SPATFORMER: SIMULATION-BASED INFERENCE WITH TRANSFORMERS FOR SPATIAL STATISTICS

Herman F. Tesso herman.tesso@aalto.fi
AIMS South Africa, Aalto University

Ayush Bharti ayush.bharti@aalto.fi
Aalto University

Elizaveta Semenova elizaveta.p.semenova@gmail.com
Imperial College London

ABSTRACT

Gaussian Process (GP) priors are widely used in spatial statistical models but suffer from cubic computational complexity in Markov Chain Monte Carlo (MCMC), limiting scalability. We propose a simulation-based inference (SBI) method using a transformer-enhanced diffusion model tailored for spatial models with latent GP priors. By leveraging transformers for sequence modeling and probabilistic diffusion for inference, our approach enables efficient, amortized Bayesian inference. Unlike traditional MCMC, it avoids repeated costly computations, allowing rapid exploration of spatial conditional distributions. Simulation experiments on one- and two-dimensional models demonstrate superior scalability compared to GP-based MCMC, making our method well-suited for large-scale applications in spatial epidemiology and other domains. The accompanying code is available at https://github.com/hermanFTT/SpatFormer.

1 INTRODUCTION

Spatial statistical models are widely used in environmental science and epidemiology to inform policy, optimize interventions, and allocate resources to high-risk areas (Gelfand, 2012; Wang et al., 2024). These models often rely on Gaussian Process (GP) priors (Rasmussen, 2003) to capture spatial correlation structures. However, the cubic complexity of GP kernel operations presents a major computational bottleneck, particularly in Bayesian inference settings that require Markov Chain Monte Carlo (MCMC).

A common approach to modeling spatial fields is to discretize the study region into a grid, where each cell represents a location at which the process is evaluated. As the resolution of this grid increases – whether to capture fine-scale spatial variation or to accommodate larger study areas – the number of locations grows, dramatically increasing the computational cost of GP-based inference. MCMC, while essential for handling non-Gaussian likelihoods and hierarchical dependencies, scales poorly with larger grids (Rue et al., 2009). Developing inference methods that efficiently scale with grid resolution is therefore crucial for enabling the practical application of spatial models in large-scale epidemiological and environmental studies.

In this paper, we tackle these shortcomings by exploring simulation-based inference (SBI) techniques (Cranmer et al., 2020): a class of methods that utilize forward simulations from the model to approximate posterior distributions when direct evaluation of the likelihood (or its gradient) is infeasible. Specifically, we leverage the capabilities of a recently developed amortised SBI method called Simformer (Gloeckler et al., 2024b), which combines the expressive power of diffusion models with the ability of transformers to handle sequence data. Our approach, called **SpatFormer**, adapts

Simformer for inference problems offering a new framework for efficient, scalable, and amortized nonparametric inference on spatial models with latent GP priors. Although approximate, the latter is highly flexible and provides a light-weight alternative to MCMC methods in this context.

2 PRELIMINARIES

Problem setting. In applied fields like spatial disease mapping (Lawson, 2018), hierarchical Bayesian generalized linear mixed models (GLMMs) offer a unified framework for modeling both discrete and continuous outcomes, provided they belong to the exponential family of distributions. This hierarchy is typically structured as follows:

$$\theta \sim p(\theta), \quad \mathbf{f} \mid \mathbf{x}, \theta \sim \mathcal{GP}\big(\mu(\mathbf{x}), K_\theta(\mathbf{x}, \mathbf{x}')\big), \quad \mathbf{y} \mid \mathbf{f} \sim p\big(g^{-1}(\mathbf{f})\big).$$

Here, θ represents hyperparameters governed by a hyperprior distribution $p(\theta)$. The latent spatial field $\mathbf{f}$ is modeled as a Gaussian Process (GP), which captures spatial correlations among input locations $\mathbf{x} \in \mathcal{X}$. The observational model p incorporates a link function g (e.g., log or logistic) to transform the latent process $\mathbf{f}$ onto a suitable scale for discrete or continuous responses. The GP is specified by a mean function $\mu : \mathcal{X} \to \mathbb{R}$, often set to zero, and a covariance function $K_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, parameterized by θ. The goal is to infer the predictive distribution: $p(\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{x}_{\text{obs}}, \mathbf{y}_{\text{obs}}) = \int p(\mathbf{y}_* \mid \mathbf{f}_*) p(\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{x}_{\text{obs}}, \mathbf{y}_{\text{obs}})\, d\mathbf{f}_*$ where $\mathbf{x}_*$ represents the new locations where predictions are made, and $\mathbf{f}_*$ is the corresponding latent GP realization. The predictive distribution is obtained by integrating out the latent process, ensuring proper uncertainty quantification.

To perform inference, one typically discretizes the spatial domain $\mathcal{X}$ into a *computational grid* of n locations, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. For these n spatial points, the latent process $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ follows $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$, where $\boldsymbol{\mu} = [\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_n)]^\top$ and $\mathbf{K} \in \mathbb{R}^{n \times n}$ has entries $K_\theta(\mathbf{x}_i, \mathbf{x}_j)$. Unless this kernel matrix is trivial (*e.g.*, the identity), its factorization requires $\mathcal{O}(n^3)$ operations and must be recomputed for each new value of θ in MCMC. These costs become prohibitive as n grows, particularly on large or dense spatial grids, motivating the use of amortized simulation-based inference methods to circumvent repeated factorization, which we introduce next.

Simulation-based inference (SBI). SBI has emerged as a framework for inferring parameters of mechanistic models with intractable likelihood functions, common in many fields such as epidemiology (Kypraios et al., 2017), particle physics (Brehmer, 2021), and radio propagation (Bharti et al., 2022). Let $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$ denote the family of distributions induced by the model with parameters θ. If the likelihood function $p(\cdot \mid \theta)$ associated with $\mathbb{P}_\theta$ cannot be evaluated point-wise, then standard Bayesian inference becomes infeasible. However, if it is straightforward to simulate independent realisations $\mathbf{y} \sim \mathbb{P}_\theta$, then SBI methods can be used to approximate the posterior $p(\theta \mid \mathbf{y}) \propto p(\mathbf{y} \mid \theta) p(\theta)$.

Amortized SBI. A popular SBI method is neural posterior estimation (NPE) (Papamakarios & Murray, 2016; Radev et al., 2022), which utilizes conditional density estimators, such as normalizing flows, to learn an approximate mapping from each data $\mathbf{y}$ to the posterior distribution $p(\theta \mid \mathbf{y})$. NPE assumes that the posterior is a member of the conditional density model $q_\phi(\theta \mid \mathbf{y})$, where ϕ constitutes the learnable parameters. Given data pairs $\{(\theta_i, \mathbf{y}_i)\}_{i=1}^m \sim p(\theta, \mathbf{y})$, generated by first sampling from the prior $\theta_i \sim p(\theta)$ and then simulating from the model $\mathbf{y}_i \sim \mathbb{P}_{\theta_i}$, q_ϕ can be trained by minimizing the empirical loss $\ell(\phi) = \frac{1}{m} \sum_{i=1}^m q_\phi(\theta_i \mid \mathbf{y}_i) \approx \mathbb{E}_{\theta, \mathbf{y} \sim p(\theta, \mathbf{y})} [q_\phi(\theta \mid \mathbf{y})]$. Once trained, the NPE posterior $q_{\hat{\phi}}(\theta \mid \mathbf{y}_{\text{obs}})$ can then be obtained by a simple forward pass of the observed data $\mathbf{y}_{\text{obs}}$ through the trained network. This property of NPEs makes them *amortized*, wherein inference on a new dataset is obtained almost instantaneously, as opposed to MCMC inference where all the computations need to be carried out again. However, NPEs cannot handle non-parametric inference wherein the input to the simulator (*e.g.* the spatial coordinate $\mathbf{x}$) is function-valued and thus ∞-dimensional (Ramesh et al., 2022), as is the case with spatial models. Note that one could in principle try to infer the hyperparameters θ and obtain the predictive distribution by marginalizing over the posterior distribution, $p(\mathbf{y} \mid \mathbf{y}_{\text{obs}}) = \int p(\mathbf{y} \mid \theta) p(\theta \mid \mathbf{y}_{\text{obs}}) d\theta$. But this fails to capture spatial correlations between $\mathbf{x}$ and $\mathbf{y}$, motivating our direct approach to the predictive density inference.

Simformer. Simformer (Gloeckler et al., 2024a) is a more flexible amortized SBI method that accommodates inference in ∞-dimensional parameter spaces, i.e., nonparametric inference. It is a

score-based diffusion model that uses a transformer architecture as a time-conditional score estimator $s_{\boldsymbol{\eta}(\cdot,t)}$, where $\boldsymbol{\eta}$ represents the weights vector of the score model. Unlike previous SBI methods that use conditional density estimators for the posterior or likelihood, Simformer is trained on the joint distribution of latent parameters and data $p(\theta, \mathbf{y})$. It can encode known causal relationships in the attention mask of the transformer and make it possible to sample arbitrary conditionals of the joint distribution $p(\theta, \mathbf{y})$. Each input sequence $(\theta, \mathbf{y})$ is processed by a data-specific tokenizer (Gu et al., 2024). The transformer then transforms the tokenized inputs into a latent representation, which is decoded by a linear layer to obtain the score estimate. This estimate is subsequently employed during the reverse diffusion process to sample the desired conditionals. Despite its flexibility, Simformer cannot be readily applied to spatial models due to two reasons: (i) it does not capture the characteristics of spatially correlated data as the training is not informed by data locations and (ii) the standard reverse diffusion process in Simformer, with constant conditioning values, does not account for observation noise in the data. We address both these problems in the next section.

3 THE PROPOSED METHOD: SPATFORMER

In order to use Simformer for inference of spatial models, we first design a tokenizer for capturing the spatial structure between the data $\mathbf{y}$ and the coordinates $\mathbf{x}$, and then implement a sampling mechanism that models observation noise.

Designing the tokenizer. We used realizations $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ of GP priors to simulate training data $\mathbf{y} = [y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)]^\top \sim p(g^{-1}(\mathbf{f}))$. We then design a tokenizer that embeds spatial coordinates $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ alongside the observed data locations, seamlessly integrating both into the training process.

Prior to being fed into the transformer architecture, the diffused sequences at time level t of the forward diffusion process (see A.1)—$\mathbf{y}_t = [y_t(\mathbf{x}_1), \ldots, y_t(\mathbf{x}_n)]$ pass through a specifically designed tokenizer which represents each variable in the sequence with three components: the *spatial coordinate* that uniquely identifies the variable, a representation of the *value* of the variable, and a *condition state*. The condition state is a binary variable that indicates whether the variable is conditioned upon or not. The condition states of all variables, represented as $\mathbf{C}_M \in \{0, 1\}^n$ collectively form a mask indicating observed and unobserved data locations. Adopting $\mathbf{C}_M^{(i)} = 1$ for certain variables in the sequence implies they remain uncorrupted during the forward diffusion, focusing inference on variables with $\mathbf{C}_M^{(i)} = 0$ which result in the training of a conditional diffusion model. The tokenizer then jointly embeds the *values*, *coordinates*, and the *condition mask* into a unified vector representation in a continuous dimensional space. In our experiments, we just replicate values into vectors of a chosen latent dimension for *value embeddings* and use learnable vector embeddings for the *coordinates* and *condition states*. We embed the diffusion time with Fourier transforms to enable the transformer to incorporate temporal information into the score estimates. A dense symmetric attention mask A_M enforces global attention, allowing all variables in the sequence to attend to each other.

The transformer takes as input the result embedding from the tokenizer and incorporates the defined attention mask A_M in the attention layers to produce the score. Condition masks $\mathbf{C}_M$ can be set during training if targeting specific conditionals. At noise level t, the forward diffusion generates a partially noisy sample described by $\mathbf{y}_t^{\mathbf{C}_M} = (1 - \mathbf{C}_M) \cdot \mathbf{y}_t + \mathbf{C}_M \cdot \mathbf{y}_0$, ensuring that the variables we want to condition on remain unchanged. The score model $s_{\boldsymbol{\eta}(\cdot,t)}$ (transformer) is then trained via time conditional denoising score-matching (Vincent, 2011) to minimize the loss

$$\mathcal{L}(\boldsymbol{\eta}) = \mathbb{E}_{\mathbf{C}_M, t, \mathbf{y}_0, \mathbf{y}_t}\left[\parallel (1 - \mathbf{C}_M). \left(s_\eta^{A_M}(\mathbf{y}_t^{\mathbf{C}_M}, t) - \nabla_{\mathbf{y}_t} \log q_t(\mathbf{y}_t \mid \mathbf{y}_0) \right) \parallel_2^2 \right] \tag{1}$$

where $s_{\boldsymbol{\eta}}^{A_M}$ denotes the score model equipped with a specific attention mask A_M, $\boldsymbol{\eta}$ represents the weights vector of the score model and $q_t(\mathbf{y}_t \mid \mathbf{y}_0) = \mathcal{N}(\mathbf{y}_t; \mu_t(\mathbf{y}_0), \sigma_t(\mathbf{y}_0))$ represents the forward diffusion process at time step t. Figure 1 depicts the architecture and its essential elements.

Posterior sampling with observation noise. To address observation noise, we solve the reverse diffusion process using the data $\mathbf{y}_{\text{obs}}$ and the specified noise model $p(. \mid \mathbf{y}_{\text{obs}})$ at the observed locations. Once the model is trained, we leverage the uncertainty associated with the observed data $\mathbf{y}_{\text{obs}}$, which contains missing values at specific locations indicated by the condition mask $\mathbf{C}_M$. This

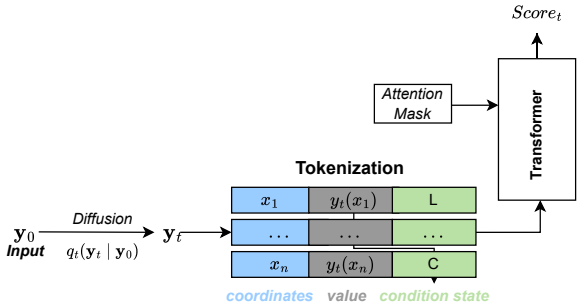


Figure 1: SpatFormer's architecture. The diffused inputs are simplified into a token representation that includes spatial coordinates, value, and conditional state (latent (L) or conditioned (C)). This sequence of tokens is then processed by a transformer architecture, where variable interactions are managed using an attention mask. The transformer then returns a score estimate, which is used to generate samples from the score-based diffusion model *Alg* 2.

uncertainty is used to generate the noisy data $\tilde{\mathbf{y}}$, which guides the sampling mechanism. The process is as follows:

- Draw $\mathbf{y}_T \sim q_T(\mathbf{y}_T) \approx \mathcal{N}(0, I)$ from the diffusion process at the final time step T.
- Sample $\tilde{\mathbf{y}}_{obs} \sim p(. \mid \mathbf{y}_{\text{obs}})$ and use $\tilde{\mathbf{y}}_T = (1 - \mathbf{C}_M) \cdot \mathbf{y}_T + \mathbf{C}_M \cdot \tilde{\mathbf{y}}_{obs}$ as the initial state to numerically solve the reverse diffusion equation (see *Alg* 2) instead of directly using $\mathbf{C}_M.\mathbf{y}_{obs}$ as is done in standard sampling with Simformer.

4 Simulation Experiments

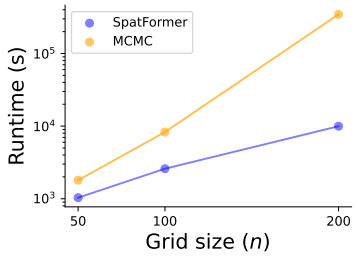
Implementation details. We assess the performance and applicability of SpatFormer through experiments focused on Gaussian process inference. We conduct inference on continuous data $\{y_i\}_{i=1}^n$, on a regular one-dimensional grid of $n = 100$ points over the interval $[0, 1]$. The observations consist of points spaced at irregular intervals subsampled from a single realization of a GP with zero mean and a standardized radial basis function (RBF) kernel (equation 5). We vary the number of observed data points to small percentages (e.g., 6%, 10%) of the total number of grid points, aiming to reconstruct the true function. Using Matérn kernel (equation 6) for the latent Gaussian field, we simulate training data $\mathbf{y}$ according to the following hierarchical structure:

$$l \sim \mathcal{U}(0.1, 1),\ \epsilon \sim \mathcal{N}^+(0, 0.2), \quad \mathbf{f} \mid \mathbf{x}, l \sim \mathcal{GP}\left(0, K^{\text{Matérn}}_{l,\nu=2.5}(\mathbf{x}, \mathbf{x}')\right), \quad \mathbf{y} \mid \mathbf{f} \sim \mathcal{N}\left(\mathbf{f}, \epsilon^2 \boldsymbol{I}\right) \tag{2}$$

The *SpatFormer* architecture employs *variance exploding SDE* for the diffusion process along with a tokenizer that concatenates a 20-dimensional *value* and *node ID* embedding with a 10-dimensional *condition* embedding. The resulting embeddings are processed by a transformer consisting of two attention heads, each with an attention size of 10 across two layers, and a widening factor of 10 is applied. Training is carried out over 25 epochs in simulated data $\mathbf{y} \sim \mathcal{N}\left(\mathbf{f}, \epsilon^2 \boldsymbol{I}\right)$, with each epoch containing an inner loop of 5000 iterations. Random masks are applied for efficient training of arbitrary conditional of the joint distribution $p(\mathbf{y})$. Once trained, we sample from the target conditionals defined by the corresponding masks, following the process outlined earlier in 2. The training and post-training inference times for various numbers of observed data points n_{obs} are recorded. We also conduct GP-based MCMC inference and record the running times for efficiency comparison. Furthermore, we investigate how performance improves as the number of simulations (training data) increases by evaluating the Maximum Mean Discrepancy (MMD) between the ground truth target conditional distribution provided by MCMC and the inference results of the Simformer.

SpatFormer is computationally scalable. Figure 2 illustrates how inference time scales with the grid size and the number of observed data, clearly highlighting the efficiency of the SpatFormer compared to MCMC-based inference. Figure 5 shows that the model is not simulation-hungry; the

distance from the ground truth distribution, measured by MMD, stabilises as the amount of training data increases. Figure 3 (right) depicts the competitive inference results of the Simformer approach for three sets of observed data points. GP-based MCMC inference requires re-running the expensive inference process for each new observed data points, whereas the Simformer provides a single network that can be queried to sample any conditional distribution induced by new observations, as long as their locations align with the condition masks applied during training.



(a) Runtime across grid sizes.

n_{obs}	*MCMC*	*SpatFormer*
10	6.45×10^3	3.2×10^3
20	6.51×10^3	3.41×10^3
40	6.6×10^3	3.6×10^3
80	7.01×10^3	3.7×10^3

(b) Runtime for varying #observed data

Figure 2: Comparison of inference runtimes (in seconds) for MCMC and the combined training/post-training inference time for*SpatFormer*. The figure (left) illustrates how runtime varies with grid size. MCMC uses the HMC algorithm with $5,000$ iterations, including a $1,000$-step warm-up, while the SpatFormer model generates 10^4 samples after being trained with 10000 simulations. The table (right) presents average runtimes for different numbers of observed data points on a fixed 1D grid of size $n = 100$. Results highlight the superior scalability of *SpatFormer* over MCMC.

SpatFormer is able to model observation noise. The effects of our adjustments on the inference results for the specific task are illustrated by Figure 3. When the model lacks information about spatial coordinates and the locations of observed data—either because the tokenizer does not embed the coordinates during training or the embedding dimension is insufficiently expressive, the Simformer fails to capture the variables' dependencies, producing non-smooth samples with high variability around unobserved locations. Integrating spatial components into the tokenizer while maintaining the standard Simformer sampling procedure enables the model to capture the pattern but tends to overfit the data, assuming no observation noise at the observed locations. By further refining the sampling process, as described in the Methods section, the architecture can effectively model observation noise while accurately capturing the intrinsic correlations across dimensions.

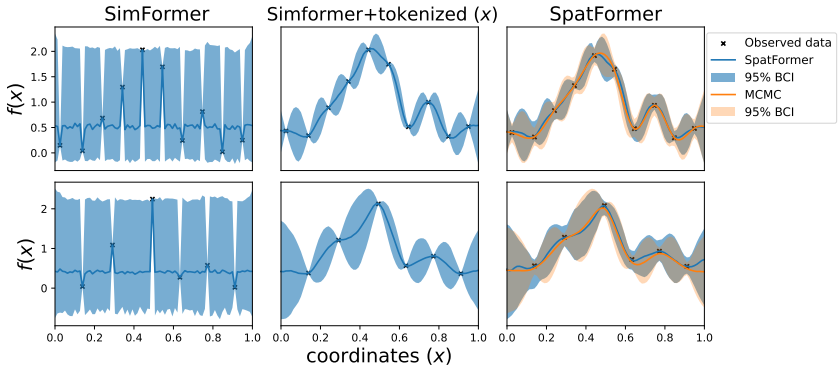


Figure 3: Illustration of improvements in inference results on a one-dimensional grid (n=100). (Left) The standard Simformer, uninformed by spatial coordinates, demonstrates poor accuracy and high variability in unobserved locations. (Middle) The adjusted Simformer incorporates a tokenizer that encodes spatial coordinates and observed data locations, enhancing inference quality but failing to model noise at those locations. (Right) *SpatFormer* combines spatial tokenization with an adjusted sampling operation that accounts for uncertainty around observed data points. The solid lines correspond to the mean estimates, and the shaded areas indicate the 95% Bayesian credible intervals.

SpatFormer efficiently scales to higher dimension. A similar experiment, as described above, is conducted on a regular grid defined over a square with 20 segments along each coordinates (two-dimensional), resulting in $n = 400$ grid cells. The Simformer effectively managed the computational burden and convergence issues that the MCMC method faced due to the high dimensionality resulting from the number of grid points, further demonstrating its scalability. As the number of observed points increases, the estimated mean becomes closer to the ground truth surface Figure 4. Regions with sparse data points exhibit higher uncertainty, while areas informed by closely located data points show reduced uncertainty.

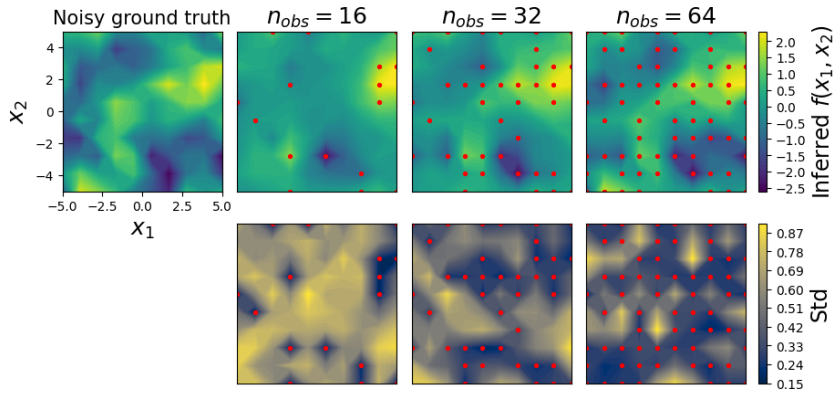


Figure 4: Demonstration of the applicability of *SpatFormer* on a two-dimensional square grid sized n=20×20. Inference is conducted using 4%, 8%, and 16% of the total grid points respectively, aiming to reconstruct the ground truth surface. The mean surface estimates (top row) are based on 5000 posterior samples. The red scatter points indicate the observed locations. As the number of observations increases, the uncertainty in the mean surface estimates (bottom row) decreases. While a single MCMC run takes several days without guaranteed convergence, *SpatFormer* completes training and inference in 6 hours and enables efficient sampling of arbitrary spatial conditionals.

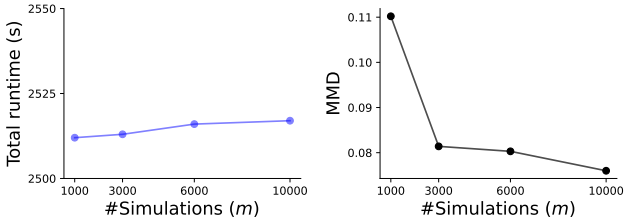


Figure 5: (Left) Average total runtime of SpatFormer as a function of simulations (training dataset), with the grid size (one-dimensional) fixed at n=100 and $n_{obs} = 10$, drawing 10^4 samples. (Right) *Maximum mean discrepancy* (measured using the RBF kernel) between the inferred conditional distributions from SpatFormer (based on different amounts of simulated data) and the ground truth distribution obtained with MCMC inference. SpatFormer maintains a relatively stable runtime, with a slight improvement in performance (MMD) as the number of simulated data m increases.

5 Conclusion and Future Work

This paper explores a novel framework for spatial statistical inference that involves Gaussian process priors. The approach is an amortized SBI algorithm based on generative models (diffusion models, transformers) equipped with a specifically designed tokenizer and an inference procedure that accounts for noise in observation. The architecture can accommodate non-parametric inference, and it exhibits strong performance and flexibility in managing inference over spatially correlated structures, which have so far relied on MCMC algorithms. Its main advantages are that it enables inference of multiple spatial conditional distributions at once, can learn from training experience to

generalize to newly observed data (amortized inference), and scales efficiently compared to MCMC-based approaches. Although we used Gaussian likelihood in our experiments, the method can be applied to any other likelihood from the exponential family. However, because Spatformer is trained on a discretization of a predefined grid, it cannot perform off-grid prediction.

Future work includes comparing our proposed method to two-stage approach like PriorVAE, PriorCVAE (Semenova et al., 2022; 2023), in small-area estimation regime (Rao & Molina, 2015), as well as to state-of-the-art scalable Bayesian GP models, such as KISS-GP (Wilson & Nickisch, 2015) and Vecchia approximation (Katzfuss & Guinness, 2021) , on real world large-scale spatial grid problems.

ACKNOWLEDGMENTS

H.T acknowledges the support of Google DeepMind and the African Institute for Mathematical Sciences (AIMS), South Africa, for the opportunity to conduct this research as a part of his MSc. AB was supported by the Research Council of Finland grant no. 362534. E.S. acknowledges AIMS-SA for an opportunity to teach and meet students. E.S. acknowledges supported in part by the AI2050 program at Schmidt Sciences (Grant [G-22-64476]).

REFERENCES

Ayush Bharti, Francois-Xavier Briol, and Troels Pedersen. A general method for calibrating stochastic radio channel models with kernels. *IEEE Transactions on Antennas and Propagation*, 70(6): 3986–4001, 2022. doi: 10.1109/tap.2021.3083761.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. 2018.

Johann Brehmer. Simulation-based inference in particle physics. *Nature Reviews Physics*, 3(5): 305–305, 2021.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Alan E Gelfand. Hierarchical modeling for spatial data problems. *Spatial statistics*, 1:30–39, 2012.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. All-in-one simulation-based inference. *arXiv preprint arXiv:2404.09636*, 2024a.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H. Macke. All-in-one simulation-based inference, 2024b.

Yuchao Gu, Xintao Wang, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Rethinking the objectives of vector-quantized tokenizers for image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7631–7640, 2024.

Matthias Katzfuss and Joseph Guinness. A general framework for vecchia approximations of gaussian processes. *Statistical Science*, 36(1):124–141, 2021.

Theodore Kypraios, Peter Neal, and Dennis Prangle. A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical Biosciences*, 287:42–53, May 2017. ISSN 0025-5564. doi: 10.1016/j.mbs.2016.07.001.

Andrew B Lawson. *Bayesian disease mapping: hierarchical modeling in spatial epidemiology*. Chapman and Hall/CRC, 2018.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2022. doi: 10.1109/TNNLS.2020.3042395.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S Greenberg, Pedro J Gonçalves, and Jakob H Macke. Gatsbi: Generative adversarial training for simulation-based inference. *arXiv preprint arXiv:2203.06481*, 2022.

John NK Rao and Isabel Molina. *Small area estimation*. John Wiley & Sons, 2015.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(2):319–392, 2009.

Elizaveta Semenova, Yidan Xu, Adam Howes, Theo Rashid, Samir Bhatt, Swapnil Mishra, and Seth Flaxman. Priorvae: encoding spatial priors with variational autoencoders for small-area estimation. *Journal of the Royal Society Interface*, 19(191):20220094, 2022.

Elizaveta Semenova, Prakhar Verma, Max Cairney-Leeming, Arno Solin, Samir Bhatt, and Seth Flaxman. Priorcvae: scalable mcmc parameter inference with bayesian deep generative modelling. *arXiv preprint arXiv:2304.04307*, 2023.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Yufeng Wang, Xue Chen, and Feng Xue. A review of bayesian spatiotemporal models in spatial epidemiology. *ISPRS International Journal of Geo-Information*, 13(3):97, 2024.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International conference on machine learning*, pp. 1775–1784. PMLR, 2015.

A APPENDIX

We used JAX (Bradbury et al., 2018) as the primary framework. The code is optimized for parallel computation across multiple cores and benefits from GPU acceleration. Experiments have been conducted on a CUDA-capable machine equipped with two T4 GPU devices, each with 15 GiB of memory.

Algorithm 1 SpatFormer training

Set attention mask A_M (*causal statement in the input sequence*)
repeat
  set condition mask $\mathbf{C}_M$
  simulate $\boldsymbol{f}_0 \sim \mathcal{GP}(.)\,,\ \boldsymbol{y}_o \sim p(u^{-1}(\boldsymbol{f}_o))$
  $t \sim Uniform(0, T)$ (*sample diffusion time*)
  diffuse sample $\boldsymbol{y}_t \sim q_t(\boldsymbol{y}_t \mid \boldsymbol{y}_0)$ (*forward diffusion*)
  $\mathrm{y}_t^{\mathbf{C}_M} = tokenize\{(1 - \mathbf{C}_M) \cdot \boldsymbol{y}_t + \mathbf{C}_M \cdot \boldsymbol{y}_0 \}$
  Take gradient descent step $\nabla_{\boldsymbol{\eta}}$ (*denoising score-matching*)
  $\mathcal{L}(\boldsymbol{\eta}) = \parallel (1 - \mathbf{C}_M).\left(s_\eta^{A_M}(\mathbf{y}_t^{\mathbf{C}_M}, t) - \nabla_{\mathbf{y}_t} \log q_t(\mathbf{y}_t \mid \mathbf{y}_0)\right) \parallel_2^2$
until Convergence (=0

Algorithm 2 SpatFormer inference

Initialize $t \leftarrow T\ ,\quad \tilde{\boldsymbol{y}}_T \sim q_T(\tilde{\boldsymbol{y}}) \approx \mathcal{N}(0, I)$
$\tilde{\mathbf{y}}_{obs} \sim p_{(.\,\mid \mathbf{y}_{\text{obs}})}$ *(observation noise model)*
$\boldsymbol{y} = (1 - \mathbf{C}_M) \cdot \tilde{\boldsymbol{y}}_T + \mathbf{C}_M \cdot \tilde{\mathbf{y}}_{obs}$
repeat
  $z \sim \mathcal{N}(0, |\Delta t|\, I)$
  $\Delta \boldsymbol{y} \leftarrow \left[f(\boldsymbol{y}, t) - g^2(t) S_\eta(\boldsymbol{y}^{C_M}, t)\right] \Delta t + g(t) z$
  $\boldsymbol{y} \leftarrow \boldsymbol{y} + \Delta \boldsymbol{y}$ *(Euler-Maruyama numerical solver)*
  $t \leftarrow t + \Delta t$
until t =0

A.1 SCORE-BASED DIFFUSION MODELS

In the forward process of a generative diffusion model, a real data point is gradually corrupted by adding Gaussian noise at each time step, ultimately resulting in pure noise. In the limit of many small time steps, the diffusion process that transforms an initial data distribution $q_0(x_0) = p(x)$ into a simpler distribution $q_T(x_T) = \mathcal{N}(x_T, 0, \mathcal{I})$ is defined as the solution to stochastic differential equations (SDEs), $dx_t = f(x_t, t)dt + g(t)dw_t$, with w_t being a standard Wiener process, and f and g representing the drift (controlling deterministic properties of the stochastic process) and diffusion coefficients, respectively. Samples from the generative model are then generated by simulating the corresponding reverse diffusion process $dx_t = [f(x_t, t) + g(t)^2 \nabla_{x_t} \log q_t(x_t)]dt + g(t)d\tilde{w}_t$ which relies on the knowledge of the score function $\nabla_{x_t} \log q_t(x_t)$. The exact score is typically intractable but can be estimated through time conditional denoising score-matching (Vincent, 2011). Given that the conditional score is known, $q_t(x_t \mid x_0) = \mathcal{N}(x_t; \mu_t(x_0), \sigma_t(x_0))$, a score model $s_\eta(x_t, t)$ is trained to minimize the loss

$$\mathcal{L}(\theta) = \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\substack{\text{diffusion}\\ \text{time } t}} \underbrace{\mathbb{E}_{x_0\sim q_0(x_0)}}_{\substack{\text{data}\\ \text{sample } x_0}} \underbrace{\mathbb{E}_{x_t\sim q_t(x_t\mid x_0)}}_{\substack{\text{diffused data}\\ \text{sample } x_t}} \left[\lambda(t) \parallel \underbrace{s_\eta(x_t, t)}_{\substack{\text{neural}\\ \text{network}}} - \underbrace{\nabla_{x_t} \log q_t(x_t \mid x_0)}_{\substack{\text{score of diffused}\\ \text{data sample}}} \parallel_2^2 \right] \tag{3}$$

where $\lambda(t)$ denotes a positive weighting function and η represents the weights vector of the score model (neural network architecture). After expectation, $s_\eta(x_t, t) \approx \nabla_{x_t} \log q_t(x_t)$.

A.2 TRANSFORMERS

Transformers are a type of neural network architecture primarily used in NLP that overcome the limitations of feed-forward networks in effectively dealing with sequential inputs by incorporating an attention mechanism to capture long-range dependencies in data. They operate by encoding input sequences through three learnable linear projections: *query* (Q), *key* (K), and *value* (V), where *query* represents the current elements, *key* helps compute attention scores, and *value* carries the actual information. The process involves transforming input embeddings into Q, K, and V using learned weight matrices, computing scaled attention scores via the dot product of Q and K , normalizing these scores with *softmax* to obtain attention weights, and finally producing an output as a weighted sum of the values (Vaswani, 2017) followed by feedforward neural networks for processing.

$$\text{Attention}(Q, K, V) = \text{Softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big).V \tag{4}$$

A.3 KERNELS

RBF kernel:

$$K(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right), \tag{5}$$

Matérn Kernel ($\nu = 2.5$) :

$$K(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}d}{l}\right)^\nu K_\nu \left(\frac{\sqrt{2\nu}d}{l}\right) \tag{6}$$

where: $d = \|x - x'\|$ is the Euclidean distance between the two points, l is the length scale, ν is the smoothness parameter, σ^2 is the variance, K_ν is the modified Bessel function of the second kind, $\Gamma(\nu)$ is the gamma function.