}(0, I_F)$, $F = 256$, and

$$\begin{aligned} \boldsymbol{\psi}(\mathbf{x}) &= \sqrt{\tfrac{2}{F}}\left[\cos(\boldsymbol{\nu}_1^\top \mathbf{x} + c_1), \ldots, \cos(\boldsymbol{\nu}_F^\top \mathbf{x} + c_F)\right]^\top, \\ \boldsymbol{\nu}_j &\overset{\text{iid}}{\sim} \mathcal{N}(0, K_{\max} I_2), \qquad c_j \overset{\text{iid}}{\sim} \mathcal{U}(0, 2\pi), \end{aligned} \tag{5}$$

fixed at initialisation. $\mathbf{Y}_0$ is then an approximate draw from a unit-variance Gaussian process with RBF kernel and lengthscale $1/\sqrt{K_{\max}}$, and the same $\mathbf{z}$ conditions every block through adaLN.

Routing settings. Let $\mathcal{H} : [0,1]^2 \to \mathbb{N}$ be the index function of a 16th-order curve, which quantises each point to one of 2^{32} lattice cells. Let $\mathbf{s}_i \in \mathbb{R}^2$ be the spatial coordinate of the ith point in the concatenation $[\mathbf{X}_C \,\|\, \mathbf{X}_Q]$. MFNP applies the curve to the shifted coordinates $\frac{1}{2}\mathbf{s}_i + \mathbf{o}$, with a fresh offset $\mathbf{o} \sim \mathcal{U}[0, \frac{1}{2})^2$ for every training episode. At inference, the N_{samples} draws are split evenly across 8 offsets and pooled. Distances $\delta_{d,i}$ use the unshifted coordinates. Each direction d caches the distance $\delta_{d,i}$ from point $\pi_d(i)$ to the nearest preceding context point in that order, scaled by $\sqrt{N}$. The sort and distance cache depend only on $(\mathbf{X}_C, \mathbf{X}_Q)$ and the routing offset $\mathbf{o}$.

For one-dimensional inputs, points are sorted by their scalar coordinate and scanned in ascending and descending order, so the ensemble has two distinct directions. The four scan slots repeat these two orderings, as described in Appendix G. Algorithm 1 shows the two-dimensional routing.

Scan implementation. Context tokens write to the recurrent state. Query tokens read the state at their own positions but do not write to it. The distance vector δ_d is concatenated to the input of each block's Δt gate, whose softplus output is clipped to $[10^{-3}, 10]$. The gated scan output is RMS-normalised before the output projection. We use the JAX/XLA native implementation of Santoni (2026).

Sampling and clipping. Algorithm 3 generates one joint draw. The latent $\mathbf{z}$ and routing offset $\mathbf{o}$ remain fixed throughout its ODE integration. A one-step Euler call is the deployment sampler. The clipping bound $y_{\max}$ is the support limit of the per-task quantile-transformed targets ($y_{\max} = 6$ on synthetic tasks and on NOAA after the train-window quantile transform of §5.1); clipping is inactive on every reported run and is retained for numerical safety only.

Table 3: Best per-cell hyperparameters on the 1D synthetic Gaussian tasks (K=128). **Architecture.** d_{model} hidden width; $d_{\text{state/lat}}$ Mamba state / NP latent width; L blocks; h / F_ϕ transformer heads or MFNP RFF frequencies; d_{ff} transformer FFN width. **Training.** AdamW lr η, batch $|B|$, optimiser steps (CFM teacher + reflow for MFNP). **Inference.** T_{NFE} integration / diffusion steps; g_{scale} NDP guidance scale (jointly tuned with architecture in BO on synthetic tasks, and set by a post-hoc grid sweep on a fixed checkpoint on NOAA, selected to minimise validation MSE). Dashes mark N/A. † our method.

	Architecture					**Training**			**Inference**	
Model	d_{model}	$d_{\text{state/lat}}$	L	h / F_ϕ	d_{ff}	η	$\lvert B\rvert$	**steps**	T_{NFE}	g_{scale}
1D Gaussian – RBF (K=128)										
Factorised predictives										
CNP	256	–	–	–	–	2.6×10^{-3}	64	100k	–	–
NP	256	256	–	–	–	2.2×10^{-3}	64	100k	–	–
ANP	256	128	–	–	–	3.0×10^{-4}	64	100k	–	–
CANP	64	–	–	–	–	1.9×10^{-3}	64	100k	–	–
Joint predictives, quadratic-time										
TNPA	128	–	6	4	128	7.2×10^{-4}	32	100k	–	–
NDP	128	–	6	8	128	8.4×10^{-4}	16	100k	100	400
FNP	256	–	8	4	256	2.7×10^{-4}	64	100k	100	–
Joint predictives, linear-time										
MFNP†	384	32	3	128	–	8.4×10^{-4}	128	80k + 20k	1	–
1D Gaussian – Matérn-3/2 (K=128)										
Factorised predictives										
CNP	256	–	–	–	–	3.1×10^{-3}	64	100k	–	–
NP	128	256	–	–	–	2.9×10^{-3}	64	100k	–	–
ANP	64	128	–	–	–	1.2×10^{-3}	32	100k	–	–
CANP	256	–	–	–	–	7.1×10^{-4}	16	100k	–	–
Joint predictives, quadratic-time										
TNPA	64	–	6	8	512	1.6×10^{-4}	32	100k	–	–
NDP	256	–	8	4	128	3.4×10^{-4}	64	100k	100	900
FNP	64	–	8	8	256	5.3×10^{-4}	64	100k	100	–
Joint predictives, linear-time										
MFNP†	128	64	8	192	–	9.9×10^{-4}	128	80k + 20k	1	–
1D Gaussian – Matérn-5/2 (K=128)										
Factorised predictives										
CNP	256	–	–	–	–	1.1×10^{-3}	64	100k	–	–
NP	128	128	–	–	–	2.5×10^{-3}	64	100k	–	–
ANP	128	256	–	–	–	1.4×10^{-3}	64	100k	–	–
CANP	64	–	–	–	–	2.1×10^{-3}	64	100k	–	–
Joint predictives, quadratic-time										
TNPA	256	–	8	8	512	1.4×10^{-4}	16	100k	–	–
NDP	256	–	8	8	512	1.1×10^{-4}	32	100k	100	2,200
FNP	128	–	4	4	256	2.6×10^{-4}	16	100k	100	–
Joint predictives, linear-time										
MFNP†	128	16	3	192	–	1.6×10^{-3}	128	80k + 20k	1	–

Table 4: Best per-cell hyperparameters on the 2D synthetic Gaussian Matérn-$3/2$ task (K=256). Column conventions as in Table 3; factorised baselines are not reported on this task. † our method.

	Architecture					Training			Inference	
Model	d_{model}	$d_{\text{state/lat}}$	L	h / F_ϕ	d_{ff}	η	$\lvert B\rvert$	**steps**	T_{NFE}	g_{scale}
2D Gaussian – Matérn-$3/2$ (K=256)										
Joint predictives, quadratic-time										
TNPA	128	–	6	4	512	4.6×10^{-4}	64	100k	–	–
NDP	256	–	4	4	256	4.3×10^{-4}	32	100k	100	6,500
FNP	64	–	6	8	256	7.4×10^{-4}	32	100k	100	–
Joint predictives, linear-time										
MFNP†	256	128	6	384	–	1.1×10^{-3}	64	80k + 20k	1	–

Algorithm 1 MambaFlowNP forward pass (detailed).

Require: Context $\mathbf{X}_C \in \mathbb{R}^{N\times D}$, $\mathbf{Y}_C \in \mathbb{R}^N$.
Require: Query coordinates $\mathbf{X}_Q \in \mathbb{R}^{M\times D}$, noisy query slice $\mathbf{Y}_t \in \mathbb{R}^M$, flow time $t \in [0,1]$, draw latent $\mathbf{z} \in \mathbb{R}^F$, routing offset $\mathbf{o} \in [0, \frac{1}{2})^2$.
Require: Frozen RFF projection ϕ; optional routing cache.
Let $K = N + M$. Concatenate coordinates $\mathbf{S} = [\mathbf{X}_C \,\|\, \mathbf{X}_Q] \in \mathbb{R}^{K\times D}$.
Phase 1: Tokenisation & Conditioning
$\mathbf{e}_t \leftarrow \text{MLP}_{\text{time}}(\text{SinusoidalEmbedding}(t)) + \text{MLP}_{\text{lat}}(\mathbf{z})$
for $i = 1$ **to** K **do**
  if $i \le N$ **then**
    $\mathbf{E}_i \leftarrow W_{\text{ctx}}\,[\phi(\mathbf{s}_i), Y_{C,i}]$
  else
    $\mathbf{E}_i \leftarrow W_{\text{qry}}\,[\phi(\mathbf{s}_i), Y_{t,i-N}]$
  end if
end for
Phase 2: Multi-directional routing (pre-computed, cached)
Reuse the cached quantities if supplied; otherwise compute the following.
$\tilde{\mathbf{S}} \leftarrow \frac{1}{2}\mathbf{S} + \mathbf{o}$ ▷ jittered scan grid
$\pi_1 \leftarrow \text{argsort}\big(\mathcal{H}(\tilde{\mathbf{S}}_{:,1}, \tilde{\mathbf{S}}_{:,2})\big)$
$\pi_2 \leftarrow \text{argsort}\big(\mathcal{H}(\tilde{\mathbf{S}}_{:,2}, \tilde{\mathbf{S}}_{:,1})\big)$
Define ensemble $\Pi = \{\pi_1, \pi_1^{\text{rev}}, \pi_2, \pi_2^{\text{rev}}\}$.
Compute the inverse permutation $\pi_d^{-1} \leftarrow \text{argsort}(\pi_d)$ for each $d \in \Pi$.
$\delta_{d,i} \leftarrow \sqrt{N}\,\|\mathbf{s}_{\pi_d(i)} - \mathbf{s}_{\pi_d(j)}\|$ for each d, i, with $j < i$ the nearest preceding context position in π_d ($\delta_{d,i} = 0$ if none).
Phase 3: Mamba-2 with distance-aware adaLN-Zero
Apply permutations: $\mathbf{E}_d \leftarrow \mathbf{E}[\pi_d]$ for each d.
for $\ell = 1$ **to** L **do**
  $(\gamma_\ell, \beta_\ell, \alpha_\ell) \leftarrow W^{(\ell)}_{\text{adaln}}\,\text{SiLU}(\mathbf{e}_t)$ ▷ zero-init scale, shift, gate
  $\mathbf{E}_d \leftarrow \mathbf{E}_d + \alpha_\ell \odot \text{Mamba2}\big(\text{LN}(\mathbf{E}_d) \odot (1+\gamma_\ell) + \beta_\ell,\ \delta_d;\ \theta\big)$ for each d.
end for
Phase 4: Fusion & Readout
$\mathbf{H}_d \leftarrow \mathbf{E}_d[\pi_d^{-1}]$ for each d. ▷ inverse permutation
$\mathbf{H} \leftarrow W_{\text{fuse}}[\mathbf{H}_{\pi_1} \,\|\, \mathbf{H}_{\pi_1^{\text{rev}}} \,\|\, \mathbf{H}_{\pi_2} \,\|\, \mathbf{H}_{\pi_2^{\text{rev}}}] + b_{\text{fuse}}$
$\hat{\mathbf{v}} \leftarrow W_{\text{out}}\big(\text{adaLN}(\text{LN}(\mathbf{H}[N+1:K]), \mathbf{e}_t)\big) + b_{\text{out}}$ ▷ zero-init output
return $\hat{\mathbf{v}}$

C Sort scaling

Table 6 reports wall-clock for the Hilbert-key `argsort` call against K, alongside a same-shape no-op identity baseline isolating fixed per-call cost (kernel launch, dispatch) from K-dependent work, and the MFNP 1-NFE Euler forward-pass wall-clock at the same K on 2D Matérn-$3/2$ (batch= 1, N_{samples}=1). Radix sort runs in $O(wK)$ time on w-bit keys, and at curve order h=16 the keys are 32-bit unsigned integers with each 2D point quantised to one of 2^{32} lattice cells (Sagan, 1994;

Table 5: Best per-cell hyperparameters on the three NOAA tiers (Utah $K{=}222$; California $K{=}486$; continental US $K{=}5{,}580$). Column conventions as in Table 3. TNPA exceeds the per-cell compute budget on the US tier (§5.3) and is omitted there. MFNP uses 4 Mamba-2 heads and a scan chunk length of 64. † our method.

	Architecture					**Training**			**Inference**	
Model	d_{model}	$d_{\text{state/lat}}$	L	h / F_ϕ	d_{ff}	η	$\lvert B\rvert$	**steps**	T_{NFE}	g_{scale}
NOAA Utah ($K{=}222$)										
Factorised predictives										
CNP	256	–	–	–	–	$1.9{\times}10^{-3}$	8	100k	–	–
NP	256	64	–	–	–	$3.7{\times}10^{-3}$	8	100k	–	–
ANP	128	64	–	–	–	$1.2{\times}10^{-3}$	4	100k	–	–
CANP	256	–	–	–	–	$9.4{\times}10^{-4}$	4	100k	–	–
Joint predictives, quadratic-time										
TNPA	128	–	8	8	512	$2.9{\times}10^{-4}$	4	100k	–	–
NDP	256	–	3	4	512	$2.0{\times}10^{-4}$	8	100k	100	25,000
FNP	256	–	4	8	768	$1.3{\times}10^{-4}$	4	100k	100	–
Joint predictives, linear-time										
MFNP†	128	16	6	256	–	$1.0{\times}10^{-3}$	16	80k + 20k	1	–
NOAA California ($K{=}486$)										
Factorised predictives										
CNP	256	–	–	–	–	$1.4{\times}10^{-3}$	8	100k	–	–
NP	256	64	–	–	–	$2.0{\times}10^{-3}$	8	100k	–	–
ANP	256	128	–	–	–	$1.2{\times}10^{-3}$	4	100k	–	–
CANP	256	–	–	–	–	$9.3{\times}10^{-4}$	4	100k	–	–
Joint predictives, quadratic-time										
TNPA	128	–	10	4	512	$1.8{\times}10^{-4}$	4	100k	–	–
NDP	256	–	4	8	256	$1.9{\times}10^{-4}$	8	100k	100	25,000
FNP	128	–	4	8	768	$2.1{\times}10^{-4}$	2	100k	100	–
Joint predictives, linear-time										
MFNP†	256	16	6	192	–	$1.5{\times}10^{-3}$	16	80k + 20k	1	–
NOAA US ($K{=}5{,}580$)										
Factorised predictives										
CNP	768	–	–	–	–	$1.3{\times}10^{-3}$	32	100k	–	–
NP	512	384	–	–	–	$9.4{\times}10^{-4}$	32	100k	–	–
ANP	256	256	–	–	–	$1.4{\times}10^{-3}$	2	100k	–	–
CANP	256	–	–	–	–	$5.3{\times}10^{-4}$	4	100k	–	–
Joint predictives, quadratic-time										
NDP	384	–	3	8	512	$1.4{\times}10^{-4}$	16	100k	100	15,000
FNP	256	–	4	4	512	$3.5{\times}10^{-4}$	4	100k	100	–
Joint predictives, linear-time										
MFNP†	256	16	6	384	–	$1.0{\times}10^{-3}$	8	80k + 20k	1	–

Algorithm 2 MambaFlowNP CFM training step.

Require: Episode sampler returning $(\mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q, \mathbf{Y}_Q)$; parameters θ.

Sample episode and $t \sim \mathcal{U}(0,1)$.
Draw $\mathbf{z} \sim \mathcal{N}(0, I_F)$, $\mathbf{o} \sim \mathcal{U}[0, \frac{1}{2})^2$, and set $Y_{0,i} \leftarrow \boldsymbol{\psi}(\mathbf{x}_{Q,i})^\top \mathbf{z}$.
$\mathbf{Y}_t \leftarrow (1-t)\mathbf{Y}_0 + t\mathbf{Y}_Q$; $\quad \mathbf{u}^\star \leftarrow \mathbf{Y}_Q - \mathbf{Y}_0$.
$\hat{\mathbf{v}} \leftarrow \text{MFNP}_\theta(\mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q, \mathbf{Y}_t, t, \mathbf{z}, \mathbf{o})$ via Algorithm 1.
$\mathcal{L} \leftarrow \|\hat{\mathbf{v}} - \mathbf{u}^\star\|^2$. Update θ by gradient step on $\mathcal{L}$.

Algorithm 3 Posterior sampling (detailed; Heun / Euler).

Require: Context $\mathbf{X}_C, \mathbf{Y}_C$, queries $\mathbf{X}_Q$, trained MFNP (Algorithm 1), steps N_{steps}, solver $\in \{\text{Euler}, \text{Heun}\}$.

Draw the latent $\mathbf{z} \sim \mathcal{N}(0, I_F)$ and the routing offset $\mathbf{o}$ of this draw (one of 8 offsets shared by $N_{\text{samples}}/8$ draws).
Pre-compute routing cache from $(\mathbf{X}_C, \mathbf{X}_Q, \mathbf{o})$: permutations $\{\pi_d : d \in \Pi\}$ with $\Pi = \{\pi_1, \pi_1^{\text{rev}}, \pi_2, \pi_2^{\text{rev}}\}$, inverse permutations $\{\pi_d^{-1}\}$, and distances $\{\delta_d\}$.
Set $Y_{0,i} \leftarrow \boldsymbol{\psi}(\mathbf{x}_{Q,i})^\top \mathbf{z}$, the random-function source of §4.
$\Delta t \leftarrow 1/N_{\text{steps}}$.
for $k = 0$ **to** $N_{\text{steps}} - 1$ **do**
    $t_1 \leftarrow k\Delta t$; $\quad t_2 \leftarrow (k+1)\Delta t$.
    $\mathbf{v}_1 \leftarrow \text{MFNP}(\mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q, \mathbf{Y}_{t_1}, t_1, \mathbf{z}, \mathbf{o}; \text{cache})$
    if Heun **then**
        $\tilde{\mathbf{Y}} \leftarrow \text{clip}_{[-y_{\max}, y_{\max}]}\big(\mathbf{Y}_{t_1} + \Delta t \cdot \mathbf{v}_1\big)$
        $\mathbf{v}_2 \leftarrow \text{MFNP}(\mathbf{X}_C, \mathbf{Y}_C, \mathbf{X}_Q, \tilde{\mathbf{Y}}, t_2, \mathbf{z}, \mathbf{o}; \text{cache})$
        $\mathbf{Y}_{t_2} \leftarrow \text{clip}_{[-y_{\max}, y_{\max}]}\big(\mathbf{Y}_{t_1} + \frac{\Delta t}{2}(\mathbf{v}_1 + \mathbf{v}_2)\big)$
    else
        $\mathbf{Y}_{t_2} \leftarrow \mathbf{Y}_{t_1} + \Delta t \cdot \mathbf{v}_1$
    end if
end for
return $\mathbf{Y}_1$ ▷ $\hat{\mathbf{Y}}_1$, a posterior sample

Hamilton and Rau-Chaplin, 2008), so w=32 is fixed by the curve order rather than by K; the sort therefore has $O(K)$ complexity within any regime in which Hilbert keys remain distinct. Once K grows large enough to force lattice collisions the sort must break ties arbitrarily and the locality argument of Moon et al. (2001) no longer applies. Above $K \gtrsim 65{,}000$ the compiled program lowers `argsort` to CUB's radix-sort kernel (Merrill, 2016); below this threshold an inline path performs a fixed number of comparisons per element. Both regimes are $O(K)$. All measurements are on a single NVIDIA L40S in one Python process with one JAX trace cache, 10 warm-ups followed by 100 timed reps, median statistic.

The identity baseline is constant at $\sim$40 μs across four orders of magnitude in K. Kernel-only cost grows from 22.1 μs at K=2,048 to 83.6 μs at K=10^6, a 3.8$\times$ increase for a 512$\times$ increase in K. Above $K \approx 5\times10^5$ the radix kernel saturates memory bandwidth and per-element cost flattens. The sort fraction of MFNP inference plateaus in the 1.0–1.9% range for $K \leq 8{,}192$ where both kernels are dispatch-bound, then decreases monotonically with K to 0.014% at K=10^6 as the linear-time scan dominates.

D Full matched-regime results

Table 7 covers the full 1D sweep at K=128 across RBF, Matérn-$^3/_2$, and Matérn-$^5/_2$; the 2D sweep is in the main-body Table 1.

E Scaling

Table 8 reports the end-to-end sampling latency plotted in Figure 3 for each joint-predictive NP against total sequence length K on 2D Gaussian Matérn-$^3/_2$.

Table 6: Sort scaling and its share of MFNP inference (NVIDIA L40S, JAX 0.9.0.1, CUDA 12). Identity is a same-shape no-op baseline; kernel is sort total minus identity. MFNP wall-clock is per-forward at the same K on 2D Matérn-$3/2$ (1-NFE Euler, batch$=1$, $N_{\text{samples}}=1$, headline checkpoint).

	Standalone sort			*Share of MFNP inference*	
K	**Identity (μs)**	**Sort (μs)**	**Kernel (μs)**	**MFNP total (ms)**	**Sort fraction**
128	41.8	44.4	2.6	3.41	1.30%
256	37.9	35.4	≈0	3.50	1.01%
512	41.3	45.3	4.0	3.61	1.26%
1,024	41.4	44.2	2.7	3.72	1.19%
2,048	35.9	58.0	22.1	3.87	1.50%
4,096	36.5	86.7	50.3	4.67	1.86%
8,192	35.9	86.1	50.2	7.27	1.18%
16,384	38.6	101.7	63.1	14.02	0.73%
32,768	36.6	102.4	65.8	31.76	0.32%
65,536	31.8	97.7	66.0	61.96	0.16%
131,072	40.9	101.4	60.5	120.23	0.084%
262,144	36.3	101.2	65.0	233.14	0.043%
524,288	40.7	115.2	74.5	460.38	0.025%
1,048,576	41.8	125.4	83.6	912.48	0.014%

Table 7: Full 1D matched-regime results at $K_{\max}=128$, all three kernels (RBF, Matérn-$3/2$, Matérn-$5/2$). Baselines are grouped by predictive class into *factorised* ($O(K)$ pooling or $O(NM)$ cross-attention with per-query Gaussian head), *joint / quadratic* ($O(K^2)$ self-attention), *joint / linear* ($O(K)$ Mamba-2 scan). MSE, CRPS (↓); C2ST (target 0.5); 95% coverage (target nominal); time = per-episode ms, MFNP and baselines on one NVIDIA L40S. Mean ± SE over 10 evaluation seeds; bold = best per column (within 1 SE). † our method.

	Predictive Quality		**Calibration Quality**		
Model	**MSE ↓**	**CRPS ↓**	**C2ST → 0.5**	**Cov-95 → 0.95**	**Time (ms) ↓**
1D Gaussian – RBF ($K=128$)					
Factorised predictives					
CNP	0.0285 ± 0.0008	0.0666 ± 0.0009	0.673 ± 0.020	0.941 ± 0.002	$\mathbf{0.40 \pm 0.02}$
NP	0.0388 ± 0.0010	0.0906 ± 0.0014	0.753 ± 0.029	0.474 ± 0.003	0.58 ± 0.00
ANP	0.0026 ± 0.0001	0.0255 ± 0.0002	0.691 ± 0.008	0.091 ± 0.001	1.31 ± 0.01
CANP	0.0047 ± 0.0002	0.0252 ± 0.0003	0.605 ± 0.005	0.880 ± 0.001	1.09 ± 0.01
Joint predictives, quadratic-time					
TNPA	0.0029 ± 0.0002	0.0197 ± 0.0002	$\mathbf{0.545 \pm 0.008}$	$\mathbf{0.956 \pm 0.001}$	442.9 ± 0.7
NDP	0.2461 ± 0.0047	0.2772 ± 0.0023	0.742 ± 0.026	0.966 ± 0.001	929.0 ± 6.5
FNP	$\mathbf{0.0018 \pm 0.0001}$	$\mathbf{0.0180 \pm 0.0001}$	$\mathbf{0.540 \pm 0.014}$	0.664 ± 0.002	$(1.516 \pm 0.002) \times 10^3$
Joint predictives, linear-time					
MFNP†	0.0034 ± 0.0002	0.0224 ± 0.0002	0.596 ± 0.006	0.863 ± 0.001	25.7 ± 0.2
1D Gaussian – Matérn-$3/2$ ($K=128$)					
Factorised predictives					
CNP	0.0556 ± 0.0011	0.1085 ± 0.0010	0.695 ± 0.014	0.931 ± 0.001	$\mathbf{0.47 \pm 0.02}$
NP	0.0466 ± 0.0007	0.1139 ± 0.0010	0.756 ± 0.019	0.505 ± 0.002	0.62 ± 0.03
ANP	0.0111 ± 0.0003	0.0504 ± 0.0005	0.643 ± 0.006	0.148 ± 0.001	1.12 ± 0.01
CANP	0.0156 ± 0.0003	0.0512 ± 0.0005	0.593 ± 0.009	0.856 ± 0.002	1.02 ± 0.01
Joint predictives, quadratic-time					
TNPA	$\mathbf{0.0110 \pm 0.0003}$	$\mathbf{0.0367 \pm 0.0004}$	$\mathbf{0.499 \pm 0.003}$	$\mathbf{0.946 \pm 0.001}$	735.7 ± 0.4
NDP	0.1720 ± 0.0028	0.2508 ± 0.0014	0.688 ± 0.018	0.996 ± 0.000	$(2.174 \pm 0.004) \times 10^3$
FNP	$\mathbf{0.0107 \pm 0.0002}$	0.0398 ± 0.0004	0.532 ± 0.011	0.752 ± 0.001	$(1.253 \pm 0.001) \times 10^3$
Joint predictives, linear-time					
MFNP†	0.0111 ± 0.0003	0.0398 ± 0.0004	0.537 ± 0.006	0.938 ± 0.001	29.5 ± 0.2
1D Gaussian – Matérn-$5/2$ ($K=128$)					
Factorised predictives					
CNP	0.0218 ± 0.0005	0.0595 ± 0.0006	0.657 ± 0.017	0.930 ± 0.001	$\mathbf{0.49 \pm 0.02}$
NP	0.0400 ± 0.0009	0.1070 ± 0.0013	0.815 ± 0.023	0.333 ± 0.002	0.53 ± 0.00
ANP	0.0064 ± 0.0002	0.0367 ± 0.0004	0.661 ± 0.004	0.131 ± 0.001	1.18 ± 0.01
CANP	0.0077 ± 0.0002	0.0320 ± 0.0003	0.577 ± 0.005	0.883 ± 0.001	1.05 ± 0.01
Joint predictives, quadratic-time					
TNPA	$\mathbf{0.0058 \pm 0.0002}$	$\mathbf{0.0269 \pm 0.0003}$	$\mathbf{0.512 \pm 0.008}$	$\mathbf{0.953 \pm 0.001}$	$(1.601 \pm 0.002) \times 10^3$
NDP	0.0745 ± 0.0010	0.1441 ± 0.0009	0.737 ± 0.008	0.977 ± 0.001	$(2.779 \pm 0.005) \times 10^3$
FNP	$\mathbf{0.0057 \pm 0.0002}$	0.0279 ± 0.0003	0.531 ± 0.010	0.680 ± 0.002	534.2 ± 0.8
Joint predictives, linear-time					
MFNP†	0.0065 ± 0.0002	0.0303 ± 0.0003	0.541 ± 0.007	0.970 ± 0.001	8.4 ± 0.1

Table 8: End-to-end sampling latency (ms) for joint-predictive NPs vs. total sequence length K on 2D Gaussian Matérn-$3/2$ on one NVIDIA L40S, batch$=1$, $N_{\text{samples}}=1$ (one full posterior draw per pass, with 1-NFE Euler for MFNP and the published step count for each baseline). Median of 5 reps after 3 warm-ups; relative SE $< 5\%$ on every completing cell. Values $\geq 10^3$ in scientific notation, rounded to 2–3 sig figs. OOM = out of memory. Bold = best per column. † our method.

Model	16	256	4,096	16,384	65,536	1,048,576
Joint predictives, quadratic-time						
TNPA	10.7	151	1.2×10^5	7.4×10^6	OOM	OOM
NDP	315	317	1.5×10^3	OOM	OOM	OOM
FNP	124.0	122.6	1.07×10^4	OOM	OOM	OOM
Joint predictives, linear-time						
MFNP†	**3.51**	**3.63**	**4.82**	**14.74**	**61.88**	**913.86**

F Full NOAA results

Context-fraction sweep. Figure 5 sweeps the context fraction N/K across fifteen values from 1% to 99% on each NOAA tier, holding $N + M$ fixed at the full station count. Each fraction is averaged over $1{,}024$ random (date, station-partition) draws, so the curves report the population-level dependence of MFNP error on the size of the observed support rather than a single realisation. RMSE and CRPS fall steeply over the first $\sim 30\%$ of context (roughly $3\times$ on Utah, from 4.9 to 1.6°C RMSE, and $2\times$ on California and the US) and then flatten out, with the remaining context yielding a further 12–21% of RMSE reduction. The 50% split used for the headline tables (Table 9) sits in this plateau.

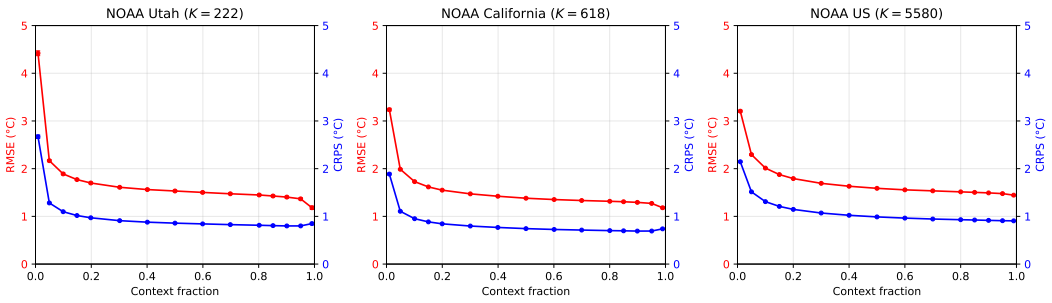

Figure 5: MFNP predictive accuracy as a function of context fraction N/K on the three NOAA tiers (Utah K=222; California K=486; continental US K=5,580), averaged over $1{,}024$ random (date, station-partition) draws per fraction. Red (left axis) shows RMSE and blue (right axis) shows CRPS, both in °C.

Table 9 covers the full NOAA sweep, adding the factorised baselines (CNP, NP, ANP, CANP) at each of the three tiers.

Gaussian-process references. Table 10 reports three GPs on the NOAA test sets. Each uses an ARD Matérn-$3/2$ kernel on the station coordinates and fits its hyperparameters to the context of each episode. The exact GP solves the full $N \times N$ system, SGPR uses inducing points (Titsias, 2009), and the nearest-neighbour GP conditions each point on its 20 nearest neighbours (Vecchia, 1988; Datta et al., 2016). MFNP is trained across dates on the same station network and the GPs use only the current episode.

Per-tier Pareto frontiers. Figures 6 and 7 reproduce the RMSE–latency analysis of Figure 2 on the Utah and California tiers; marker, error-bar, and Pareto-staircase conventions are identical.

Qualitative posteriors. Figures 8–9 show MFNP posterior predictions on a single held-out date for Utah and California at a $25\%/75\%$ context/query split, in the layout of Figure 4. They show the predicted field and the spatial structure of its error on a single realisation. The population-averaged dependence on context fraction is reported separately in Figure 5.

Table 9: Full NOAA temperature interpolation on Utah (K=222), California (K=486), and the continental US (K=5,580); baselines grouped by predictive class as in Table 7. MSE (°C)^2, RMSE, CRPS °C (↓); Cov-95 (target 0.95); C2ST omitted (no tractable reference posterior on real fields); time = per-episode ms on one NVIDIA L40S. Mean ± SE over 10 evaluation seeds, denormalised scale; bold = best per column within tier (within 1 SE). † our method.

	Predictive Quality			Calibration	
Model	**MSE (°C)^2 ↓**	**RMSE (°C) ↓**	**CRPS (°C) ↓**	**Cov-95 → 0.95**	**Time (ms) ↓**
NOAA Utah (K=222)					
Factorised predictives					
CNP	4.230 ± 0.010	2.057 ± 0.002	1.075 ± 0.001	0.930 ± 0.000	$\mathbf{0.46 \pm 0.02}$
NP	6.196 ± 0.017	2.489 ± 0.003	1.551 ± 0.002	0.481 ± 0.000	0.75 ± 0.00
ANP	3.653 ± 0.010	1.911 ± 0.003	1.246 ± 0.002	0.268 ± 0.000	1.34 ± 0.06
CANP	3.783 ± 0.011	1.945 ± 0.003	1.005 ± 0.001	0.900 ± 0.000	0.99 ± 0.01
Joint predictives, quadratic-time					
TNPA	2.524 ± 0.007	1.589 ± 0.002	0.783 ± 0.001	0.846 ± 0.000	$(4.923 \pm 0.004) \times 10^3$
NDP	4.665 ± 0.012	2.160 ± 0.003	1.221 ± 0.002	0.742 ± 0.000	$(2.414 \pm 0.004) \times 10^3$
FNP	2.712 ± 0.007	1.647 ± 0.002	0.853 ± 0.001	0.730 ± 0.000	$(3.071 \pm 0.003) \times 10^3$
Joint predictives, linear-time					
MFNP†	$\mathbf{2.464 \pm 0.016}$	$\mathbf{1.570 \pm 0.005}$	$\mathbf{0.767 \pm 0.002}$	$\mathbf{0.940 \pm 0.000}$	43.3 ± 0.4
NOAA California (K=486)					
Factorised predictives					
CNP	3.575 ± 0.015	1.891 ± 0.004	1.013 ± 0.002	0.938 ± 0.000	$\mathbf{0.48 \pm 0.02}$
NP	4.365 ± 0.021	2.089 ± 0.005	1.323 ± 0.003	0.496 ± 0.001	1.24 ± 0.00
ANP	3.340 ± 0.012	1.828 ± 0.003	1.213 ± 0.002	0.331 ± 0.000	2.35 ± 0.01
CANP	3.431 ± 0.015	1.852 ± 0.004	0.976 ± 0.002	0.892 ± 0.000	1.08 ± 0.05
Joint predictives, quadratic-time					
TNPA	1.898 ± 0.005	1.378 ± 0.002	0.701 ± 0.001	0.938 ± 0.000	$(3.004 \pm 0.002) \times 10^4$
NDP	3.071 ± 0.008	1.753 ± 0.002	1.064 ± 0.001	0.590 ± 0.000	$(9.154 \pm 0.015) \times 10^3$
FNP	2.395 ± 0.008	1.548 ± 0.003	0.839 ± 0.001	0.785 ± 0.001	$(8.844 \pm 0.004) \times 10^3$
Joint predictives, linear-time					
MFNP†	$\mathbf{1.808 \pm 0.006}$	$\mathbf{1.344 \pm 0.002}$	$\mathbf{0.681 \pm 0.001}$	$\mathbf{0.945 \pm 0.000}$	67.9 ± 0.5
NOAA US (K=5,580)					
Factorised predictives					
CNP	7.764 ± 0.032	2.786 ± 0.006	1.501 ± 0.003	0.911 ± 0.000	$\mathbf{2.35 \pm 0.69}$
NP	11.039 ± 0.034	3.322 ± 0.005	2.349 ± 0.003	0.229 ± 0.000	33.5 ± 0.0
ANP	9.312 ± 0.025	3.052 ± 0.004	1.999 ± 0.002	0.272 ± 0.000	25.9 ± 0.0
CANP	5.585 ± 0.027	2.363 ± 0.006	1.196 ± 0.002	0.905 ± 0.000	8.45 ± 0.01
Joint predictives, quadratic-time					
TNPA	T/O	T/O	T/O	T/O	T/O
NDP	3.528 ± 0.087	1.878 ± 0.023	1.194 ± 0.013	0.322 ± 0.005	$(7.075 \pm 0.020) \times 10^5$
FNP	2.872 ± 0.064	1.695 ± 0.019	0.864 ± 0.009	0.786 ± 0.001	$(8.831 \pm 0.002) \times 10^5$
Joint predictives, linear-time					
MFNP†	$\mathbf{2.554 \pm 0.010}$	$\mathbf{1.598 \pm 0.003}$	$\mathbf{0.774 \pm 0.001}$	$\mathbf{0.955 \pm 0.001}$	$(1.126 \pm 0.000) \times 10^3$

Table 10: Gaussian-process references on NOAA. MSE (°C)^2, RMSE, CRPS °C (↓); Cov-95 (target 0.95). Mean ± SE over evaluation seeds (10 for MFNP and the nearest-neighbour GP; 10, 10 and 8 for the exact GP; 4, 4 and 3 for SGPR). SGPR uses 64 inducing points on Utah and California and 512 on the US. † our method.

Model	**MSE (°C)^2 ↓**	**RMSE (°C) ↓**	**CRPS (°C) ↓**	**Cov-95 → 0.95**
NOAA Utah (K=222)				
Exact GP	16.240 ± 0.034	3.936 ± 0.004	2.246 ± 0.002	0.916 ± 0.000
SGPR	16.501 ± 0.073	3.965 ± 0.007	2.277 ± 0.004	0.936 ± 0.001
Nearest-neighbour GP	16.255 ± 0.065	3.937 ± 0.007	2.239 ± 0.004	0.916 ± 0.000
MFNP†	$\mathbf{2.464 \pm 0.016}$	$\mathbf{1.570 \pm 0.005}$	$\mathbf{0.767 \pm 0.002}$	$\mathbf{0.940 \pm 0.000}$
NOAA California (K=486)				
Exact GP	11.813 ± 0.049	3.399 ± 0.007	1.896 ± 0.004	0.924 ± 0.000
SGPR	12.330 ± 0.057	3.476 ± 0.008	1.948 ± 0.004	0.936 ± 0.000
Nearest-neighbour GP	12.119 ± 0.050	3.441 ± 0.007	1.914 ± 0.004	0.921 ± 0.000
MFNP†	$\mathbf{1.808 \pm 0.006}$	$\mathbf{1.344 \pm 0.002}$	$\mathbf{0.681 \pm 0.001}$	$\mathbf{0.945 \pm 0.000}$
NOAA US (K=5,580)				
Exact GP	9.440 ± 0.025	3.050 ± 0.004	1.690 ± 0.002	0.929 ± 0.000
SGPR	9.757 ± 0.009	3.101 ± 0.001	1.724 ± 0.001	0.932 ± 0.000
Nearest-neighbour GP	9.498 ± 0.032	3.059 ± 0.005	1.692 ± 0.003	0.925 ± 0.000
MFNP†	$\mathbf{2.554 \pm 0.010}$	$\mathbf{1.598 \pm 0.003}$	$\mathbf{0.774 \pm 0.001}$	$\mathbf{0.955 \pm 0.001}$

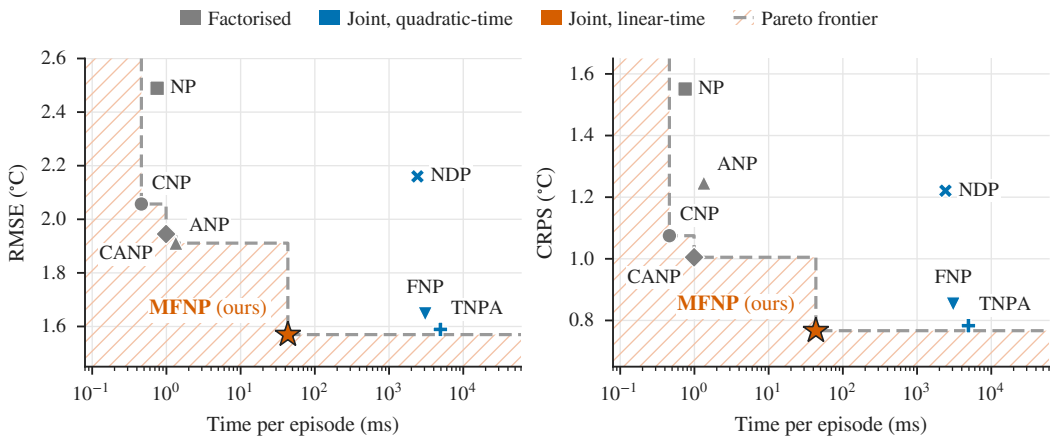

Figure 6: Per-episode latency (x, log) versus RMSE (left panel) and CRPS (right panel) on NOAA Utah (K=222). Conventions as in Figure 2.

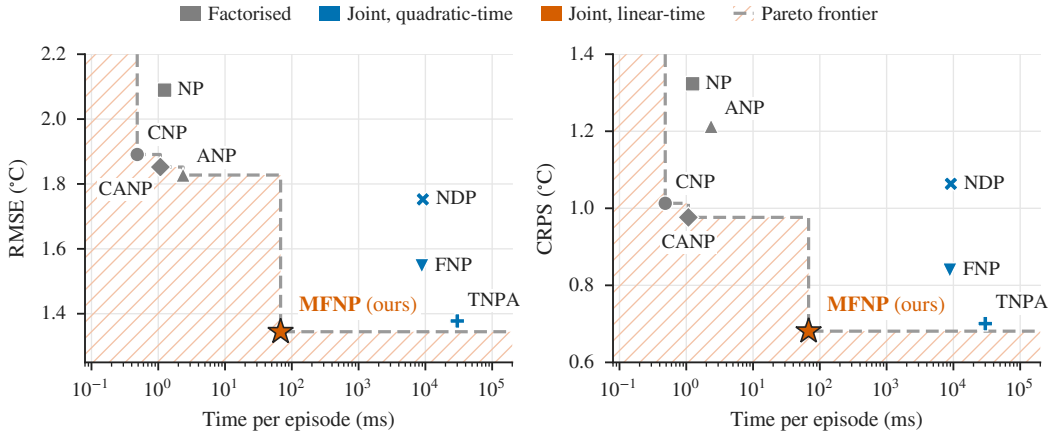

Figure 7: Per-episode latency (x, log) versus RMSE (left panel) and CRPS (right panel) on NOAA California (K=486). Conventions as in Figure 2.

G Architectural ablations

Table 11 reports the architectural ablations. Each row retrains MFNP from scratch under the full hyperparameter configuration and uses the same evaluation protocol as the main tables (N_{samples}=128, batch = 1, 10 evaluation seeds). The full row matches the headline 2D synthetic and NOAA configurations.

Variants. *Scan routing. Single direction* uses only π_1; *two directions* uses π_1 and its reversal (no axis-transposed curve). In 1D there is no transposed curve, so the full model fills its four direction slots with the ascending and descending orders twice, and *two directions* uses each order once. *Point ordering. Random order* uses a random permutation per episode, isolating the effect of locality in the serialisation. *State flow. Single hidden state* runs a unidirectional SSM over the context, takes its final hidden state as a single summary, and initialises a separate decoder over queries from that vector.

Space-filling-curve orderings. Table 12 retrains MFNP on 2D Matérn-$3/2$ with four point orderings, one training seed each. Morton and lexicographic orderings give 4% and 9% lower MSE than the Hilbert curve, lexicographic ordering raises CRPS by 16%, and random ordering is 18% worse. We use the Hilbert curve because it keeps consecutive points adjacent in the plane, as in prior Hilbert

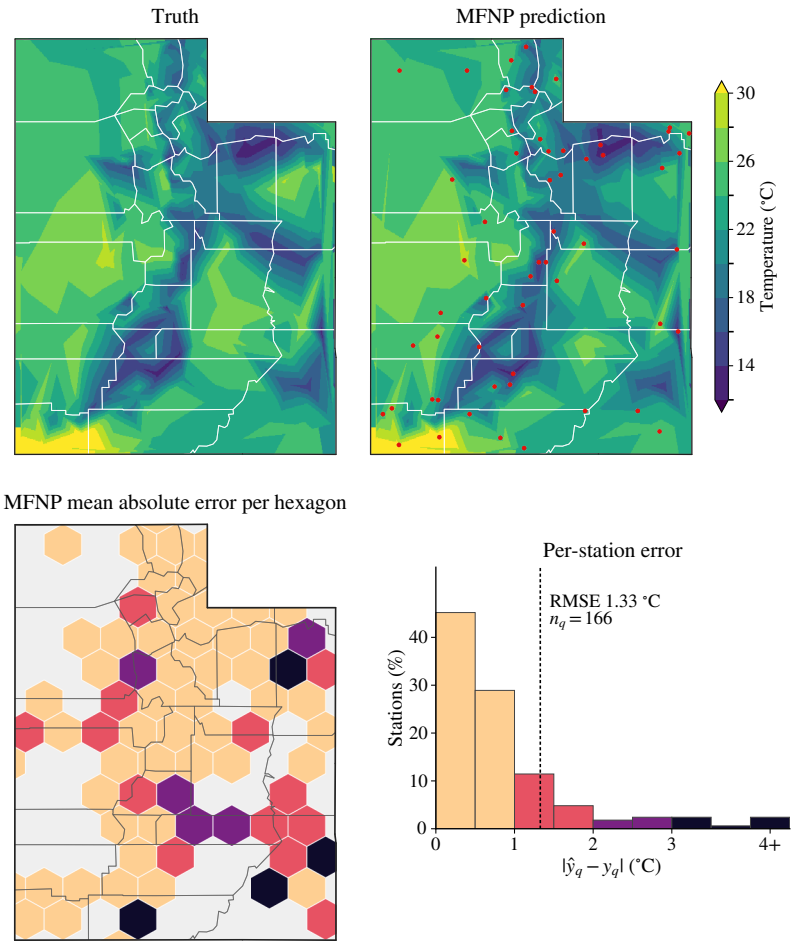

Figure 8: NOAA Utah (K=222) posterior on a held-out date, 25%/75% context/query split. Panels as in Figure 4, with ∼50 km hexagons.

serialisation of irregular point sets (Liang et al., 2024), while the Morton curve jumps at each quadrant boundary.

H Solver and distillation

On NOAA, the teacher generates the reflow couplings with 10 Heun steps. Table 13 sweeps Euler and Heun integrators at $\{1, 15, 30\}$ steps on the reflow-distilled MambaFlowNP student across all four headline tasks. On 1D Matérn-$3/2$, 1-step Euler matches the multi-step configurations within one SE on MSE and CRPS, and coverage moves by at most 0.005 with more steps. On 2D Matérn-$3/2$, 1-step Euler gives an MSE of 0.0167, a CRPS of 0.0537 and a coverage of 0.971, against 0.0166, 0.0537 and 0.975 at 15 steps. On both NOAA tiers 1-step Euler gives the lowest MSE, the lowest CRPS (0.767 against 0.808 at 30 steps on Utah, 0.681 against 0.748 at 15 steps on California) and the coverage closest to nominal (0.940 and 0.945). Heun-1 is within one SE of Euler-1 on MSE on 1D (0.0112 against 0.0111) and on 2D (0.0167 for both). On the NOAA tiers it inflates MSE by 1.6–1.9×. The Euler 1-step row is therefore the deployment configuration used in every other table.

Architecture vs. distillation at matched NFE. The headline scaling figure (Figure 3) leaves each baseline at its native solver length, so MFNP's advantage there conflates reflow distillation with the linear-K encoder. To isolate the two, we re-time NDP, FNP and MFNP at 1-NFE on the same 2D Matérn-$3/2$ sweep with all three resized to ∼600k parameters and random initialisation; forward-pass

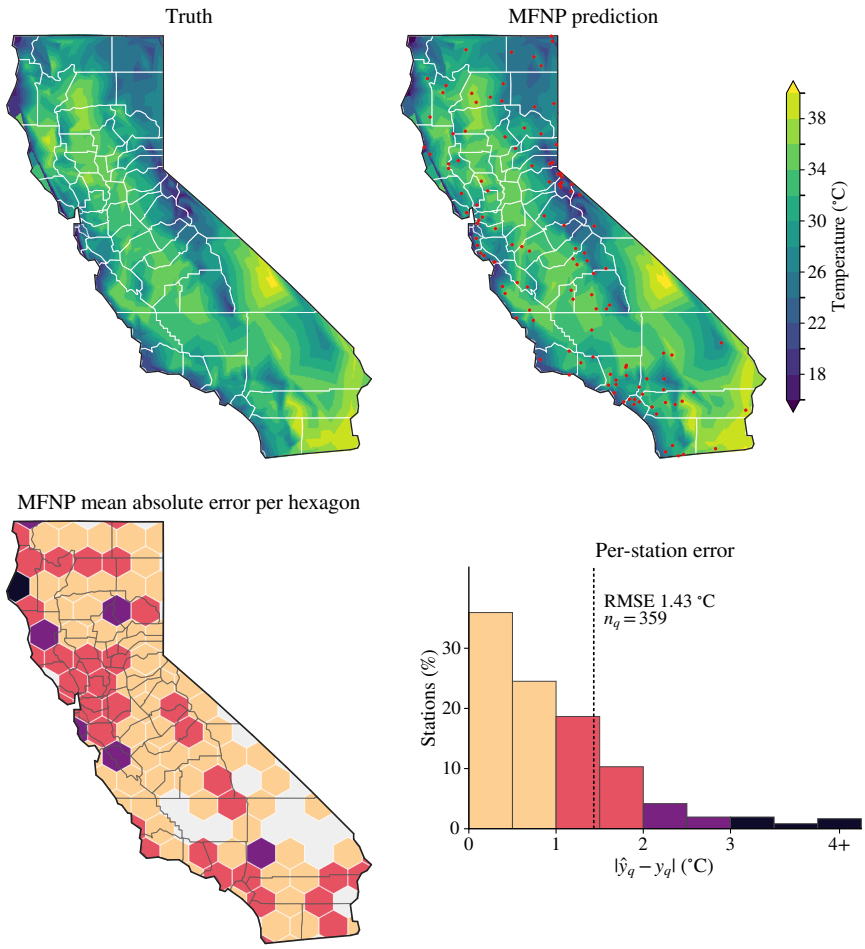

Figure 9: NOAA California (K=486) posterior on a held-out date, 25%/75% context/query split. Panels as in Figure 4, with ∼70 km hexagons. Five coastal stations outside the state outline are omitted.

latency depends only on architecture shape, so weights and trained quality are immaterial here (Figure 10). At $K \leq 1{,}024$ all three sit on a 1.5–3.5 ms dispatch floor, so the 34–90× speedup observed at $K \leq 256$ (Table 8) is attributable to reflow, not the encoder. The architectural advantage emerges in the crossover regime (NOAA US, K=5,580), where attention's K^2 cost begins to dominate, and is the binding constraint beyond it. At 1-NFE FNP exhausts memory at K=16,384 and NDP at K=65,536, while MFNP scales to $K=10^6$. The two contributions compose multiplicatively at NOAA US for the 628–784× end-to-end speedup reported in Table 2.

I Boundary-adjacency diagnostic

The Hilbert sort visits 2D points along a single space-filling curve, and at the curve's seams two points that are close in the plane can be placed far apart in the 1D order. Reversing a 1D sequence preserves all pairwise distances and index adjacencies, so π_d^{rev} does not relocate seams introduced by π_d; its role is to remove the causal-direction bias of the recurrence. The two distinct orderings π_1 and π_2 place seams in different locations, so a query whose neighbourhood is split under π_1 is typically contiguous under π_2. We test whether MFNP's predictive error concentrates on queries near these seams. Each query is ranked by its *isolation*, defined as the larger of the Euclidean distances to its two immediate Hilbert-sort neighbours in the combined context–query set, minimised across the axis-transposed orderings π_1 and π_2. We then compare the mean squared error of the top decile

Table 11: Architectural ablations on MambaFlowNP. MSE/CRPS on each task's native scale (dimensionless synthetic; $(°C)^2$/°C NOAA). Mean ± SE over 10 evaluation seeds at $N_{\text{samples}}=128$ on a single training checkpoint per variant; bold = best per column within block (within 1 SE); times on one NVIDIA L40S. C2ST omitted on NOAA tiers (no tractable reference posterior on real fields), shown as -. † our method.

	Predictive Quality		Calibration Quality		
Variant	**MSE** ↓	**CRPS** ↓	**C2ST** → 0.5	**Cov-95** → 0.95	**Time (ms)** ↓
1D Gaussian – Matérn-3/2 (K=128)					
Single direction	0.0471 ± 0.0012	0.0860 ± 0.0010	0.661 ± 0.012	0.893 ± 0.002	$\mathbf{8.0 \pm 0.1}$
Two directions	$\mathbf{0.0101 \pm 0.0002}$	$\mathbf{0.0373 \pm 0.0004}$	$\mathbf{0.515 \pm 0.003}$	0.936 ± 0.001	13.3 ± 0.1
Random order	0.0191 ± 0.0004	0.0552 ± 0.0005	0.619 ± 0.017	0.922 ± 0.001	28.8 ± 0.3
Single hidden state	0.0122 ± 0.0002	0.0425 ± 0.0004	0.562 ± 0.007	0.926 ± 0.001	10.1 ± 0.5
MFNP† (full, 2 dir. ×2)	0.0111 ± 0.0003	0.0398 ± 0.0004	0.537 ± 0.006	$\mathbf{0.938 \pm 0.001}$	29.5 ± 0.2
2D Gaussian – Matérn-3/2 (K=256)					
Single direction	0.0281 ± 0.0008	0.0709 ± 0.0008	0.584 ± 0.008	0.989 ± 0.000	$\mathbf{25.2 \pm 0.2}$
Two directions	$\mathbf{0.0163 \pm 0.0005}$	$\mathbf{0.0537 \pm 0.0006}$	0.575 ± 0.005	0.856 ± 0.001	52.1 ± 0.3
Random order	0.0197 ± 0.0006	0.0596 ± 0.0006	0.588 ± 0.007	$\mathbf{0.962 \pm 0.001}$	103.3 ± 0.3
Single hidden state	0.0196 ± 0.0005	0.0608 ± 0.0006	0.612 ± 0.009	0.885 ± 0.000	25.8 ± 0.2
MFNP† (full, 4-dir)	$\mathbf{0.0167 \pm 0.0005}$	$\mathbf{0.0537 \pm 0.0005}$	$\mathbf{0.554 \pm 0.005}$	0.971 ± 0.000	103.6 ± 0.3
NOAA Utah (K=222)					
Single direction	4.055 ± 0.008	0.996 ± 0.001	–	0.714 ± 0.000	6.04 ± 0.07
Two directions	2.528 ± 0.007	0.852 ± 0.001	–	0.698 ± 0.000	13.44 ± 0.26
Random order	4.940 ± 0.013	1.272 ± 0.002	–	0.645 ± 0.000	27.89 ± 0.10
Single hidden state	4.060 ± 0.008	1.007 ± 0.001	–	0.682 ± 0.000	$\mathbf{5.85 \pm 0.01}$
MFNP† (full, 4-dir)	$\mathbf{2.464 \pm 0.016}$	$\mathbf{0.767 \pm 0.002}$	–	$\mathbf{0.940 \pm 0.000}$	43.3 ± 0.4
NOAA California (K=486)					
Single direction	2.860 ± 0.006	0.858 ± 0.001	–	0.840 ± 0.000	$\mathbf{25.32 \pm 0.10}$
Two directions	1.965 ± 0.006	0.748 ± 0.001	–	0.830 ± 0.000	50.52 ± 0.13
Random order	9.111 ± 0.049	1.755 ± 0.004	–	0.935 ± 0.000	96.98 ± 0.11
Single hidden state	2.855 ± 0.006	0.861 ± 0.001	–	0.832 ± 0.000	$\mathbf{25.22 \pm 0.08}$
MFNP† (full, 4-dir)	$\mathbf{1.808 \pm 0.006}$	$\mathbf{0.681 \pm 0.001}$	–	$\mathbf{0.945 \pm 0.000}$	67.9 ± 0.5

Table 12: Point orderings on 2D Gaussian Matérn-3/2 (K=256). Each ordering is retrained with the same budget and one training seed; the random row is the random-order run of Table 11. MSE and CRPS relative to the Hilbert run, mean over 10 evaluation seeds at $N_{\text{samples}}=128$.

Ordering	**MSE / Hilbert** ↓	**CRPS / Hilbert** ↓
Hilbert	1.00	1.00
Morton	0.96	**0.98**
Lexicographic	**0.91**	1.16
Random	1.18	1.11

(*boundary*) against the bottom decile (*interior*) over 256 evaluation episodes per task. FNP, which uses full attention with no serialisation, serves as the no-sort control.

The diagnostic is run on 2D Gaussian Matérn-3/2 (K=256 uniform draws on $[0,1]^2$), NOAA Utah (K=222 stations), and NOAA California (K=486 stations). The synthetic task is the cleanest control, because a smooth isotropic stochastic process on a regular bounded domain isolates the diagnostic from the anisotropy, station clustering and discontinuous coastlines of the real-data tiers. Results are in Table 14.

On the synthetic task, the MFNP boundary/interior MSE ratio is 3.48 and the FNP ratio is 3.00. The two 95% bootstrap confidence intervals overlap. On both NOAA tiers, MFNP and FNP yield comparable ratios (1.40 vs. 1.54 on Utah; 1.43 vs. 1.47 on California), again with overlapping bootstrap intervals. The per-point Pearson and Spearman correlations between squared error and isolation remain below 0.18 in every cell, with $r^2 < 0.04$; 11 of the 12 have $p < 10^{-5}$ and one has p=0.05; this significance reflects the $n \geq 28{,}000$ sample size rather than effect size.

FNP, which never invokes the Hilbert sort, exhibits a boundary penalty of comparable magnitude to MFNP on every task. The penalty is therefore attributable to spatial extrapolation in regions with fewer informative context neighbours, not to the Hilbert serialisation. The synthetic task amplifies the absolute ratios while preserving the cross-architecture equivalence.

Table 13: Solver sweep on the reflow-distilled MFNP *student* and the matched *CFM-only teacher* (same architecture, hyperparameters, and evaluation; no reflow). Heun uses 2 network function evaluations per step; time = per-episode ms, on one NVIDIA L40S. Mean ± SE over 10 evaluation seeds; bold = best per column within model block (within 1 SE). C2ST undefined on NOAA.

	Predictive Quality		Calibration Quality		
Config	**MSE** ↓	**CRPS** ↓	**C2ST** → 0.5	**Cov-95** → 0.95	**Time (ms)** ↓
1D Gaussian – Matérn-3/2 (K=128)					
Teacher (pre-distillation)					
Euler 1-step	$\mathbf{0.0112 \pm 0.0003}$	0.0506 ± 0.0006	0.621 ± 0.005	0.302 ± 0.002	$\mathbf{28.5 \pm 0.1}$
Euler 15-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0397 \pm 0.0004}$	0.536 ± 0.005	0.903 ± 0.001	298.9 ± 0.3
Euler 30-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	0.533 ± 0.005	0.923 ± 0.001	585.4 ± 0.3
Heun 1-step	$\mathbf{0.0113 \pm 0.0003}$	0.0460 ± 0.0005	$\mathbf{0.527 \pm 0.005}$	0.925 ± 0.002	48.4 ± 0.3
Heun 15-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.531 \pm 0.005}$	$\mathbf{0.940 \pm 0.001}$	585.8 ± 0.4
Heun 30-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.531 \pm 0.005}$	$\mathbf{0.940 \pm 0.001}$	$(1.160 \pm 0.001) \times 10^3$
Student (reflow-distilled)					
Euler 1-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0398 \pm 0.0004}$	$\mathbf{0.537 \pm 0.006}$	0.938 ± 0.001	$\mathbf{29.5 \pm 0.2}$
Euler 15-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.532 \pm 0.005}$	0.942 ± 0.001	299.1 ± 0.7
Euler 30-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.532 \pm 0.005}$	0.942 ± 0.001	585.5 ± 0.4
Heun 1-step	$\mathbf{0.0112 \pm 0.0003}$	0.0403 ± 0.0004	$\mathbf{0.538 \pm 0.006}$	$\mathbf{0.954 \pm 0.001}$	48.7 ± 0.3
Heun 15-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.532 \pm 0.005}$	0.943 ± 0.001	585.2 ± 0.2
Heun 30-step	$\mathbf{0.0111 \pm 0.0003}$	$\mathbf{0.0396 \pm 0.0004}$	$\mathbf{0.532 \pm 0.005}$	0.943 ± 0.001	$(1.157 \pm 0.002) \times 10^3$
2D Gaussian – Matérn-3/2 (K=256)					
Teacher (pre-distillation)					
Euler 1-step	$\mathbf{0.0164 \pm 0.0005}$	0.0578 ± 0.0006	0.594 ± 0.007	0.669 ± 0.001	$\mathbf{104.0 \pm 0.3}$
Euler 15-step	$\mathbf{0.0164 \pm 0.0005}$	$\mathbf{0.0530 \pm 0.0005}$	$\mathbf{0.554 \pm 0.005}$	$\mathbf{0.962 \pm 0.001}$	$(1.459 \pm 0.000) \times 10^3$
Euler 30-step	$\mathbf{0.0164 \pm 0.0005}$	$\mathbf{0.0532 \pm 0.0005}$	$\mathbf{0.553 \pm 0.005}$	0.970 ± 0.001	$(2.909 \pm 0.001) \times 10^3$
Heun 1-step	$\mathbf{0.0164 \pm 0.0005}$	0.0569 ± 0.0006	0.565 ± 0.009	0.915 ± 0.001	198.2 ± 0.3
Heun 15-step	$\mathbf{0.0164 \pm 0.0005}$	$\mathbf{0.0535 \pm 0.0005}$	$\mathbf{0.552 \pm 0.005}$	0.976 ± 0.000	$(2.913 \pm 0.000) \times 10^3$
Heun 30-step	$\mathbf{0.0164 \pm 0.0005}$	$\mathbf{0.0535 \pm 0.0005}$	$\mathbf{0.552 \pm 0.005}$	0.976 ± 0.000	$(5.819 \pm 0.001) \times 10^3$
Student (reflow-distilled)					
Euler 1-step	$\mathbf{0.0167 \pm 0.0005}$	$\mathbf{0.0537 \pm 0.0005}$	$\mathbf{0.554 \pm 0.005}$	$\mathbf{0.971 \pm 0.000}$	$\mathbf{103.6 \pm 0.3}$
Euler 15-step	$\mathbf{0.0166 \pm 0.0005}$	$\mathbf{0.0537 \pm 0.0005}$	$\mathbf{0.553 \pm 0.005}$	0.975 ± 0.000	$(1.459 \pm 0.000) \times 10^3$
Euler 30-step	$\mathbf{0.0166 \pm 0.0005}$	$\mathbf{0.0538 \pm 0.0005}$	$\mathbf{0.553 \pm 0.005}$	0.975 ± 0.000	$(2.908 \pm 0.000) \times 10^3$
Heun 1-step	$\mathbf{0.0167 \pm 0.0005}$	0.0543 ± 0.0005	$\mathbf{0.555 \pm 0.004}$	0.979 ± 0.000	198.6 ± 0.3
Heun 15-step	$\mathbf{0.0166 \pm 0.0005}$	$\mathbf{0.0538 \pm 0.0005}$	$\mathbf{0.553 \pm 0.005}$	0.975 ± 0.000	$(2.914 \pm 0.001) \times 10^3$
Heun 30-step	$\mathbf{0.0166 \pm 0.0005}$	$\mathbf{0.0538 \pm 0.0005}$	$\mathbf{0.553 \pm 0.005}$	0.975 ± 0.000	$(5.824 \pm 0.001) \times 10^3$
NOAA Utah (K=222)					
Teacher (pre-distillation)					
Euler 1-step	$\mathbf{2.536 \pm 0.016}$	0.991 ± 0.002	–	0.309 ± 0.001	$\mathbf{28.26 \pm 0.08}$
Euler 15-step	2.559 ± 0.017	0.846 ± 0.002	–	0.669 ± 0.000	331.77 ± 0.15
Euler 30-step	2.571 ± 0.017	0.832 ± 0.002	–	0.725 ± 0.000	656.49 ± 0.19
Heun 1-step	3.075 ± 0.017	1.451 ± 0.001	–	$\mathbf{0.996 \pm 0.000}$	49.71 ± 0.06
Heun 15-step	2.579 ± 0.017	$\mathbf{0.813 \pm 0.002}$	–	0.811 ± 0.000	658.35 ± 0.17
Heun 30-step	2.582 ± 0.017	0.817 ± 0.002	–	0.793 ± 0.000	$(1.310 \pm 0.000) \times 10^3$
Student (reflow-distilled)					
Euler 1-step	$\mathbf{2.464 \pm 0.016}$	$\mathbf{0.767 \pm 0.002}$	–	$\mathbf{0.940 \pm 0.000}$	$\mathbf{43.3 \pm 0.4}$
Euler 15-step	$\mathbf{2.471 \pm 0.015}$	0.809 ± 0.002	–	0.771 ± 0.000	333.51 ± 0.19
Euler 30-step	$\mathbf{2.478 \pm 0.015}$	0.808 ± 0.002	–	0.779 ± 0.000	656.77 ± 0.10
Heun 1-step	3.896 ± 0.017	1.192 ± 0.001	–	0.988 ± 0.000	50.02 ± 0.06
Heun 15-step	2.483 ± 0.015	0.798 ± 0.002	–	0.828 ± 0.000	659.34 ± 0.16
Heun 30-step	2.484 ± 0.015	0.802 ± 0.002	–	0.808 ± 0.000	$(1.310 \pm 0.000) \times 10^3$
NOAA California (K=486)					
Teacher (pre-distillation)					
Euler 1-step	$\mathbf{1.994 \pm 0.006}$	0.917 ± 0.001	–	0.366 ± 0.000	$\mathbf{97.95 \pm 0.18}$
Euler 15-step	2.037 ± 0.006	0.803 ± 0.001	–	0.671 ± 0.000	$(1.373 \pm 0.000) \times 10^3$
Euler 30-step	2.050 ± 0.006	0.792 ± 0.001	–	0.715 ± 0.001	$(2.740 \pm 0.000) \times 10^3$
Heun 1-step	2.238 ± 0.005	1.331 ± 0.001	–	$\mathbf{0.999 \pm 0.000}$	188.96 ± 0.14
Heun 15-step	2.055 ± 0.006	$\mathbf{0.769 \pm 0.001}$	–	0.800 ± 0.000	$(2.738 \pm 0.000) \times 10^3$
Heun 30-step	2.060 ± 0.006	0.776 ± 0.001	–	0.777 ± 0.000	$(5.469 \pm 0.000) \times 10^3$
Student (reflow-distilled)					
Euler 1-step	$\mathbf{1.808 \pm 0.006}$	$\mathbf{0.681 \pm 0.001}$	–	$\mathbf{0.945 \pm 0.000}$	$\mathbf{67.9 \pm 0.5}$
Euler 15-step	1.973 ± 0.005	0.748 ± 0.001	–	0.827 ± 0.000	$(1.373 \pm 0.000) \times 10^3$
Euler 30-step	1.978 ± 0.005	0.749 ± 0.001	–	0.827 ± 0.000	$(2.739 \pm 0.000) \times 10^3$
Heun 1-step	3.368 ± 0.007	1.026 ± 0.001	–	0.850 ± 0.000	188.98 ± 0.17
Heun 15-step	1.986 ± 0.005	0.750 ± 0.001	–	0.832 ± 0.000	$(2.738 \pm 0.000) \times 10^3$
Heun 30-step	1.985 ± 0.005	0.750 ± 0.001	–	0.830 ± 0.000	$(5.469 \pm 0.000) \times 10^3$

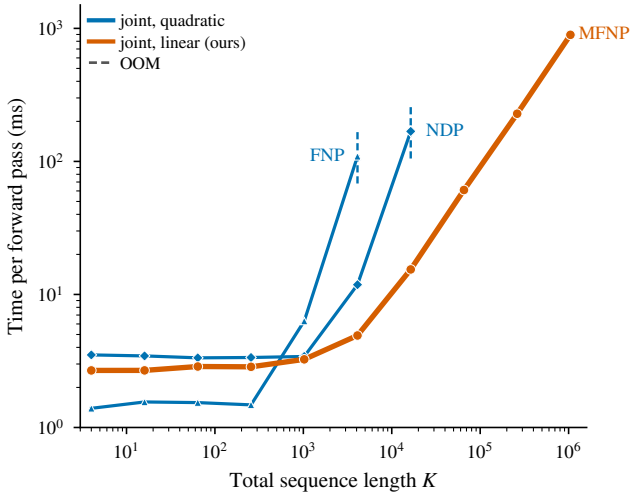

Figure 10: Time per forward pass vs. total sequence length K (log–log), 2D Gaussian Matérn-$3/2$, batch$=1$, *1-NFE for all models* and matched $\sim$600k parameters with random initialisation. We plot per-call latency rather than the time per joint predictive draw because at 1-NFE only MFNP produces a usable sample; NDP and FNP need their full solver budget to be sampled from in practice. A short dashed line through a curve's last point means the next K runs out of memory. With reflow's NFE budget equalised, NDP and FNP collapse onto MFNP's latency at $K \leq 1{,}024$ and diverge only once attention's K^2 cost dominates; FNP exhausts memory at $K{=}16{,}384$ and NDP at $K{=}65{,}536$.

Table 14: Boundary-adjacency diagnostic. Boundary and interior MSE are the top and bottom deciles of query targets ranked by isolation score (distance to the two Hilbert-sort neighbours). NOAA values are on the quantile-normalised target scale, so are not comparable to the $(°C)^2$ MSE values in Table 2; the 2D Matérn-$3/2$ values are on the native model output scale. Ratios are dimensionless; 95% CIs are from 2,000 bootstrap resamples. Correlations are per-point squared error against isolation; $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. † our method.

Model	Boundary MSE ↓	Interior MSE ↓	Ratio [95% CI]	Pearson r	Spearman ρ
2D Gaussian – Matérn-$3/2$ ($K{=}256$)					
FNP	0.0189	0.0063	3.00 [2.40, 3.77]	0.070^{***}	0.123^{***}
MFNP†	0.0267	0.0077	3.48 [2.74, 4.30]	0.064^{***}	0.130^{***}
NOAA Utah ($K{=}222$)					
FNP	0.0308	0.0200	1.54 [1.18, 1.99]	0.028^{***}	0.075^{***}
MFNP†	0.0260	0.0185	1.40 [1.07, 1.82]	0.012	0.059^{***}
NOAA California ($K{=}486$)					
FNP	0.0317	0.0215	1.47 [1.36, 1.60]	0.034^{***}	0.046^{***}
MFNP†	0.0211	0.0148	1.43 [1.31, 1.55]	0.019^{***}	0.043^{***}

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The four contributions enumerated in Section 1 (sub-quadratic joint-predictive NP, single-state context-compression diagnostic, lowest RMSE on every NOAA tier with 114–784$\times$ speed-up, and 10^6-point tractability) are each backed by a labelled table. Tables 2 and 1 in the body report accuracy, and Tables 11 and 8 in the appendix report the compression and routing ablations and the scaling.

   Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 contains a Limitations paragraph; Section 5.2 states the per-cell compute cap that excludes TNPA from the NOAA US tier; and the boundary diagnostic in Appendix I is included specifically to test a potential weakness of the Hilbert serialisation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper contains no formal theoretical results. The single-state compression argument in Section 4 is empirical, with the ablation in Table 11 (Appendix G) as supporting evidence.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The architecture is specified in Section 4 and Algorithms 1, 2, and 3 (Appendix B); Section 5.1 specifies task generation and NOAA splits; Section 5.2 and 5.3 specify the training and evaluation protocols; Appendix A (Tables 3, 4, and 5) lists the BO-selected per-cell architecture, learning rate, batch size, training-step counts, and inference NFE / guidance scale for every (model, task) cell.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: Code, trained weights, and training and evaluation scripts for the NOAA experiments are released at https://github.com/CosmoNaught/MambaFlowNP, with instructions to reproduce the results. The NOAA GHCN-Daily data is public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 5.2 specifies the optimiser family (AdamW), warmup-cosine schedule, EMA, Bayesian-optimisation sweep on validation MSE, and per-cell GPU-hour cap; Appendix A fills in the remaining details (weight decay 10^{-4}, gradient clip 1.0, $16{+}16$ BO trials per cell, and one sweep of 41 to 50 trials per task for MFNP on the synthetic tasks) and gives the per-cell architecture, learning rate, batch size, training steps, and inference settings in Tables 3, 4, and 5; Section 5.3 specifies the evaluation protocol (10 seeds, $N_{\text{samples}}{=}128$, denormalised scale, latency measurement).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The model-comparison, ablation and solver tables report mean $\pm$ one standard error across 10 evaluation seeds (Section 5.3), with each seed covering 256 synthetic episodes or $1{,}024$ NOAA episodes. The boundary diagnostic in Appendix I additionally reports 95% bootstrap confidence intervals from $2{,}000$ resamples, and the remaining table captions state their own statistics.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All training and evaluation run on NVIDIA L40S GPUs and all timings are on a single L40S (Sections 5.2 and 5.3), and each (model, task) cell is capped at 100 GPU-hours covering training and all 10 evaluation seeds (Section 5.2); per-episode inference latency is reported alongside every quality metric, and Table 8 reports forward-pass latency vs. sequence length up to $K{=}10^6$.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses publicly released NOAA temperature records and synthetic Gaussian-process simulations; no human subjects, scraped personal data, or sensitive attributes are involved, and the submission follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational methodological work on amortised joint-predictive inference for sparse spatial fields. Positive applications (temperature, disease prevalence and rainfall mapping) are noted in Section 1; we are not aware of a direct path to negative societal impact beyond those generic to scaling Bayesian inference.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released artefact is a regression model for spatial interpolation trained on public weather-station temperatures; it does not generate images, language, or any output with identifiable misuse potential.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All baselines (FNP, TNPA, NDP, CNP/NP/ANP/CANP), Mamba-2, conditional flow matching, reflow distillation, and Hilbert-curve point-cloud routing are cited at point of use in Sections 2.2–4; the NOAA GHCN-Daily dataset is in the U.S. public domain and cited as Menne et al. (2012b,a).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released code, weights and data are documented in the repository README.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper involves no crowdsourcing or human-subjects research; experiments use synthetic Gaussian-process draws and public weather-station records.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subjects research is performed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for prose editing and code-organisation assistance; they are not part of the core method (a Mamba-2 state-space scan trained with conditional flow matching) and do not affect the scientific rigour or originality of the work, so per the question's own guidance no declaration is required.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.