Efficient Inference for Coupled Hidden Markov Models in Continuous Time and Discrete Space

Giosue Migliorini
Department of Statistics
University of California, Irvine

Padhraic Smyth
Department of Computer Science
University of California, Irvine

Abstract

Systems of interacting continuous-time Markov chains are a powerful model class, but inference is typically intractable in high-dimensional settings. Auxiliary information, such as noisy observations, is typically only available at discrete times, and incorporating it via a Doob's h-transform gives rise to an intractable posterior process that requires approximation. We introduce Latent Interacting Particle Systems, a model class parameterizing the generator of each Markov chain in the system. Our inference method involves estimating look-ahead functions (twist potentials) that anticipate future information, for which we introduce an efficient parameterization. We incorporate this approximation in a twisted Sequential Monte Carlo sampling scheme. We demonstrate the effectiveness of our approach on a challenging posterior inference task for a latent SIRS model on a graph, and on a neural model for wildfire spread dynamics trained on real data.

1 INTRODUCTION

Many real-world phenomena, from epidemics to wildfires, can be modeled as systems of interacting components evolving in continuous time, where the underlying dynamics are governed by discrete latent states (Lanchier, 2024). This modeling approach builds upon concepts from continuous-time hidden Markov models (Baum and Petrie, 1966; Kouemou, 2011) and extends them to spatially-structured, high-dimensional processes. Interacting particle systems (IPSs) (Liggett, 1985; Lanchier, 2024) provide a powerful mathematical framework for describing local propagation dynamics in discrete state spaces and continuous time, and are an important subset of the broader class of continuous-time Markov chains (CTMCs). We formulate our goal as performing probabilistic inference on systems whose latent dynamics follow an IPS, given only incomplete or indirect information.

We aim to endow the IPS with a flexible parameterization (e.g., a neural network), yielding a discrete analog of latent/neural stochastic differential equations (Movellan et al., 2002; Tzen and Raginsky, 2019; Liu et al., 2020; Li et al., 2020; Bartosh et al., 2025). While neural-based inference methods for continuous-time discrete-state processes have been explored (Seifner and Sánchez, 2023; Berghaus et al., 2024), they have primarily demonstrated efficacy in low-dimensional settings.

The core technical challenge for inference with latent IPSs lies in sampling from a smoothed path measure over latent trajectories. To address this problem, we make the following contributions:

- We propose a twisted sequential Monte Carlo (tSMC) scheme (Guarniero et al., 2017; Heng et al., 2020) tailored to latent IPSs. We learn a twist function that can approximate the likelihood of future observations, and directly incorporate it into the rate matrix of an approximate posterior process, lifting the need to learn a separate proposal distribution (Lawson et al., 2018, 2022, 2023).
- We learn the twist function using a mass-covering Kullback-Leibler divergence loss in an amortized fashion (Zhao et al., 2024). Moreover, we design an efficient parameterization with favorable inductive biases.
- We demonstrate this on (i) a spatial susceptible–infected–recovered–susceptible (SIRS) model on graphs with up to 256 nodes, and (ii) a neural

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

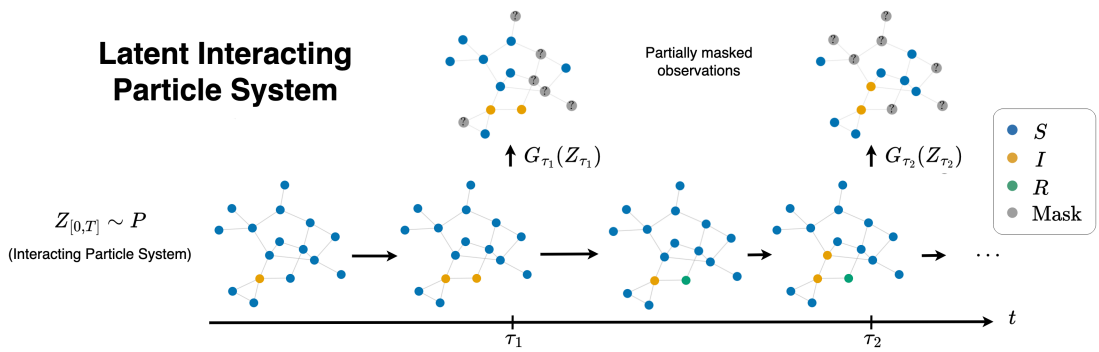

Figure 1: Example of a latent IPS as a state space model. Here, the latent trajectory is sampled from a continuous-time SIR model, and potentials are emission distributions of partially masked observations.

wildfire-spread model on 64×64 grids using the WildFireSpreadTS dataset (Gerard et al., 2023).

2 BACKGROUND

2.1 Continuous-time Markov chains

Consider a Markov process $Z_{[0,T]} \coloneqq \{Z_t\}_{t\in[0,T]}$ taking values in a discrete state space $\mathcal{Z} = \{1, \dots, V\}$ over a finite time horizon $[0, T]$. This system, known as a CTMC, is characterized by an initial distribution $p_0 \in \mathcal{P}(\mathcal{Z})$ and a measurable family of rate matrices $[R_t(z, \tilde{z})]_{z,\tilde{z}\in\mathcal{Z}}$, $t \in [0, T]$, where each entry satisfies

$$R_t(z, \tilde{z}) \coloneqq \lim_{\Delta t\to 0} \frac{1}{\Delta t} \mathbb{P}(Z_{t+\Delta t} = \tilde{z} \mid Z_t = z),$$

for $\tilde{z} \neq z$, and $R_t(z, z) \coloneqq -\sum_{\tilde{z}\neq z} R_t(z, \tilde{z})$ on the diagonal. Assuming $\sup_{t,\, z} -R_t(z, z) < \infty$ (non-explosion), sample paths are càdlàg, piecewise-constant, with finitely many jumps on $[0, T]$.

We refer to the induced distribution on the Skorokhod space $D([0, T], \mathcal{Z})$ as the *path measure*. Writing $0 < t_1 < \cdots < t_N < T$ for the jump times and z_t for the state at a time $t \in [0, T]$, the density of this measure with respect to counting measure on states and Lebesgue measure on jump times is

$$p_0(z_0) \left[\prod_{n=1}^{N} R_{t_n}(z_{t_{n-1}}, z_{t_n}) \right] \exp\left(\int_0^T R_t(z_t, z_t)\, dt \right).$$

A detailed introduction can be found in Norris (1998); Del Moral and Penev (2017). We provide an overview of inference with CTMCs in Appendix A.

2.2 Interacting particle systems

While CTMCs can be extended to high-dimensional systems, the size of the associated rate matrix increases exponentially with the number of dimensions, making exact inference intractable. We restrict attention to state spaces of the form $\mathcal{Z} = \mathcal{V}^d$, where $\mathcal{V} = \{1, \dots, V\}$ is a fixed *vocabulary*. We assume that transitions affect only a single coordinate (i.e., dimension) at a time, and that the rate for updating a coordinate i may depend on the current global state $z \in \mathcal{Z}$. The dependence of the rate on z is often specified through a graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, \dots, d\}$ indexes coordinates and $\mathcal{E}$ encodes neighborhood structure (e.g., spatial dependence) (Liggett, 1985).

To describe local updates, we use the shorthand

$$z^{i\to v} \;=\; (z^1, \dots, z^{i-1}, v, z^{i+1}, \dots, z^d)$$

for the configuration obtained from z by replacing the i-th coordinate with $v \in \mathcal{V}$. The local rate $r_{i,t}(v \mid z) \geq 0$ is defined as the instantaneous intensity of transitioning from z to $z^{i\to v}$, i.e.

$$R_t(z, z^{i\to v}) \;=\; r_{i,t}(v \mid z), \qquad v \neq z^i.$$

Under these assumptions, the global generator R_t decomposes as a sum of local generators:

$$R_t(z, \tilde{z}) = \sum_{i\in\mathcal{I}} r_{i,t}(\tilde{z}^i \mid z) \prod_{j\neq i} \delta_{z^j}(\tilde{z}^j), \tag{1}$$

reflecting the assumption that only one coordinate may change at a time. Typically the local rates $r_{i,t}(v \mid z)$ depend only on the neighborhood $(z^j)_{j\in\mathcal{N}(i)}$ of i in the graph $\mathcal{G}$. CTMCs with generators of this form are known as *interacting particle systems* (IPS) on a finite state space and finite graph (Liggett, 1985; Lanchier, 2017, 2024). We are interested in system that may exhibit time dependence, i.e. non-homogeneous processes (Norris, 1998).

Example *(SIR model)*: In an SIR model on a network of d nodes, the state space is $\{S, I, R\}^d$ and the rate of infection of each node can be a function of the

states of its neighbors. For an illustration, see the process $Z_{[0,T]}$ in Figure 1.

Simulation. Exact simulation from IPSs is feasible for models like the contact process and the voting process (Lanchier, 2017), leveraging a construction from independent Poisson processes (Lanchier, 2017). However, simulation can be challenging for non-homogeneous processes. For this reason, in this work we employ a simple first-order Euler discretization (Sun et al., 2023). We assume conditional independence among coordinates in intervals of width Δ_t, and let the transition probability be:

$$q_{t,\Delta_t}(\tilde{z} \mid z) \;:=\; \prod_{i\in\mathcal{I}} \big(\delta_{\tilde{z}^i, z^i} \;+\; \Delta_t r_{i,t}(\tilde{z}^i \mid z)\big)\,, \quad (2)$$

where $r_{i,t}(z^i \mid z) := -\sum_{v\neq z^i} r_{i,t}(v \mid z)$. Note that q_{t,Δ_t} may assign mass to multi-site flips for any finite step size. Despite its simplicity, this simulation scheme has shown good performance in generative modeling applications, such as simulating the reverse process of discrete diffusion and discrete flow matching models (Campbell et al., 2024; Gat et al., 2024).

In the following Proposition, we show that the error of this kernel can be bounded in total variation. Let $\lambda_t(z) := \sum_{i\in\mathcal{I}}\sum_{v\neq z^i} r_{i,t}(v \mid z)$. We assume that the following properties hold for each $t \in [0,T]$:

(A1) **Bounded total rate:** $\sup_{t,z} \lambda_t(z) \le \bar{\lambda} < \infty$.

(A2) **Small time interval:** we bound the step size as $\Delta_t \le 1/\sup_{t,z,i}\sum_{v\neq z^i} r_{i,t}(v\,|\,z)$.

(A3) **Local Lipschitz in time:** for all $u \in [0,\Delta_t]$, $\big|r_{i,t+u}(v \mid z) - r_{i,t}(v \mid z)\big| \le L\,u$.

Since $\mathcal{Z}$ is finite, we identify the transition kernel of an IPS $P_{t,\Delta_t}(z_t, dz) := \mathbb{P}(Z_{t+\Delta_t} \in dz \mid Z_t = z_t)$ with its pmf $p_{t,\Delta_t}(\tilde{z} \mid z) := P_{t,\Delta_t}(z, \{\tilde{z}\})$.

Proposition 1. *Under* (A1)–(A3)*, there exists* $C < \infty$ *such that*

$$\big\|q_{t,\Delta_t}(\,\cdot\,|\,z) - p_{t,\Delta_t}(\,\cdot\,|\,z)\big\|_{\mathrm{TV}} \;\le\; C\,\Delta_t^2,$$

uniformly in $t \in [0,T]$ *and* $z \in \mathcal{Z}$.

A proof is provided in Appendix B.1.

2.3 Sequential Monte Carlo

In our approximate posterior inference scheme for IPSs, we employ sequential Monte Carlo (SMC) methods. SMC approximates a terminal target π_T by evolving a population of weighted samples through a sequence of intermediate targets $\{\pi_t\}_{t\in[0,T]}$ (Doucet and Johansen, 2009; Naesseth et al., 2019; Chopin and Papaspiliopoulos, 2020). We assume π_0 is easy to sample from, and that we can evaluate unnormalized densities $\{\gamma_t\}_{t\in[0,T]}$ with $\pi_t \propto \gamma_t$. In our setting, it is natural to let π_t be a distribution on path prefixes $Z_{[0,t]} \in D([0,t],\mathcal{Z})$. Given particles $\{Z^{(s)}_{[0,t-\Delta_t]}, w^{(s)}_{t-\Delta_t}\}_{s=1}^S$ targeting $\pi_{t-\Delta_t}$ and a proposal kernel $q_t(\cdot \mid Z^{(s)}_{t-\Delta_t})$ that advances each trajectory by Δ_t, we draw a segment $Z^{(s)}_{(t-\Delta_t,t]} \sim q_t(\cdot \mid Z^{(s)}_{t-\Delta_t})$ and update unnormalized incremental weights as

$$\tilde{w}^{(s)}_t \;=\; \frac{\gamma_t\big(dZ^{(s)}_{[0,t]}\big)}{\gamma_{t-\Delta_t}\big(dZ^{(s)}_{[0,t-\Delta_t]}\big)\, q_t\big(Z^{(s)}_{(t-\Delta_t,t]} \mid Z^{(s)}_{t-\Delta_t}\big)}\,,$$

$$w^{(s)}_t \propto w^{(s)}_{t-\Delta_t}\,\tilde{w}^{(s)}_t, \quad \bar{w}^{(s)}_t = \frac{w^{(s)}_t}{\sum_{j=1}^S w^{(j)}_t}. \quad (3)$$

A distinctive feature of SMC is resampling, where samples are selected and propagated based on their importance weights. This is implemented by drawing ancestry indices according to $\bar{w}^{(1:S)}_t$, adaptively when the effective sample size $\mathrm{ESS}_t = \big(\sum_{s=1}^S (\bar{w}^{(s)}_t)^2\big)^{-1}$ falls below a threshold (Naesseth et al., 2019). Two design choices are central to performance: *(i) intermediate targets* $\{\pi_t\}$, which can be crafted to reduce variance of incremental weights; and *(ii) the proposal distribution* q_t, which should ensure proposed segments land in high-probability regions of π_t. Poor choices of either can trigger rapid weight collapse, especially in high-dimensional, sparse-observation regimes (Naesseth et al., 2019; Chopin et al., 2023).

3 EFFICIENT INFERENCE FOR LATENT INTERACTING PARTICLE SYSTEMS

3.1 Posterior path measure for an IPS

Let $Z_{[0,T]}$ be an Interacting Particle System (IPS) with path measure P, initial distribution p_0, and local transition rates $r_{i,t}(v\,|\,z)$. We are interested in conditioning this process on auxiliary information available at discrete times $\tau_1,\ldots,\tau_K \in [0,T]$. We represent this information by nonnegative *potential functions* $G_{\tau_k} : \mathcal{Z} \to \mathbb{R}_+$ that can be evaluated pointwise. The resulting *posterior path measure* is

$$P^\star(dZ_{[0,T]}) \;\propto\; \left(\prod_{k=1}^K G_{\tau_k}(Z_{\tau_k})\right) P(dZ_{[0,T]}), \quad (4)$$

a special case of a Feynman–Kac (FK) path measure (Chopin and Papaspiliopoulos, 2020; Lu and Wang, 2024; Park et al., 2025). A key property is that the process $Z^\star_{[0,T]}$ governed by $P^\star$ is itself an IPS.

This follows from Doob's h-transform applied to the original dynamics (Del Moral and Penev, 2017; Corstanje et al., 2023; Corstanje and Meulen, 2025). Define the *look-ahead function*

$$h_t^\star(z) \coloneqq \mathbb{E}_P\Big[\prod_{k:\tau_k>t} G_{\tau_k}(Z_{\tau_k}) \,\Big|\, Z_t = z\Big], \qquad h_T^\star(z) = 1, \tag{5}$$

which is right-continuous with left limits (càdlàg) with reset conditions at potential times (Eich et al., 2025):

$$h^\star_{\tau_k^-}(z) \;=\; G_{\tau_k}(z)\, h^\star_{\tau_k}(z). \tag{6}$$

Note that we can marginalize equation 4 to any time $t \in [0, T]$, and get

$$P^\star(dZ_{[0,t]}) \propto P(dZ_{[0,t]}) h_t^\star(Z_t) \prod_{k:\tau_k\le t} G_{\tau_k}(Z_{\tau_k}). \tag{7}$$

The initial distribution and local rates of the posterior IPS are characterized in the following Proposition.

Proposition 2 (Doob's h-transform for IPS)**.** *Under $P^\star$, the process $\{Z_t^\star\}$ is an IPS with initial distribution*

$$p_0^\star(z) \;\propto\; p_0(z)\, h_0^\star(z),$$

and off-diagonal local transition rates

$$r^\star_{i,t}(v \mid z) \;=\; r_{i,t}(v \mid z)\, s^\star_{i,t}(v, z), \qquad v \neq z^i, \tag{8}$$

where $s^\star_{i,t}(v, z) \coloneqq h_t^\star(z^{i\to v})/h_t^\star(z)$. Diagonal terms are defined as usual by $r^\star_{i,t}(z^i \mid z) = -\sum_{v\neq z^i} r^\star_{i,t}(v \mid z)$.

Note that this proposition follows from standard arguments on the conditioning of CTMCs, see (Huang et al., 2016; Corstanje et al., 2023; Corstanje and Meulen, 2025). We provide a complete proof of Proposition 2 in Appendix B.2. This formulation generalizes the posterior distribution used in reward-guided fine-tuning for discrete diffusion models (Li et al., 2025; Wang et al., 2025; Cheuk-Kit et al., 2025) to multiple potential times. Simulating directly from these transformed rates $r^\star_{i,t}$ is generally intractable because computing the look-ahead function $h_t^\star(z)$ requires solving a high-dimensional integral over future trajectories under P.

3.2 Latent IPS

Let $\{P_\theta : \theta \in \Theta\}$ denote a family of prior path measures on $D([0, T], \mathcal{Z})$, where each P_θ corresponds to an IPS with initial distribution p_0^θ and local transition rates $r^\theta_{i,t}(v \mid z)$.

Auxiliary information about the latent trajectory is available in the form of discrete-time observations

$$Y = (Y_{\tau_1}, \dots, Y_{\tau_K}), \qquad 0 < \tau_1 < \cdots < \tau_K \le T.$$

Here and throughout, we use θ to denote the parameters of the full generative model, including both the prior IPS dynamics and the observation model. For an observed trajectory $y_{1:K}$, potential functions in equation 4 correspond to emission distributions

$$G_{\tau_k,\theta}(z) \;=\; p_\theta(y_k \mid Z_{\tau_k} = z), \qquad k = 1, \dots, K,$$

where we omit Y_{τ_k} to simplify the notation. We denote the corresponding posterior path measure by $P_\theta^\star$, which, by Proposition 2, is again an IPS. We refer to this model as a *Latent Interacting Particle System (Latent IPS)*. Figure 1 illustrates this construction for a latent susceptible-infected-recovered (SIR) model on a graph.

Maximum likelihood. The logarithm of the marginal likelihood of the observations is

$$\mathcal{L}(\theta) \coloneqq \log \mathbb{E}_{P_\theta}\left[\prod_{k=1}^K G_{\tau_k,\theta}(Z_{\tau_k})\right]. \tag{9}$$

This objective can be optimized via gradient steps using Fisher's identity (see Appendix B.3), rewriting $\nabla_\theta \mathcal{L}(\theta)$ as

$$\mathbb{E}_{P_\theta^\star}\left[\sum_{k=1}^K \nabla_\theta \log G_{\tau_k,\theta}(Z_{\tau_k}) + \nabla_\theta \log P_\theta(Z_{[0,T]})\right]. \tag{10}$$

These updates are not directly tractable, as they require samples from $P_\theta^\star$. Viable options include optimizing an evidence lower bound (Hinton et al., 1995), importance sampling (Bornschein and Bengio, 2015), and SMC (Lawson et al., 2023; McNamara et al., 2024). Due to the sequential nature of latent IPSs, we opt for an SMC approach.

3.3 Twisted SMC for posterior inference in latent IPS

Classic design choices for SMC algorithms, such as the bootstrap particle filter (BPF) (Doucet and Johansen, 2009), notoriously display poor performance in continuous-time problems with sparse potentials, as weights are uniform in between potential times, leading to particle degeneracy (Chopin et al., 2023).

Twisted targets. Twisted SMC (tSMC) addresses this limitation by specifying a favorable choice for the intermediate target distributions of SMC (Guarniero et al., 2017; Heng et al., 2020). The core idea is to introduce a learnable function $h_t^\psi : \mathcal{Z} \to \mathbb{R}_+$, the *twist function*, approximating the look-ahead function $h_t^\star(z)$ from equation 5. The tSMC algorithm targets a

sequence of *twisted* distributions

$$P_\theta^\psi(dZ_{[0,t]}) \propto P_\theta(dZ_{[0,t]}) h_t^\psi(Z_t) \prod_{k:\tau_k \le t} G_{\tau_k,\theta}(Z_{\tau_k}), \tag{11}$$

for $t \in [0,T]$. Note that h_t^ψ can be a function of future observations $y_{>t}$, and we omit this dependence for notational simplicity.

Twist-induced proposal. We can approximately sample from this IPS using an approximation of a first-order Euler discretization as in 2.2, resulting in the twist-induced proposal distribution

$$q_{t,\Delta_t}^{\theta,\psi}(z \mid z_t) = \prod_{i\in\mathcal{I}} \Big(\delta_{z_t^i, z^i} + \Delta_t\, r_{i,t}^{\theta,\psi}(z^i \mid z_t)\Big), \tag{12}$$

$$r_{i,t}^{\theta,\psi}(z^i \mid z_t) := \begin{cases} r_{i,t}^\theta(z^i \mid z_t) s_{i,t}^\psi(z^i, z_t), & z^i \neq z_t^i, \\ -\sum_{v \neq z_t^i} r_{i,t}^\theta(v \mid z_t) s_{i,t}^\psi(v, z_t), & z^i = z_t^i, \end{cases} \tag{13}$$

where $s_{i,t}^\psi(z^i, z_t) := h_t^\psi(z_t^{i\to z^i})/h_t^\psi(z_t)$ is the concrete score (Meng et al., 2022). Note that the twist function h^ψ might not satisfy the reset conditions in equation 6. We can quantify the extent to which it is violated by introducing the *reset residual*

$$\rho_t(z) := \Big(\log h_{t^-}^\psi(z) - \log G_t(z) - \log h_t^\psi(z)\Big)\, \mathbf{1}_{t\in\{\tau_k\}}.$$

Let the incremental effective sample size (ESS) with respect to the target posterior be defined as

$$\mathrm{ESS}_{t,\Delta_t}(z_t) := \mathbb{E}_{q_{t,\Delta_t}^{\theta,\psi}(\cdot|z_t)} \left[\left(\frac{p_{t,\Delta_t}^{\theta,\star}(Z_{t+\Delta_t} \mid z_t)}{q_{t,\Delta_t}^{\theta,\psi}(Z_{t+\Delta_t} \mid z_t)} \right)^2 \right]^{-1}, \tag{14}$$

where

$$p_{t,\Delta_t}^{\theta,\star}(z \mid z_t) \propto p_{t,\Delta_t}^\theta(z \mid z_t) h_{t+\Delta_t}^\star(z) G_t(z)^{\mathbf{1}_{t\in\{\tau_k\}}}.$$

In the following Theorem, we show that the incremental ESS at each step can be lower bounded by composing terms depending on the approximation error of the twist function, the reset residual, and the discretization error.

Theorem 3. *For $t \in [0,T]$, bound the twist error and the reset residual by*

$$\varepsilon_t := \sup_{z\in\mathcal{Z}} \big|\log h_t^\star(z) - \log h_t^\psi(z)\big|, \quad \delta_t := \sup_{z\in\mathcal{Z}} |\rho_t(z)|.$$

Under (A1)-(A3) for the prior dynamics and assuming h^ψ to be Lipschitz continuous between observation times, there exists a $C < \infty$ such that the incremental ESS in equation 14 satisfies, uniformly in z_t,

$$\mathrm{ESS}_{t,\Delta_t}(z_t) \;\ge\; \frac{\exp(-4(\varepsilon_{t+\Delta_t} + \delta_{t+\Delta_t}))}{1 + C\Delta_t^2}.$$

For a proof, see Appendix B.4.

3.4 Efficient twist parameterization.

Parameterizing the twist–induced proposal in equation 12 is challenging because, at each step and for a given state z_t, we need all ratios

$$s_{i,t}^\psi(v, z_t) = \frac{h_t^\psi(z_t^{i\to v})}{h_t^\psi(z_t)}, \qquad i \in \mathcal{I},\ v \in \mathcal{V},$$

that is, the twist evaluated at every single–site modification $z_t^{i\to v}$ of z_t. A naïve implementation would call h_t^ψ separately on each configuration $z_t^{i\to v}$, requiring $O(dV)$ forward passes per time step. To make this structure explicit, define for each configuration $z \in \mathcal{Z}$ the matrix

$$H_t^\psi(z) \in \mathbb{R}_+^{d\times V}, \qquad H_t^\psi(z)_{[i,v]} := h_t^\psi\big(z^{i\to v}\big).$$

In particular, $H_t^\psi(z)_{[i,z^i]} = h_t^\psi(z)$ for all $i \in \mathcal{I}$. By construction, replacing the i-th coordinate of z by any label $u \in \mathcal{V}$ leaves the *row* $H_t^\psi(z)_{[i,\cdot]}$ unchanged:

$$H_t^\psi(z)_{[i,v]} = H_t^\psi\big(z^{i\to u}\big)_{[i,v]}, \qquad u, v \in \mathcal{V},\ i \in \mathcal{I}. \tag{15}$$

This is the functional identity we would like the learned mapping $z \mapsto H_t^\psi(z)$ to satisfy: each entry $H_t^\psi(z)_{[i,v]}$ should depend on z only through the context provided by the other coordinates and the choice of index–value pair (i, v), not on the current value z^i itself.

A direct implementation of $H_t^\psi(z)$ by explicitly forming all $z^{i\to v}$ would be prohibitively expensive. Instead, we directly exploit the product structure of the state space to build an efficient alternative. We first use a **context encoder** that can depend on any contextual information c_t we wish to make available for the model at time t, except for the state z_t itself:

$$\Phi_t(c_t) \in \mathbb{R}^{d\times V\times m}, \tag{16}$$

where c_t can include, for instance, future observations and observation times $(Y_{\ge t}, \tau_{\ge t})$, covariates, and positional information if available. We highlight that the encoder in equation 16 is independent of the current state z_t, and produces a tensor of embeddings associated with setting each coordinate i to a state v. Based on these embeddings, we can compose the output $h_t^\psi(z)$ by aggregating the representations at the positions indexed by our target state, via a map

$$\log h_t^\psi(z) = \phi\left(\sum_{i\in\mathcal{I}} \Phi_t(c_t)_{[i,z^i]}\right), \tag{17}$$

where $\phi : \mathbb{R}^m \to \mathbb{R}$ can be a learned map (e.g., a small MLP), a construction mirroring the DeepSets model (Zaheer et al., 2017). Let $S_t(z) := \sum_{j\in\mathcal{I}} \Phi_t(c_t)_{[j,z^j]}$

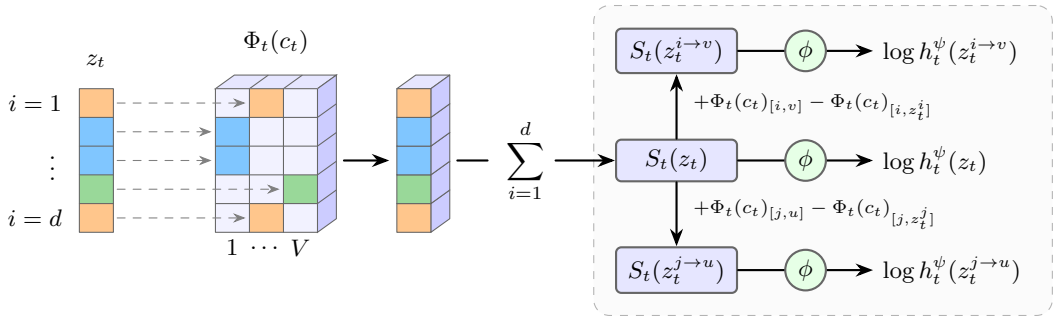

Figure 2: TwistNet uses a context encoder to produce a table of embeddings for every coordinate and every possible state value, independently of the current state z_t. The state z_t then selects one embedding per coordinate, and the selected embeddings are pooled into a single representation of the full state. To evaluate the twist at any single-site modification of z_t, TwistNet reuses this pooled representation and updates only the contribution of the modified coordinate, enabling parallel evaluation of all single-site neighboring states.

denote the pooled representation of the current configuration. For each single–site update $z^{i\to v}$, we can reuse $S_t(z)$ and adjust only the contribution of coordinate i:

$$\log h_t^{\psi}\left(z^{i\to v}\right) = \phi\Big(S_t(z) + \Phi_t(c_t)_{[i,v]} - \Phi_t(c_t)_{[i,z^i]}\Big), \tag{18}$$

an operation trivial to parallelize. Moreover, the identity in equation 15 holds exactly for this parameterization: replacing z^i by any u changes both $S_t(z)$ and the subtraction $-\Phi_t(c_t)_{[i,z^i]}$ in a way that cancels out, leaving the argument of ϕ (and thus $H_t^{\psi}(\cdot)_{[i,v]}$) invariant for every v.

We refer to this architecture, which (i) amortizes the cost of encoding contextual information across all (i, v) pairs and (ii) enforces the desired invariance of equation 15, as **TwistNet**. Figure 2 illustrates this construction: the context encoder computes $\Phi_t(c_t)$ once, the current state z_t selects one embedding per coordinate to form $S_t(z_t)$, and each single-site modification $z_t^{i\to v}$ is evaluated by replacing only the i-th contribution in the pooled representation.

Twist learning. Multiple objective functions have been proposed to learn the twist, from approaches based on consistency inspired by reinforcement learning (Heng et al., 2020; Lawson et al., 2018), to density ratio estimation (Lawson et al., 2022, 2023). Recently, Zhao et al. (2024) proposed to minimize the mass-covering forward KL divergence with respect to the true posterior in the context of autoregressive language models. Forward KL objectives had previously been proposed for learning the proposal distribution in SMC by Gu et al. (2015); Lawson et al. (2023). We adapt this approach to our setting, and learn an amortized twist function by optimizing $D_{\mathrm{KL}}(P_\theta^\star \,||\, P_\theta^\psi)$, which is proportional to

$$\mathbb{E}_{P_\theta^\star}\Bigg[\sum_{i\in\mathcal{I}}\Bigg(\int_0^T r_{i,t}^{\theta,\psi}(Z_t^i \mid Z_t)dt + \sum_{u:Z_u^i \neq Z_{u^-}^i} \log s_{i,u}^{\theta,\psi}(Z_u^i, Z_{u^-})\Bigg)\Bigg] + \mathbb{E}_{p_0^\star}\left[\log q_0^{\psi}(Z_0)\right], \tag{19}$$

where $s_{i,t}^{\theta,\psi}(v, z) = h_t^{\psi}(z^{i\to v})/h_t^{\psi}(z)$ is the concrete score, and h_t^{ψ} may depend on future observations $y_{>t}$.

Wake-sleep. In order to compute a tractable approximation to equation 19, we employ the strategy proposed in Zhao et al. (2024) and in *wake-sleep* algorithms (Hinton et al., 1995; Bornschein and Bengio, 2015; Le et al., 2020; McNamara et al., 2024), and perform ancestral sampling of the trajectory from the prior, and of the observations from the emission distribution. We obtain the time-discretized *sleep objective*

$$\mathcal{L}_{\mathrm{s}}(\psi;\, z_{t_0:t_M}, y_{1:K}, \theta) = \tag{20}$$

$$-\log q_0^{\psi}(z_0 \mid y_{1:K}) - \sum_{m=0}^{M-1}\sum_{i\in\mathcal{I}}\Big[\Delta_{t_{m+1}}\, r_{i,t_m}^{\theta,\psi}(z_{t_m}^i \mid z_{t_m}) + \mathbf{1}[z_{t_m}^i \neq z_{t_{m+1}}^i]\log s_{i,t_m}^{\psi}(z_{t_{m+1}}^i, z_{t_m})\Big], \tag{21}$$

where $0 = t_0 < t_1 < \cdots < t_M = T$ is a time grid containing $\{\tau_k\}_{k=1}^K$ with $\Delta_{t_m} = t_m - t_{m-1}$, and q_0^{ψ} is an approximate posterior initial distribution. Note that, although omitted for notational simplicity, s_{i,t_m}^{ψ} is conditioned on future observations $y_{>t_m}$. We optimize this objective with respect to the parameters ψ, while holding θ fixed. Note that optimizing equation 20 does not require backpropagating through the sampled trajectories, avoiding the need to use reparameterization (Jang et al., 2017) or REINFORCE (Williams, 1992).

We alternate $\psi-$updates on the objective in equation 20 with $\theta-$updates approximating equation 10 based on

Algorithm 1 Wake–sleep with tSMC for latent IPS

Inputs: Dataset $\mathcal{D}$; time grid $0 = t_0 < \cdots < t_M = T$; particles S; batch size B
Initialize θ, ψ
repeat
  Sleep phase: *# update ψ, hold θ fixed*
  for $b = 1, \ldots, B$ **do**
    Simulate $z^{(b)}_{t_0:t_M} \sim P_\theta$ via Euler steps
    Simulate $\tilde{y}^{(b)}_k \sim p_\theta(\cdot \mid z^{(b)}_{\tau_k}), \quad k = 1, \ldots, K$
    Compute $\ell^{(b)}_{\mathrm{s}} = \mathcal{L}_{\mathrm{s}}(\psi; z^{(b)}_{t_0:t_M}, \tilde{y}^{(b)}_{1:K}, \theta)$ via equation 20
  end for
  $\psi \leftarrow \textsc{GradStep}_\psi\Big(\nabla_\psi \frac{1}{B}\sum_{b=1}^{B} \ell^{(b)}_{\mathrm{s}}\Big)$
  Wake phase: *# update θ, hold ψ fixed*
  for $b = 1, \ldots, B$ **do**
    Sample $(y^{(b)}_{1:K}, \tau^{(b)}_{1:K}) \sim \mathcal{D}$
    Run $\{z^{(b,s)}_{t_0:t_M}, \bar{w}^{(b,s)}_T\}_{s=1}^{S} \leftarrow \mathrm{tSMC}(P^{\psi}_{\theta}; y^{(b)}_{1:K}, \tau^{(b)}_{1:K})$
    Compute $g^{(b)}_{\mathrm{w}} = \widehat{\nabla_\theta \mathcal{L}}^{(b)}_{\mathrm{w}}(\theta; \{z^{(b,s)}_{t_0:t_M}, \bar{w}^{(b,s)}_T\}_{s=1}^{S}, y^{(b)}_{1:K})$ via equation 23
  end for
  $\theta \leftarrow \textsc{GradStep}_\theta\Big(\frac{1}{B}\sum_{b=1}^{B} g^{(b)}_{\mathrm{w}}\Big)$
until convergence

trajectories sampled using tSMC, similarly to Lawson et al. (2023) and McNamara et al. (2024). For the θ-updates, we use the discretized log path density of the prior

$$\log \hat{P}_\theta(z_{[0,T]}) = \log p^{\theta}_0(z_0) + \sum_{m=0}^{M-1}\sum_{i\in\mathcal{I}}\Big[\Delta_{t_{m+1}} r^{\theta}_{i,t_m}(z^i_{t_m} \mid z_{t_m}) + \mathbf{1}[z^i_{t_m} \neq z^i_{t_{m+1}}] \log r^{\theta}_{i,t_m}(z^i_{t_{m+1}} \mid z_{t_m})\Big], \quad (22)$$

to form the wake gradient estimator

$$\widehat{\nabla_\theta \mathcal{L}_{\mathrm{w}}}\big(\theta; \{z^{(s)}_{t_0:t_M}, \bar{w}^{(s)}_T\}_{s=1}^{S}, y_{1:K}\big) = \quad (23)$$

$$\sum_{s=1}^{S} \bar{w}^{(s)}_T \left[\sum_{k=1}^{K} \nabla_\theta \log G_{\tau_k,\theta}(z^{(s)}_{\tau_k}) + \nabla_\theta \log \hat{P}_\theta(z^{(s)}_{[0,T]})\right], \quad (24)$$

where $\bar{w}^{(s)}_T$ are SMC weights as in equation 3. We note that the estimator in equation 23 is asymptotically consistent for the gradient of the maximum likelihood objective (Lawson et al., 2023). Simplified pseudocode is presented in Algorithm 1, and the detailed training algorithm and its complexity are reported in Appendix C.

4 EXPERIMENTS

4.1 Latent SIRS model

We study inference in a latent spatial SIRS IPS on graphs with state space $\mathcal{Z} = \{S, I, R\}^d$ (Lanchier, 2024). We simulate a dataset of trajectories using Gillespie's algorithm (Gillespie, 1977; Wilkinson, 2018) with the following local rates:

$$r_{i,t}(I \mid z) = \big(\alpha_0 + \alpha_1 \textstyle\sum_{j\neq i} a_{ij}\, \sigma(\langle \xi_i, \xi_j\rangle)\, \delta_{z^j,I}\big)\, \delta_{z^i,S},$$
$$r_{i,t}(R \mid z) = \beta\, \delta_{z^i,I}, \qquad r_{i,t}(S \mid z) = \gamma\, \delta_{z^i,R}, \quad (25)$$

where a_{ij} is an adjacency matrix sampled from a graph with fixed expected degree, ξ_i are normalized features acting as edge weights, and σ is a logistic function ensuring non-negativity. We let observation times be sampled uniformly in $[0, T]$ at $K = 10$ irregular *snapshots* $\tau_1 < \cdots < \tau_K$, and we sample observations from an emission distribution masking each node (denoted $y_k = \varnothing$) independently with probability $p_{\mathrm{mask}} = 0.5$.

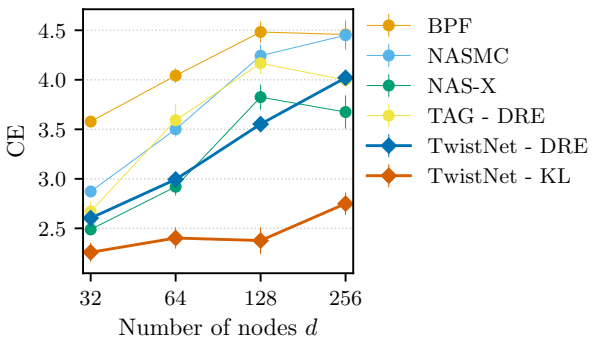

Figure 3: Latent trajectory reconstruction, measured by cross-entropy loss on the test set of ground truth trajectories with respect to the posterior approximations. Error bars correspond to two standard errors computed across trajectories.

Task. We conduct two experiments. First, we perform *latent trajectory inference* (Eich et al., 2025): given observations $y_{1:K}$ and parameters θ at their true values, we measure how accurately our approximate posterior samples match the ground truth trajectory. We focus on scalability with respect to graph dimensionality, quantifying performance via categorical cross-entropy loss averaged over a discretized time grid. Second, we *estimate parameters* from an arbitrary initialization using only 50 training trajectories. Following Section 3.4, we employ a *wake-sleep* algorithm and evaluate accuracy via individual parameter estimates and total relative parameter error $\sum_j |\hat{\theta}_j - \theta_j|/|\theta_j|$ over the four parameters in equation 25.

Methods. For our TwistNet parameterization, we experiment learning the twist using both our forward KL loss from equation 20 and the DRE loss introduced by (Lawson et al., 2022), using a graph transformer as context encoder and a two-layer MLP as ρ in equation 17. We consider NASMC (Gu et al., 2015) and

Table 1: Parameter estimates and relative parameter error (RPE) for the SIRS model with 32 nodes, mean ± 2 standard deviations across 10 random seeds.

	α_0	α_1	β	γ	RPE
Ground truth	0.1	1.0	0.4	0.05	–
TwistNet - KL	**0.113** ± 0.03	0.922 ± 0.08	0.393 ± 0.03	0.046 ± 0.01	**0.330** ± 0.35
TwistNet - DRE	0.115 ± 0.03	**0.957** ± 0.15	0.384 ± 0.03	**0.048** ± 0.01	0.360 ± 0.34
TAG - DRE	0.124 ± 0.05	0.673 ± 0.43	0.377 ± 0.14	0.055 ± 0.02	0.982 ± 0.68
NAS-X	0.136 ± 0.04	0.880 ± 0.10	**0.405** ± 0.04	0.058 ± 0.01	0.673 ± 0.39
NASMC	0.076 ± 0.09	0.936 ± 0.26	0.444 ± 0.11	0.061 ± 0.04	0.962 ± 0.56
BPF	0.791 ± 1.90	1.354 ± 2.05	0.618 ± 0.24	0.001 ± 0.00	9.461 ± 17.88

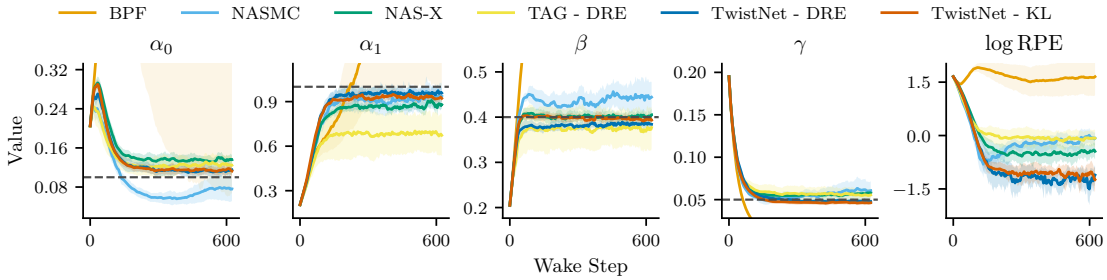

Figure 4: Evolution of the parameters and log relative parameter error (log RPE) through wake steps updating the parameters θ, for a SIRS model on a graph with 32 nodes.

NAS-X (Lawson et al., 2023), and parameterize the proposal distribution with a graph transformer as the context encoder used in TwistNet, taking the current state as input and predicting a concrete score field. The twist for NAS-X follows the same parameterization, with output pooled to a scalar. We experiment using TAG (Nisonoff et al., 2025) with DRE loss, and a regular BPF (Doucet and Johansen, 2009). We discuss further details in Appendix D.1 and D.2.

Results. As illustrated in Figure 3, we find the combination of TwistNet and the forward KL loss in equation 19 to scale the best with respect to dimensions for latent trajectory inference. We report results on parameter estimation on a graph with 32 nodes in Table 1, and display convergence through wake steps in Figure 4. Additional results are reported in Appendix E.1.

4.2 Wildfire trajectories

Forecasting wildfire spread is a problem of critical importance, given the impacts of wildfires and their increasing severity globally in recent years (Abatzoglou and Williams, 2016; Walker et al., 2019). Physics-based models have long existed (Rothermel, 1972) but are difficult to calibrate given real-world constraints, limiting forecasting utility. With the increasing availability of satellite data, ML approaches for spatio-temporal prediction have emerged (e.g., Coffield et al. (2019); Prapas et al. (2021); Apostolakis et al. (2022); Gerard et al. (2023)), yet they can lack physical insight (e.g., local propagation constraints) and have not yet seen widespread application in real-world scenarios. A third class of model in this context is based on the notion of local spreading of fire (across pixels) via stochastic cellular automata (SCAs), closely related to our latent IPS framework. Prior work in developing SCAs for wildfires has largely focused on relatively simple discrete-time models (Grieshop and Wikle, 2024), often hand-tuned (Clarke et al., 1994; Hargrove et al., 2000).

We model wildfires as latent IPSs, and consider a state space where each pixel in a 64×64 grid can be unburned (U), active (A), or burned (B), i.e. $\mathcal{Z} = \{U, A, B\}^{64\times 64}$. We let the graph be a lattice with Moore neighborhood. We parameterize outgoing, off-diagonal local rates by a single neural network F_θ mapping to $\mathbb{R}_+^{d\times(V-1)}$, using the UTAE architecture (Garnot and Landrieu, 2021). We constrain the dynamics by zeroing out the rate of flipping to A whenever no neighbors are ignited, and do not allow transitions from U to B and vice versa.

Task. Using the WildFireSpreadTS dataset (Gerard et al., 2023), we extract 156 week-long trajectories on 64×64 grids where at least one pixel is initially active with two subsequent days of activity, split 130-26 into a train and test set. We evaluate reconstruction of active fire maps at observation times (i.e., conditioning on the whole sequence) and prediction (initialized by encoding

Table 2: Binary cross-entropy loss for reconstruction and prediction of active fire maps, mean ± 2 standard errors across test trajectories.

	Full-week covariates		Past covariates only	
	Reconstruction	**Prediction**	**Reconstruction**	**Prediction**
TwistNet - KL	**0.877** ± 0.36	**1.149** ± 0.41	**1.046** ± 0.389	**1.424** ± 0.499
TwistNet - DRE	1.985 ± 0.59	1.541 ± 0.52	1.702 ± 0.532	1.649 ± 0.536
NAS-X	2.392 ± 0.72	1.755 ± 0.53	1.787 ± 0.531	1.576 ± 0.523
NASMC	2.008 ± 0.60	1.759 ± 0.57	3.350 ± 0.897	1.992 ± 0.563

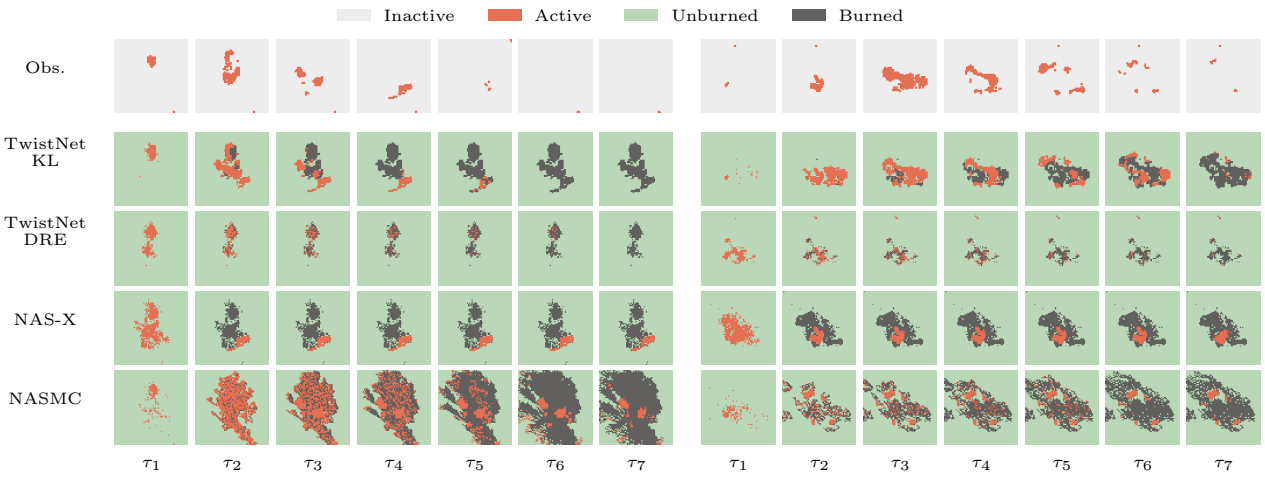

Figure 5: Observed active fire maps from WildFireSpreadTS (Gerard et al., 2023) and snapshots of approximate posterior samples at observation times, for the *full-week covariate* regime. A pixel can only turn active if another one in its Moore neighborhood is, and transitions from unburned to burned are impossible. Observations are noisy, as satellite imagery for wildfires can be obscured by smoke and clouds (Schroeder et al., 2014).

only the first observation), both measured via binary cross-entropy from the empirical distribution given by 16 particles per datapoint, averaged across time. We report results in two regimes: conditioning on VIIRS reflectance channels for the full week, and conditioning only on channels observed up to each timepoint.

Methods. We use the UTAE architecture for the context encoder in TwistNet, and for proposal and twist in NAS-X and NASMC. We train initial distributions – both posterior and prior – using a smaller model with the same architecture. We let our emission distribution be factorized, and let $p(y_k^i = 1 \mid Z_{\tau_k}^i = A) = \sigma(\theta_{\text{detect}})$, where θ_{detect} is a scalar logit and σ the logistic function. Due to their poor performance, we do not compare with TAG and BPF. For each method we also train an encoder for the first observation, used in the prediction task, by minimizing a forward KL loss in the *sleep* phase. We report experimental details in Appendix D.3.

Results. Table 2 reports results both with full-week covariates and with covariates restricted to those available up to each timepoint. TwistNet with the KL loss achieves the lowest average error in both regimes. By examining trajectories from the second best model (TwistNet with a DRE loss) in the full-week regime in Figure 5, we see a large gap in terms of how natural the trajectories look, and especially in how well the approximate posterior can bridge between observed states. Additional results are reported in Appendix E.2.

5 CONCLUSIONS

We introduced an efficient posterior inference framework for systems of latent interacting CTMCs, and empirically demonstrated its effectiveness on challenging tasks on latent state inference and parameter learning for both simulated and real data. Our proposed approach outperforms the baselines, and in our experiments we saw this performance gap increase with the dimensionality of the problem under consideration. A major bottleneck in the training of our method is the cost of simulating trajectories with twisted SMC. Recent advances in latent SDEs have shown that *simulation-free* algorithms are possible (Bartosh et al., 2025; Kiyohara et al., 2025), and we believe adapting these methods to latent IPSs is an exciting avenue for future work.

Acknowledgements

We thank the reviewers for their feedback on improving the paper. This work was supported by the National Science Foundation under awards NSF 2505006 and NSF 2425932, by the National Institutes of Health under awards R01-LM013344 and R01CA297869, by the Hasso Plattner Institute (HPI) Research Center in Machine Learning and Data Science at UCI, and by funding support from Google and from SAP.

References

John T Abatzoglou and A Park Williams. Impact of anthropogenic climate change on wildfire across western us forests. *Proceedings of the National Academy of Sciences*, 113(42):11770–11775, 2016.

Alexis Apostolakis, Stella Girtsou, Giorgos Giannopoulos, Nikolaos S Bartsotas, and Charalampos Kontoes. Estimating next day's forest fire risk via a complete machine learning methodology. *Remote Sensing*, 14 (5):1222, 2022.

Tomàs Artés, Duarte Oom, Daniele De Rigo, Tracy Houston Durrant, Pieralberto Maianti, Giorgio Libertà, and Jesús San-Miguel-Ayanz. A global wildfire dataset for the analysis of fire regimes and fire behaviour. *Scientific Data*, 6(1):296, 2019.

Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. SDE matching: Scalable and simulation-free training of latent stochastic differential equations. In *International Conference on Machine Learning*, pages 3054–3070. PMLR, 2025.

Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37 (6):1554–1563, 1966.

David Berghaus, Kostadin Cvejoski, Patrick Seifner, César Ojeda, and Ramses J Sanchez. Foundation inference models for Markov jump processes. *Advances in Neural Information Processing Systems*, 37: 129407–129442, 2024.

Mogens Bladt and Michael Sørensen. Statistical inference for discretely observed Markov jump processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(3):395–410, 2005.

Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. *International Conference on Learning Representations*, 2015.

Richard J Boys, Darren J Wilkinson, and Thomas BL Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18:125–135, 2008.

Alexandre Bureau, Stephen Shiboski, and James P Hughes. Applications of continuous time hidden Markov models to the study of misclassified disease outcomes. *Statistics in Medicine*, 22(3):441–462, 2003.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, pages 5453–5512, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583, 2018.

Lee Cheuk-Kit, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential Monte Carlo, February 2025. arXiv:2502.06079 [cs].

Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer International Publishing, Cham, 2020.

Nicolas Chopin, Andras Fulop, Jeremy Heng, and Alexandre H Thiery. Computational Doob h-transforms for online filtering of discretely observed diffusions. In *International Conference on Machine Learning*, pages 5904–5923. PMLR, 2023.

Keith C Clarke, James A Brass, and Philip J Riggan. A cellular automation model of wildfire propagation and extinction. *Photogrammetric Engineering and Remote Sensing*, 60(11):1355–1367, 1994.

Shane R. Coffield, Casey A. Graff, Yang Chen, Padhraic Smyth, Efi Foufoula-Georgiou, and James T. Randerson. Machine learning to predict final fire size at the time of ignition. *International Journal of Wildland Fire*, 28(11):861–873, 2019.

Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. Mean field variational approximation for continuous-time Bayesian networks. *The Journal of Machine Learning Research*, 11:2745–2783, 2010.

Marc Corstanje and Frank van der Meulen. Guided simulation of conditioned chemical reaction networks. *Statistical Inference for Stochastic Processes*, 28(2): 8, 2025.

Marc Corstanje, Frank van der Meulen, and Moritz Schauer. Conditioning continuous-time Markov processes by guiding. *Stochastics*, 95(6):963–996, 2023.

Pierre Del Moral and Spiridon Penev. *Stochastic Processes: From Applications to Theory*. Chapman and Hall/CRC, 2017.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *The Oxford Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

Yannick Eich, Bastian Alt, and Heinz Koeppl. Entropic matching for expectation propagation of Markov jump processes. In *The 28th International Conference on Artificial Intelligence and Statistics*, pages 856–864. PMLR, 2025.

Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4872–4881, 2021.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385, 2024.

Sebastian Gerard, Yu Zhao, and Josephine Sullivan. WildfirespreadTS: A dataset of multi-modal time series for wildfire spread prediction. *Advances in Neural Information Processing Systems*, 36:74515–74529, 2023.

Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

Andrew Golightly and Chris Sherlock. Efficient sampling of conditioned Markov jump processes. *Statistics and Computing*, 29:1149–1163, 2019.

Nicholas Grieshop and Christopher K Wikle. Data-driven modeling of wildfire spread with stochastic cellular automata and latent spatio-temporal dynamics. *Spatial Statistics*, 59:100794, 2024.

Shixiang Shane Gu, Zoubin Ghahramani, and Richard E Turner. Neural adaptive sequential Monte Carlo. *Advances in Neural Information Processing Systems*, 28:2629–2637, 2015.

Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112 (520):1636–1647, 2017.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

William W Hargrove, RH Gardner, MG Turner, WH Romme, and DG Despain. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135(2-3):243–263, 2000.

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential Monte Carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214): 1158–1161, 1995.

Asger Hobolth and Eric A Stone. Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. *The Annals of Applied Statistics*, 3(3):1204, 2009.

Peter Holderrieth, Michael Samuel Albergo, and Tommi Jaakkola. Leaps: A discrete neural sampler via locally equivariant networks. In *International Conference on Machine Learning*, pages 23397–23416. PMLR, 2025.

Lirong Huang, Loic Pauleve, Christoph Zechner, Michael Unger, Anders S Hansen, and Heinz Koeppl. Reconstructing dynamic molecular states from single-cell time series. *Journal of The Royal Society Interface*, 13(122):20160533, 2016.

Ilia Igashov, Arne Schneuing, Marwin Segler, Michael Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with Markov bridges. *arXiv preprint arXiv:2308.16212*, 2023.

Christopher Jackson. Multi-state models for panel data: the msm package for R. *Journal of Statistical Software*, 38:1–28, 2011.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Naoki Kiyohara, Edward Johns, and Yingzhen Li. Neural stochastic flows: Solver-free modelling and inference for SDE solutions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Achim Klenke. *Probability Theory: A Comprehensive Course*. Springer, 2008.

Lukas Köhs, Bastian Alt, and Heinz Koeppl. Variational inference for continuous-time switching dynamical systems. *Advances in Neural Information Processing Systems*, 34:20545–20557, 2021.

Guy Leonard Kouemou. History and theoretical basics of hidden Markov models. In *Hidden Markov Models, Theory and Applications*. IntechOpen, 2011.

Nicolas Lanchier. *Stochastic Modeling*. Springer, 2017.

Nicolas Lanchier. *Stochastic Interacting Systems in Life and Social Sciences*, volume 5. Walter de Gruyter GmbH & Co KG, 2024.

Dieterich Lawson, George Tucker, Christian A Naesseth, Chris Maddison, Ryan P Adams, and Yee Whye Teh. Twisted variational sequential Monte Carlo. In *Third Workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

Dieterich Lawson, Allan Raventós, Andrew Warrington, and Scott Linderman. SIXO: Smoothing inference with twisted objectives. *Advances in Neural Information Processing Systems*, 35:38844–38858, 2022.

Dieterich Lawson, Michael Li, and Scott Linderman. NAS-X: Neural adaptive smoothing via twisting. *Advances in Neural Information Processing Systems*, 36: 8602–8633, 2023.

Tuan Anh Le, Adam R Kosiorek, N Siddharth, Yee Whye Teh, and Frank Wood. Revisiting reweighted wake-sleep for models with stochastic control flow. In *Uncertainty in Artificial Intelligence*, pages 1039–1049. PMLR, 2020.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

Thomas Milton Liggett. *Interacting Particle Systems*, volume 2. Springer, 1985.

Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. How does noise help robustness? explanation and exploration under the neural sde framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 282–290, 2020.

Yu-Ying Liu, Shuang Li, Fuxin Li, Le Song, and James M Rehg. Efficient learning of continuous-time hidden Markov models for disease progression. *Advances in Neural Information Processing Systems*, 28:3600–3608, 2015.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Jianfeng Lu and Yuliang Wang. Guidance for twisted particle filter: a continuous-time perspective. *arXiv preprint arXiv:2409.02399*, 2024.

Robert T McGibbon and Vijay S Pande. Efficient maximum likelihood parameterization of continuous-time Markov processes. *The Journal of Chemical Physics*, 143(3), 2015.

Declan McNamara, Jackson Loper, and Jeffrey Regier. Sequential Monte Carlo for inclusive KL minimization in amortized variational inference. In *International Conference on Artificial Intelligence and Statistics*, pages 4312–4320. PMLR, 2024.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

Javier R Movellan, Paul Mineiro, and Ruth J Williams. A Monte Carlo EM approach for partially observable diffusion processes: theory and applications to neural networks. *Neural Computation*, 14(7):1507–1544, 2002.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR, 2018.

Christian A Naesseth, Fredrik Lindsten, and Thomas B Schön. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3): 187–306, 2019.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *The Thirteenth International Conference on Learning Representations*, 2025.

James R Norris. *Markov Chains*. Cambridge University Press, 1998.

Manfred Opper and Guido Sanguinetti. Variational inference for Markov jump processes. *Advances in Neural Information Processing Systems*, 20:1105–1112, 2007.

Byoungwoo Park, Hyungi Lee, and Juho Lee. Amortized control of continuous state space Feynman-Kac

model for irregular time series. In *International Conference on Learning Representations (ICLR 2025)*, 2025.

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

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 3942–3951, 2018.

Ioannis Prapas, Spyros Kondylatos, Ioannis Papoutsis, Gustau Camps-Valls, Michele Ronco, Miguel-Ángel Fernández-Torres, Maria Piles Guillem, and Nuno Carvalhais. Deep learning methods for daily wildfire danger forecasting. *arXiv preprint arXiv:2111.02736*, 2021.

Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and extensions. *Journal of Machine Learning Research*, 14:3295–3320, 2013.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael M Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth International Conference on Learning Representations*, 2024.

Richard C Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest & Range Experiment Station, Forest Service, 1972.

Wilfrid Schroeder, Patricia Oliva, Louis Giglio, and Ivan A Csiszar. The new VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, 143:85–96, 2014.

Patrick Seifner and Ramsés J Sánchez. Neural Markov jump processes. In *International Conference on Machine Learning*, pages 30523–30552. PMLR, 2023.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv:1905.09883*, 2019.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Xanthe J Walker, Jennifer L Baltzer, Steven G Cumming, Nicola J Day, Christopher Ebert, Scott Goetz, Jill F Johnstone, Stefano Potter, Brendan M Rogers, Edward AG Schuur, et al. Increasing wildfires threaten historic carbon sink of boreal forest soils. *Nature*, 572(7770):520–523, 2019.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hangchen Wang, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *The Thirteenth International Conference on Learning Representations*, 2025.

Christian Wildner and Heinz Koeppl. Moment-based variational inference for Markov jump processes. In *International Conference on Machine Learning*, pages 6766–6775. PMLR, 2019.

Darren J Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. Deep sets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3394–3404, 2017.

Boqian Zhang, Jiangwei Pan, and Vinayak A Rao. Collapsed variational Bayes for Markov jump processes. *Advances in Neural Information Processing Systems*, 30:3752–3760, 2017.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *International Conference on Machine Learning*, pages 60704–60748. PMLR, 2024.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. MDNS: Masked diffusion neural sampler via stochastic optimal control. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Efficient Inference for Coupled Hidden Markov Models in Continuous Time and Discrete Space: Supplementary Materials

A Background

Inference for CTMCs. Inference methods for CTMCs have been extensively studied. Maximum likelihood estimation for time-homogeneous CTMCs is discussed in Jackson (2011); Bladt and Sørensen (2005); McGibbon and Pande (2015). Expectation-maximization techniques for continuous-time hidden Markov models can be found in Bureau et al. (2003); Jackson (2011); Liu et al. (2015). Bayesian approaches include Markov chain Monte Carlo methods (Boys et al., 2008; Hobolth and Stone, 2009; Rao and Teh, 2013) and variational methods. The latter include mean-field (Opper and Sanguinetti, 2007; Cohn et al., 2010), moment-based methods (Wildner and Koeppl, 2019), combinations with MCMC (Zhang et al., 2017), and extensions to hybrid processes (Köhs et al., 2021). More recent methods include black-box variational inference with neural networks (Seifner and Sánchez, 2023), foundation models (Berghaus et al., 2024), and expectation propagation (Eich et al., 2025).

Related work. Another directly related line of research focuses on simulation methods for Markov bridges, notably Hobolth and Stone (2009); Huang et al. (2016); Golightly and Sherlock (2019); Corstanje et al. (2023); Corstanje and Meulen (2025). While less directly related, it is worth noting recent work discrete flow matching and diffusion methods based on CTMCs (Sun et al., 2023; Meng et al., 2022; Campbell et al., 2022; Igashov et al., 2023; Lou et al., 2023; Campbell et al., 2024), as well as discrete neural samplers (Holderrieth et al., 2025; Zhu et al., 2025). In particular, reward-guided generation for discrete diffusion models targets the posterior path measure of a CTMC, given by trajectories from a prior generative model tilted by a reward function at the endpoint T (Wang et al., 2025; Rector-Brooks et al., 2024), see (Uehara et al., 2025) for a review. While these methods have shown remarkable performance on posterior inference tasks, they are specific to the processes considered by discrete diffusion models.

B Proofs

B.1 Proof of Proposition 1

By the functional form of the rates, reaching any $\tilde{z}$ with at least two coordinates changed requires at least two jumps. By (A1) the jump times are dominated by an homogeneous Poisson process of rate $\bar{\lambda}$, hence on $[t, t+\Delta_t]$

$$\mathbb{P}\big(\geq 2 \text{ jumps}\big) \;\leq\; \mathbb{P}\big(U \geq 2\big) \;=\; O(\Delta_t^2), \quad U \sim \text{Poisson}(\bar{\lambda}\Delta_t).$$

Therefore,

$$\sum_{\tilde{z}:\, |\{i : \tilde{z}^i \neq z^i\}| \geq 2} p_{t,\Delta_t}(\tilde{z} \mid z) = O(\Delta_t^2).$$

As a next step, considering a starting state z, we partition the state space $\mathcal{Z}$ into three cases and denote them as $S_0 = \{z\}$ (no change), $S_1 = \{z^{i\to v} : i \in \mathcal{I},\ v \neq z^i\}$ (single flips), and $S_{\geq 2} = \{\tilde{z} : |\{i : \tilde{z}^i \neq z^i\}| \geq 2\}$ (multi flips). By (A3), for any (t, z, i, v),

$$\left|\Delta_t\, r_t^i(v \mid z) - \int_t^{t+\Delta_t} r_u^i(v \mid z)\, du\right| \leq \int_t^{t+\Delta_t} \big|r_t^i(v \mid z) - r_u^i(v \mid z)\big| du \leq \int_t^{t+\Delta_t} L(u-t) du = \frac{L}{2}\,\Delta_t^2,$$

and similarly

$$\left|\Delta_t\, \lambda_t(z) - \int_t^{t+\Delta_t} \lambda_u(z)\, du\right| \leq \frac{L_\lambda}{2}\,\Delta_t^2, \qquad L_\lambda := d \times (V-1) \times L.$$

Hence the following first–order Taylor expansions holds with uniform $O(\Delta_t^2)$ remainders:

$$k_{t,\Delta_t}(z' \mid z) = \begin{cases} 1 - \int_t^{t+\Delta_t} \lambda_u(z)du + O(\Delta_t^2) = 1 - \lambda_t(z)\,\Delta_t \;+\; O(\Delta_t^2) & z' \in S_0 \\ \int_t^{t+\Delta_t} r_u^i(v \mid z)du \;+\; O(\Delta_t^2) = r_t^i(v \mid z)\,\Delta_t \;+\; O(\Delta_t^2) & z' \in S_1 \\ O(\Delta_t^2) & z' \in S_{\geq 2} \end{cases}$$

Expanding equation 2:

$$q_{t,\Delta_t}(z' \mid z) = \begin{cases} 1 - \lambda_t(z)\,\Delta_t \;+\; O(\Delta_t^2) & z' \in S_0 \\ r_t^i(v \mid z)\,\Delta_t \;+\; O(\Delta_t^2) & z' \in S_1 \\ O(\Delta_t^2) & z' \in S_{\geq 2} \end{cases}$$

Recall $\|\mu - \nu\|_{\mathrm{TV}} = \frac{1}{2}\sum_{z'} |\mu(z') - \nu(z')|$. Then,

$$\sum_{z' \in S_0} \left|k_{t,\Delta_t}(z' \mid z) - q_{t,\Delta_t}(z' \mid z)\right| = \left|k_{t,\Delta_t}(z \mid z) - q_{t,\Delta_t}(z \mid z)\right| \;\leq\; C_0\,\Delta_t^2,$$

$$\sum_{z' \in S_1} \left|k_{t,\Delta_t}(z' \mid z) - q_{t,\Delta_t}(z' \mid z)\right| \leq \sum_i \sum_{v \neq z^i} C_1\,\Delta_t^2 \;=\; d\,(|\mathcal{V}| - 1)\,C_1\,\Delta_t^2,$$

$$\sum_{z' \in S_{\geq 2}} \left|k_{t,\Delta_t}(z' \mid z) - q_{t,\Delta_t}(z' \mid z)\right| \leq \sum_{z' \in S_{\geq 2}} k_{t,\Delta_t}(z' \mid z) \;+\; \sum_{z' \in S_{\geq 2}} q_{t,\Delta_t}(z' \mid z) \;\leq\; C_2\,\Delta_t^2,$$

where C_0, C_1 arise from the $O(\Delta_t^2)$ remainders in the first-order expansions, and C_2 from the multi-flip probabilities. Therefore,

$$\left\|Q_{t,\Delta_t}(\cdot \mid z) - K_{t,\Delta_t}(\cdot \mid z)\right\|_{\mathrm{TV}} \;=\; \frac{1}{2}\sum_{z'} \left|k_{t,\Delta_t}(z' \mid z) - q_{t,\Delta_t}(z' \mid z)\right| \;\leq\; \frac{C_0 + d(|\mathcal{V}| - 1)C_1 + C_2}{2}\,\Delta_t^2 \;\leq\; C\,\Delta_t^2,$$

with $C < \infty$.

B.2 Proof of Proposition 2

We proceed by analyzing the transition kernel of $P^\star$ in an arbitrary interval $[s, t] \subseteq [0, T]$.

Let

$$M_T \coloneqq \frac{P^\star(dz_{[0,T]})}{P(dz_{[0,T]})} = \frac{\prod_{k=1}^K G_{\tau_k}(z_{\tau_k})}{\mathbb{E}_P[\prod_{k=1}^K G_{\tau_k}(z_{\tau_k})]}, \tag{26}$$

then, let the *filtered* path measure $P_t^\star$ be a restriction of $P^\star$ to the filtration $\mathcal{H}_t$, where $0 < t < T$, and denote

$$\frac{P_t^\star(dz_{[0,t]})}{P_t(dz_{[0,t]})} = M_t. \tag{27}$$

For an event $B \in \mathcal{H}_t$, by a simple application of the Radon-Nykodym theorem and the tower property we can write

$$P_t^\star(B) = P^\star(B) = \mathbb{E}_P[1_B M_T] = \mathbb{E}_P[1_B \mathbb{E}_P[M_T \mid \mathcal{H}_t]] = \mathbb{E}_{P_t}[1_B \mathbb{E}_{P_t}[M_T \mid \mathcal{H}_t]], \tag{28}$$

where the last step follows from $\mathbb{E}_P[M_T \mid \mathcal{H}_t]$ being measurable with respect to $\mathcal{H}_t$. Hence,

$$M_t = \mathbb{E}_{P_t}[M_T \mid \mathcal{H}_t] = \frac{1}{\mathbb{E}_P[\prod_{k=1}^K G_{\tau_k}(z_{\tau_k})]} \mathbb{E}_P\left[\prod_{k=1}^K G_{\tau_k}(Z_{\tau_k}) \middle| \mathcal{H}_t\right]. \tag{29}$$

By change of measure under conditional expectation, it follows that

$$\mathbb{E}_{P^\star}[f(Z_t) \mid Z_s = z] = \frac{\mathbb{E}_P[f(Z_t)M_t \mid Z_s = z]}{\mathbb{E}_P[M_t \mid Z_s = z]} = \frac{\mathbb{E}_P[f(Z_t)h_t^\star(Z_t) \mid Z_s = z]}{h_s^\star(z)}, \tag{30}$$

where $h^\star$ is the look-ahead function in equation 5. By definition, we can express the generator $\mathcal{L}_t^\star$ of $P_t^\star$ as

$$\mathcal{L}_t^\star(f)(z) = \lim_{\Delta_t \to 0} \frac{\mathbb{E}_{P^\star}[f(Z_{t+\Delta_t}) \mid Z_t = z] - f(z)}{\Delta_t} \tag{31}$$

$$= \lim_{\Delta_t \to 0} \frac{\mathbb{E}_P\left[f(Z_{t+\Delta_t}) \frac{h^\star_{t+\Delta_t}(Z_{t+\Delta_t})}{h^\star_t(z)} \mid Z_t = z\right] - f(z)}{\Delta_t} \tag{32}$$

Moreover, we can approximate $h^\star_{t+\Delta_t}(z)$ for $t \in [\tau_k, \tau_{k+1} - \Delta_t)$, $k \in [1:K]$ using a Taylor expansion around time t

$$h^\star_{t+\Delta_t}(z) = h^\star_t(z) + \Delta_t \frac{\partial h^\star_t(z)}{\partial t} + o(\Delta_t) \tag{33}$$

$$= h^\star_t(z) - \Delta_t \sum_{i, v \neq z^i} r^i_t(v \mid z) \left[h^\star_t(z^{i \to v}) - h^\star_t(z)\right] + o(\Delta_t), \tag{34}$$

where the last line follows from Kolmogorov backward equation (Norris, 1998):

$$\frac{\partial h^\star_t(z)}{\partial t} = -\mathcal{L}_t(h^\star_t)(z) = -\sum_{i, v \neq z^i} r^i_t(v \mid z) \left[h^\star_t(z^{i \to v}) - h^\star_t(z)\right].$$

For small Δ_t, we can express $\mathbb{E}_P\left[f(Z_{t+\Delta_t}) \frac{h^\star_t(Z_{t+\Delta_t})}{h^\star_t(z)} \mid Z_t = z\right]$ using the law of total expectation, where we split the expectation based on the number of jumps in the interval $[t, t+\Delta_t]$.

$$\mathbb{E}_P\left[f(Z_{t+\Delta_t}) h^\star_t(Z_{t+\Delta_t}) \mid Z_t = z\right] \tag{35}$$

$$= f(z) h^\star_t(z) \left(1 - \Delta_t \sum_{i, v \neq z^i} r^i_t(v \mid z)\right) + \sum_{i, v \neq z^i} f(z^{i \to v}) h^\star_t(z^{i \to v}) \Delta_t r^i_t(v \mid z) + o(\Delta_t) \tag{36}$$

$$= f(z) h^\star_t(z) + \Delta_t \sum_{i, v \neq z^i} r^i_t(v \mid z) \left[f(z^{i \to v}) h^\star_t(z^{i \to v}) - f(z) h^\star_t(z)\right] + o(\Delta_t). \tag{37}$$

Combining equation 34 and equation 37, we obtain

$$\mathbb{E}_P\left[f(Z_{t+\Delta_t}) h^\star_{t+\Delta_t}(Z_{t+\Delta_t}) \mid Z_t = z\right] \tag{38}$$

$$= f(z) h^\star_t(z) + \Delta_t \sum_{i, v \neq z^i} r^i_t(v \mid z) h^\star_t(z^{i \to v}) \left[f(z^{i \to v}) - f(z)\right] + o(\Delta_t). \tag{39}$$

Then, plugging equation 39 back into equation 32, we get

$$\mathcal{L}_t^\star(f)(z) = \lim_{\Delta_t \to 0} \frac{\Delta_t \sum_{i, v \neq z^i} r^i_t(v \mid z) \frac{h^\star_t(z^{i \to v})}{h^\star_t(z)} \left[f(z^{i \to v}) - f(z)\right] + o(\Delta_t)}{\Delta_t} \tag{40}$$

$$= \sum_{i, v \neq z^i} r^i_t(v \mid z) \frac{h^\star_t(z^{i \to v})}{h^\star_t(z)} \left[f(z^{i \to v}) - f(z)\right] \tag{41}$$

By inspection, we recognize in equation 41 the generator of an IPS, with local rates

$$r^{\star,i}_t(v \mid z) := \begin{cases} r^i_t(v \mid z) \frac{h^\star_t(z^{i \to v})}{h^\star_t(z)}, & v \neq z^i, \\ -\sum_{u \neq v} r^i_t(u \mid z) \frac{h^\star_t(z^{i \to u})}{h^\star_t(z)}, & v = z^i. \end{cases} \tag{42}$$

The initial distribution follows from a simple application of the Bayes theorem, and is equal to

$$p^\star_0(z) = \frac{p_0(z) h^\star_0(z)}{\mathbb{E}_{p_0}[h_0(Z_0)]}. \tag{43}$$

B.3 Fisher's identity

Let μ be a θ-independent dominating measure on the path space $D([0,T],\mathcal{Z})$. With a slight abuse of notation, we denote the density of paths under $P_\theta(dZ_{[0,T]})$ as $P_\theta(Z_{[0,T]}) = P_\theta(dZ_{[0,T]})/\mu(dZ_{[0,T]})$. Moreover, we assume conditions hold to interchange differentiation and integration, see Klenke (2008, Thm. 6.28) for details.

We follow the same steps as Lawson et al. (2023), adapting them to our setup. This amounts to using the derivative of the logarithm and the log-derivative trick inside the expectation:

$$
\begin{aligned}
&\nabla_\theta \mathcal{L}(\theta) \\
&= \nabla_\theta \log \mathbb{E}_{P_\theta}\left[\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right] \\
&= \frac{1}{\mathbb{E}_{P_\theta}\left[\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right]} \nabla_\theta \int_{D([0,T],\mathcal{Z})} \left(\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right) P_\theta(Z_{[0,T]})\mu(dZ_{[0,T]}) \\
&= \frac{1}{\mathbb{E}_{P_\theta}\left[\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right]} \int_{D([0,T],\mathcal{Z})} \nabla_\theta \left(\left(\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right) P_\theta(Z_{[0,T]})\right) \mu(dZ_{[0,T]}) \\
&= \frac{1}{\mathbb{E}_{P_\theta}\left[\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right]} \int_{D([0,T],\mathcal{Z})} \left(\sum_{k=1}^{K} \nabla_\theta \log G_{\tau_k,\theta}(Z_{\tau_k}) + \nabla_\theta \log P_\theta(Z_{[0,T]})\right)\left(\prod_{k=1}^{K} G_{\tau_k,\theta}(Z_{\tau_k})\right) P_\theta(dZ_{[0,T]}) \\
&= \mathbb{E}_{P_\theta^\star}\left[\sum_{k=1}^{K} \nabla_\theta \log G_{\tau_k,\theta}(Z_{\tau_k}) + \nabla_\theta \log P_\theta(Z_{[0,T]})\right],
\end{aligned}
$$

where

$$
P_\theta^\star(dZ_{[0,T]}) \;\propto\; \left(\prod_{k=1}^{K} G_{\tau_k}(Z_{\tau_k})\right) P_\theta(dZ_{[0,T]}).
$$

B.4 Proof of Theorem 3

Lemma 4. *Consider a twisted IPS with local rates $r_{i,t}^{\theta,\psi}(v \mid z) = r_{i,t}^{\theta}(v \mid z) s_{i,t}^{\psi}(v,z)$ where $s_{i,t}^{\psi}(v,z) = \frac{h_t^{\psi}(z^{i\to v})}{h_t^{\psi}(z)}$. Let $\bar{p}_{t,\Delta_t}^{\theta,\psi}$ be its true transition kernel and $q_{t,\Delta_t}^{\theta,\psi}$ be the Euler approximation:*

$$
q_{t,\Delta_t}^{\theta,\psi}(z \mid z_t) = \prod_{i\in\mathcal{I}} \left(\delta_{z^i,z_t^i} + \Delta_t r_{i,t}^{\theta,\psi}(z^i \mid z_t)\right).
$$

Under assumptions (A1)-(A3) for the twisted rates (which hold when h^ψ is Lipschitz continuous in time), there exists $C < \infty$ such that:

$$
\chi^2(\bar{p}_{t,\Delta_t}^{\theta,\psi}(\cdot \mid z_t) \| q_{t,\Delta_t}^{\theta,\psi}(\cdot \mid z_t)) \le C\Delta_t^2,
$$

uniformly in $t \in [0,T]$ and $z_t \in \mathcal{Z}$.

Proof. Recall that $\chi^2(p\|q) = \sum_z \frac{(p(z)-q(z))^2}{q(z)}$. We partition $\mathcal{Z}$ as in Proposition 1:

- $S_0 = \{z_t\}$ (no change)
- $S_1 = \{z_t^{i\to v} : i \in \mathcal{I}, v \neq z_t^i\}$ (single flips)
- $S_{\ge 2} = \{z' : |\{i : z'^i \neq z_t^i\}| \ge 2\}$ (multi flips)

Let $\lambda_t^{\theta,\psi}(z_t) = \sum_{i,v\neq z_t^i} r_{i,t}^{\theta,\psi}(v \mid z_t)$. From the proof of Proposition 1, under (A1)-(A3) we have the first-order expansions:

$$
\bar{p}_{t,\Delta_t}^{\theta,\psi}(z' \mid z_t) = \begin{cases} 1 - \lambda_t^{\theta,\psi}(z_t)\Delta_t + O(\Delta_t^2) & z' \in S_0 \\ r_{i,t}^{\theta,\psi}(v \mid z_t)\Delta_t + O(\Delta_t^2) & z' = z_t^{i\to v} \in S_1 \\ O(\Delta_t^2) & z' \in S_{\ge 2} \end{cases} \tag{44}
$$

and

$$q^{\theta,\psi}_{t,\Delta_t}(z' \mid z_t) = \begin{cases} 1 - \lambda^{\theta,\psi}_t(z_t)\Delta_t + O(\Delta_t^2) & z' \in S_0 \\ r^{\theta,\psi}_{i,t}(v \mid z_t)\Delta_t + O(\Delta_t^2) & z' = z_t^{i\to v} \in S_1 \\ O(\Delta_t^2) & z' \in S_{\geq 2} \end{cases} \tag{45}$$

Case 1: $z' \in S_0$**.**

$$\frac{(\bar{p}^{\theta,\psi}(z_t) - q^{\theta,\psi}(z_t))^2}{q^{\theta,\psi}(z_t)} = \frac{(O(\Delta_t^2))^2}{1 - \lambda^{\theta,\psi}_t(z_t)\Delta_t + O(\Delta_t^2)} = \frac{O(\Delta_t^4)}{1 + O(\Delta_t)} = O(\Delta_t^4).$$

Case 2: $z' \in S_1$**.** For $z' = z_t^{i\to v}$:

$$\frac{(\bar{p}^{\theta,\psi}(z') - q^{\theta,\psi}(z'))^2}{q^{\theta,\psi}(z')} = \frac{(O(\Delta_t^2))^2}{r^{\theta,\psi}_{i,t}(v \mid z_t)\Delta_t + O(\Delta_t^2)} = \frac{O(\Delta_t^4)}{\Theta(\Delta_t)} = O(\Delta_t^3).$$

Summing over S_1 (which has $|S_1| = d(|\mathcal{V}| - 1)$ elements):

$$\sum_{z' \in S_1} \frac{(\bar{p}(z') - q(z'))^2}{q(z')} = d(|\mathcal{V}| - 1) \cdot O(\Delta_t^3) = O(\Delta_t^3).$$

Case 3: $z' \in S_{\geq 2}$**.**

$$\frac{(\bar{p}(z') - q(z'))^2}{q(z')} = \frac{(O(\Delta_t^2))^2}{O(\Delta_t^2)} = O(\Delta_t^2).$$

Since $|S_{\geq 2}| \leq |\mathcal{Z}| < \infty$:

$$\sum_{z' \in S_{\geq 2}} \frac{(\bar{p}(z') - q(z'))^2}{q(z')} = O(\Delta_t^2).$$

Total chi-squared divergence. Combining all cases:

$$\chi^2(\bar{p}^{\theta,\psi} \| q^{\theta,\psi}) = O(\Delta_t^4) + O(\Delta_t^3) + O(\Delta_t^2) = O(\Delta_t^2).$$

Therefore, there exists $C < \infty$ such that $\chi^2(\bar{p}^{\theta,\psi}_{t,\Delta_t} \| q^{\theta,\psi}_{t,\Delta_t}) \leq C\Delta_t^2$ uniformly in t and z_t. □

As the first step in proving Theorem 3, we decompose the importance weight using auxiliary distributions. Define:

- $p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t) \propto p^{\theta}_{t,\Delta_t}(z \mid z_t) h^{\psi}_{t+\Delta_t}(z) G_{t+\Delta_t}(z)^{\mathbf{1}_{t+\Delta_t \in \{\tau_k\}}}$ (twisted with approximate lookahead)
- $\bar{p}^{\theta,\psi}_{t,\Delta_t}(z \mid z_t) \propto p^{\theta}_{t,\Delta_t}(z \mid z_t) h^{\psi}_{(t+\Delta_t)^-}(z)$ (twisted without potential)

The importance weight decomposes as:

$$\frac{p^{\theta,\star}_{t,\Delta_t}(z \mid z_t)}{q^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} = \frac{p^{\theta,\star}_{t,\Delta_t}(z \mid z_t)}{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} \cdot \frac{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)}{\bar{p}^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} \cdot \frac{\bar{p}^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)}{q^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)}. \tag{46}$$

Step 1: Bounding the twist error. We have:

$$\frac{p^{\theta,\star}_{t,\Delta_t}(z \mid z_t)}{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} = \frac{h^{\star}_{t+\Delta_t}(z)}{h^{\psi}_{t+\Delta_t}(z)} \cdot \frac{Z^{\psi}_t(z_t)}{Z^{\star}_t(z_t)},$$

where

$$Z^{\psi}_t(z_t) = \sum_{z'} p^{\theta}_{t,\Delta_t}(z' \mid z_t) h^{\psi}_{t+\Delta_t}(z') G_{t+\Delta_t}(z')^{\mathbf{1}_{t+\Delta_t \in \{\tau_k\}}},$$

$$Z^{\star}_t(z_t) = \sum_{z'} p^{\theta}_{t,\Delta_t}(z' \mid z_t) h^{\star}_{t+\Delta_t}(z') G_{t+\Delta_t}(z')^{\mathbf{1}_{t+\Delta_t \in \{\tau_k\}}}.$$

By definition of $\varepsilon_{t+\Delta_t}$, for all $z \in \mathcal{Z}$:

$$e^{-\varepsilon_{t+\Delta_t}} \leq \frac{h^{\psi}_{t+\Delta_t}(z)}{h^{\star}_{t+\Delta_t}(z)} \leq e^{\varepsilon_{t+\Delta_t}}.$$

The normalizing constant ratio can be written as an expectation, by multiplying and dividing each summand in the numerator by $h^{\star}_{t+\Delta_t}(z')$:

$$\frac{Z^{\psi}_t(z_t)}{Z^{\star}_t(z_t)} = \sum_{z'} \frac{p^{\theta}_{t,\Delta_t}(z' \mid z_t) h^{\star}_{t+\Delta_t}(z') G_{t+\Delta_t}(z')^{\mathbf{1}_{t+\Delta_t \in \{\tau_k\}}}}{Z^{\star}_t(z_t)} \cdot \frac{h^{\psi}_{t+\Delta_t}(z')}{h^{\star}_{t+\Delta_t}(z')} = \mathbb{E}_{p^{\theta,\star}_{t,\Delta_t}(\cdot|z_t)} \left[\frac{h^{\psi}_{t+\Delta_t}(Z)}{h^{\star}_{t+\Delta_t}(Z)} \right]. \tag{47}$$

Since the bounds hold pointwise for each z':

$$e^{-\varepsilon_{t+\Delta_t}} \leq \frac{Z^{\psi}_t(z_t)}{Z^{\star}_t(z_t)} \leq e^{\varepsilon_{t+\Delta_t}}.$$

Combining these bounds:

$$e^{-2\varepsilon_{t+\Delta_t}} \leq \frac{p^{\theta,\star}_{t,\Delta_t}(z \mid z_t)}{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} \leq e^{2\varepsilon_{t+\Delta_t}}. \tag{48}$$

Step 2: Bounding the reset error. We have:

$$\frac{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)}{\bar{p}^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} = \frac{h^{\psi}_{t+\Delta_t}(z)}{h^{\psi}_{(t+\Delta_t)^-}(z)} \cdot G_{t+\Delta_t}(z)^{\mathbf{1}_{t+\Delta_t \in \{\tau_k\}}} \cdot \frac{\bar{Z}^{\psi}_t(z_t)}{Z^{\psi}_t(z_t)},$$

where $\bar{Z}^{\psi}_t(z_t) = \sum_{z'} p^{\theta}_{t,\Delta_t}(z' \mid z_t) h^{\psi}_{(t+\Delta_t)^-}(z')$.

When $t + \Delta_t \notin \{\tau_k\}$, h^{ψ} is continuous, so $h^{\psi}_{(t+\Delta_t)^-}(z) = h^{\psi}_{t+\Delta_t}(z)$ and the ratio equals 1.

When $t + \Delta_t \in \{\tau_k\}$, by definition of $\delta_{t+\Delta_t}$:

$$|\log h^{\psi}_{(t+\Delta_t)^-}(z) - \log G_{t+\Delta_t}(z) - \log h^{\psi}_{t+\Delta_t}(z)| \leq \delta_{t+\Delta_t},$$

which gives:

$$e^{-\delta_{t+\Delta_t}} \leq \frac{h^{\psi}_{t+\Delta_t}(z)}{h^{\psi}_{(t+\Delta_t)^-}(z)} \cdot G_{t+\Delta_t}(z) \leq e^{\delta_{t+\Delta_t}}.$$

For the normalizing constant ratio, we can use the exact same procedure of equation 47:

$$\frac{\bar{Z}^{\psi}_t(z_t)}{Z^{\psi}_t(z_t)} = \mathbb{E}_{p^{\theta,\psi}_{t,\Delta_t}(\cdot|z_t)} \left[\frac{h^{\psi}_{(t+\Delta_t)^-}(Z)}{h^{\psi}_{t+\Delta_t}(Z) G_{t+\Delta_t}(Z)} \right].$$

Since the pointwise bound holds for all z, it holds for the expectation:

$$e^{-\delta_{t+\Delta_t}} \leq \frac{\bar{Z}^{\psi}_t(z_t)}{Z^{\psi}_t(z_t)} \leq e^{\delta_{t+\Delta_t}}.$$

Combining:

$$e^{-2\delta_{t+\Delta_t}} \leq \frac{p^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)}{\bar{p}^{\theta,\psi}_{t,\Delta_t}(z \mid z_t)} \leq e^{2\delta_{t+\Delta_t}}. \tag{49}$$

Step 3: Bounding the discretization error. The distribution $\bar{p}^{\theta,\psi}_{t,\Delta_t}$ corresponds to the transition kernel of a twisted IPS with twist function $h^{\psi}_{(t+\Delta_t)^-}$. The proposal $q^{\theta,\psi}_{t,\Delta_t}$ is its Euler approximation. Between observation times, h^{ψ} is Lipschitz continuous, so assumptions (A1)-(A3) hold for the twisted rates. By Lemma 4:

$$\mathbb{E}_{q^{\theta,\psi}(\cdot \mid z_t)} \left[\left(\frac{\bar{p}^{\theta,\psi}_{t,\Delta_t}(Z \mid z_t)}{q^{\theta,\psi}_{t,\Delta_t}(Z \mid z_t)} \right)^2 \right] = 1 + \chi^2(\bar{p}^{\theta,\psi}_{t,\Delta_t} \| q^{\theta,\psi}_{t,\Delta_t}) \leq 1 + C\Delta_t^2. \tag{50}$$

Step 4: Combining the bounds. From equation 46, squaring and taking expectations:

$$\begin{aligned}\mathbb{E}_{q^{\theta,\psi}(\cdot\mid z_t)}\left[\left(\frac{p^{\theta,\star}_{t,\Delta_t}(Z\mid z_t)}{q^{\theta,\psi}_{t,\Delta_t}(Z\mid z_t)}\right)^2\right] &= \mathbb{E}_{q^{\theta,\psi}(\cdot\mid z_t)}\left[\left(\frac{p^{\theta,\star}(Z\mid z_t)}{p^{\theta,\psi}(Z\mid z_t)}\right)^2\left(\frac{p^{\theta,\psi}(Z\mid z_t)}{\bar{p}^{\theta,\psi}(Z\mid z_t)}\right)^2\left(\frac{\bar{p}^{\theta,\psi}(Z\mid z_t)}{q^{\theta,\psi}(Z\mid z_t)}\right)^2\right]\\ &\le e^{4\varepsilon_{t+\Delta_t}}\cdot e^{4\delta_{t+\Delta_t}}\cdot(1+C\Delta_t^2)\\ &= e^{4\varepsilon_{t+\Delta_t}+4\delta_{t+\Delta_t}}(1+C\Delta_t^2),\end{aligned}$$

where we used that the first two ratios are bounded uniformly by equation 48 and equation 49.

Therefore:

$$\mathrm{ESS}_{t,\Delta_t}(z_t)=\mathbb{E}_{q^{\theta,\psi}(\cdot\mid z_t)}\left[\left(\frac{p^{\theta,\star}_{t,\Delta_t}(Z\mid z_t)}{q^{\theta,\psi}_{t,\Delta_t}(Z\mid z_t)}\right)^2\right]^{-1}\ge\frac{\exp(-4(\varepsilon_{t+\Delta_t}+\delta_{t+\Delta_t}))}{1+C\Delta_t^2}.$$

This bound holds uniformly in z_t since $\varepsilon_{t+\Delta_t}$ and $\delta_{t+\Delta_t}$ are defined as suprema over all states, and the constant from Lemma 4 is uniform.

B.5 Wake-sleep objective

We discuss here the objectives used in Section 3.4. Fix an observed sequence $y_{1:K}$, and let $P^\star_\theta(\cdot\mid y_{1:K})$ denote the posterior path measure from equation 4. We write $P^\psi_\theta(\cdot\mid y_{1:K})$ for the approximate posterior IPS with initial distribution $q^\psi_0(\cdot\mid y_{1:K})$ and local rates

$$r^{\theta,\psi}_{i,t}(v\mid z)=r^\theta_{i,t}(v\mid z)\,s^\psi_{i,t}(v,z),\qquad s^\psi_{i,t}(v,z):=\frac{h^\psi_t(z^{i\to v})}{h^\psi_t(z)}.$$

Continuous-time forward KL. Let $z_{[0,T]}$ be a càdlàg trajectory with jump times $0<u_1<\cdots<u_N<T$, and let i_n denote the coordinate that changes at time u_n. Using the standard density of a CTMC path measure, the log-density of $P^\psi_\theta(\cdot\mid y_{1:K})$ is

$$\log P^\psi_\theta(z_{[0,T]}\mid y_{1:K})=\log q^\psi_0(z_0\mid y_{1:K})+\sum_{n=1}^N\log r^{\theta,\psi}_{i_n,u_n}(z^{i_n}_{u_n}\mid z_{u_n^-})+\int_0^T\sum_{i\in\mathcal{I}}r^{\theta,\psi}_{i,t}(z^i_t\mid z_t)\,dt,\tag{51}$$

where, as in the main text, $r^{\theta,\psi}_{i,t}(z^i\mid z)$ denotes the diagonal entry $r^{\theta,\psi}_{i,t}(z^i\mid z)=-\sum_{v\neq z^i}r^{\theta,\psi}_{i,t}(v\mid z)$. We study the objective

$$D_{\mathrm{KL}}\Big(P^\star_\theta(\cdot\mid y_{1:K})\,\Big\|\,P^\psi_\theta(\cdot\mid y_{1:K})\Big)=C(\theta,y_{1:K})-\mathbb{E}_{P^\star_\theta}\Big[\log P^\psi_\theta(Z_{[0,T]}\mid y_{1:K})\Big],\tag{52}$$

where $C(\theta,y_{1:K})$ collects all terms independent of ψ. Using

$$\log r^{\theta,\psi}_{i,u}(Z^i_u\mid Z_{u^-})=\log r^\theta_{i,u}(Z^i_u\mid Z_{u^-})+\log s^\psi_{i,u}(Z^i_u,Z_{u^-}),$$

we can absorb the $\log r^\theta_{i,u}$ term into the constant when optimizing with respect to ψ, yielding the continuous-time sleep objective

$$\mathcal{L}^{\mathrm{cont}}_{\mathrm{s}}(\psi;y_{1:K},\theta):=-\mathbb{E}_{P^\star_\theta}\Bigg[\log q^\psi_0(Z_0\mid y_{1:K})+\int_0^T\sum_{i\in\mathcal{I}}r^{\theta,\psi}_{i,t}(Z^i_t\mid Z_t)\,dt+\sum_{i\in\mathcal{I}}\sum_{u:Z^i_u\neq Z^i_{u^-}}\log s^\psi_{i,u}(Z^i_u,Z_{u^-})\Bigg].\tag{53}$$

Equivalently, up to an additive constant and an overall sign convention, this is the quantity reported in equation 19.

Time discretization. Let $0=t_0<t_1<\cdots<t_M=T$ be a grid containing $\{\tau_k\}_{k=1}^K$, and write $\Delta_{t_m}=t_m-t_{m-1}$. Using a Riemann approximation for the integral term in equation 53, and replacing the jump sum by the corresponding coordinate changes on the grid, we obtain

$$\mathcal{L}_{\mathrm{s}}(\psi;\,z_{t_0:t_M},y_{1:K},\theta)=-\log q^\psi_0(z_0\mid y_{1:K})-\sum_{m=0}^{M-1}\sum_{i\in\mathcal{I}}\Big[\Delta_{t_{m+1}}\,r^{\theta,\psi}_{i,t_m}(z^i_{t_m}\mid z_{t_m})+\mathbf{1}[z^i_{t_m}\neq z^i_{t_{m+1}}]\log s^\psi_{i,t_m}(z^i_{t_{m+1}},z_{t_m})\Big],\tag{54}$$

which is precisely the objective in equation 20. In practice, we estimate the expectation of equation 54 under the joint law of $(Z_{[0,T]}, Y_{1:K})$ by ancestral sampling from the prior dynamics and emission model.

Wake phase. For θ, we optimize the marginal likelihood equation 9. By Fisher's identity,

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{P^\star_\theta(\cdot|y_{1:K})}\left[\sum_{k=1}^{K} \nabla_\theta \log G_{\tau_k,\theta}(Z_{\tau_k}) + \nabla_\theta \log P_\theta(Z_{[0,T]})\right]. \tag{55}$$

To obtain a tractable estimator, we discretize the prior path density using the same Euler grid as above. The resulting approximation is

$$\log \hat{P}_\theta(z_{[0,T]}) = \log p_0^\theta(z_0) + \sum_{m=0}^{M-1}\sum_{i\in\mathcal{I}}\Big[\Delta_{t_{m+1}}\, r^\theta_{i,t_m}(z^i_{t_m} \mid z_{t_m}) + \mathbf{1}[z^i_{t_m} \neq z^i_{t_{m+1}}] \log r^\theta_{i,t_m}(z^i_{t_{m+1}} \mid z_{t_m})\Big], \tag{56}$$

which matches equation 22. Substituting equation 56 into equation 55, and approximating the posterior expectation by weighted trajectories produced by tSMC, gives the wake estimator

$$\widehat{\nabla_\theta \mathcal{L}_{\mathrm{w}}}\big(\theta;\, \{z^{(s)}_{t_0:t_M}, \bar{w}^{(s)}_T\}_{s=1}^S, y_{1:K}\big) = \sum_{s=1}^{S} \bar{w}^{(s)}_T \left[\sum_{k=1}^{K} \nabla_\theta \log G_{\tau_k,\theta}(z^{(s)}_{\tau_k}) + \nabla_\theta \log \hat{P}_\theta(z^{(s)}_{[0,T]})\right], \tag{57}$$

which is the estimator reported in equation 23. Observe that, when optimizing θ, only the emission terms and the prior path density contribute to the gradient; the twist affects the wake phase only through the quality of the particle approximation to the posterior trajectories.

C Algorithm

We provide detailed pseudocode of our training scheme using TwistNet and the KL loss in Algorithm 2. Note that Algorithm 2 runs tSMC as a subroutine, which we describe in detail in Algorithm 3.

Let the cost of a forward pass of the rates, the score, and the potentials be C_r, C_s, C_G, respectively. Then, the time complexity of an update for a single datapoint in the sleep phase is

$$\Theta\big(\underbrace{MC_r + KC_G}_{\text{Simulation}} + \underbrace{MC_s}_{\text{Loss}}\big),$$

while in the wake phase it is

$$\Theta\big(\underbrace{\overbrace{S(M(C_r + C_s) + KC_G)}^{\text{Simulation}} + \overbrace{MS}^{\text{Resampling}}}_{\text{tSMC}} + \underbrace{MC_r}_{\text{Loss}}\big),$$

where the cost of computing the twist is absorbed into that of computing the score, since we are using the efficient parameterization described in Section 3.4. Using this parameterization, rather than having to compute $d\,(|\mathcal{V}| - 1) + 1$ forward passes of a twist model, we have a cost that is roughly $\Theta(C_\Phi + (d\,(|\mathcal{V}| - 1) + 1)C_\rho)$, where C_Φ and C_ρ are the cost of the context encoder and the aggregator in equation 17. In our experiments, we let Φ bear the cost of heavy operations such as processing future observations, observation times, as well as covariates and positional information, while ρ is a simple two-layer MLP.

The time and memory cost of the loss terms can be reduced to $\Theta(C_s)$ and $\Theta(C_r)$ by employing a Monte Carlo approximation of time, only considering a single timestep for each update. This is particularly useful when large neural networks are employed in parameterizing either the rates or the score.

A few practical notes:

(i) To amortize simulation cost, after sampling a mini-batch of trajectories (sleep) or running tSMC (wake) we perform several optimizer steps on the *same* batch before regenerating trajectories.

Algorithm 2 Wake–Sleep with twisted SMC for latent IPSs

Inputs: $\mathcal{D}_{\text{train}} = \{(y^{(b)}_{1:K}, \tau^{(b)}_{1:K})\}$; optimizer steps $\textsc{GradStep}_\psi, \textsc{GradStep}_\theta$; time grid $0 = t_0 < \cdots < t_M = T$; # of particles S; batch size B; global updates G; updates per phase N; initializations ψ_0, θ_0; Monte Carlo loss flag `mc_loss`
$\psi \leftarrow \psi_0, \theta \leftarrow \theta_0$.
for $g = 1, \dots, G$ **do**
  # Sleep phase
  for $n = 1, \dots, N$ **do**
    for $b = 1, \dots, B$ **do**
      Simulate $z^{(b)}_{[t_0:t_M]}$ from the prior P_θ via Euler steps.
      Simulate synthetic observations $\tilde{y}^{(b)}_k \sim p_\theta(\cdot \mid z^{(b)}_{\tau_k})$, for $k = 1, \dots, K$.
      if `mc_loss` **then**
        $m \sim \mathcal{U}(\{0, \dots, M-1\})$
        $\ell^{(b)}_{\text{sleep}} = -\log q^\psi_0(z^{(b)}_0 \mid \tilde{y}^{(b)}_{1:K}) - M \sum_{i \in \mathcal{I}} \left[\Delta_{t_{m+1}} r^{\theta,\psi}_{i,t_m}(z^{i,(b)}_{t_m} \mid z^{(b)}_{t_m}) + \mathbf{1}[z^{i,(b)}_{t_m} \neq z^{i,(b)}_{t_{m+1}}] \log s^\psi_{i,t_m}(z^{i,(b)}_{t_{m+1}}, z^{(b)}_{t_m}) \right]$
      else
        $\ell^{(b)}_{\text{sleep}}(\psi) = -\log q^\psi_0(z^{(b)}_0 \mid \tilde{y}^{(b)}_{1:K}) - \sum_{m=0}^{M-1} \sum_{i \in \mathcal{I}} \left[\Delta_{t_{m+1}} r^{\theta,\psi}_{i,t_m}(z^{i,(b)}_{t_m} \mid z^{(b)}_{t_m}) + \mathbf{1}[z^{i,(b)}_{t_m} \neq z^{i,(b)}_{t_{m+1}}] \log s^\psi_{i,t_m}(z^{i,(b)}_{t_{m+1}}, z^{(b)}_{t_m}) \right]$.
      end if
    end for
    $\psi \leftarrow \textsc{GradStep}_\psi \left(\nabla_\psi \frac{1}{B} \sum_{b=1}^B \ell^{(b)}_{\text{sleep}}(\psi) \right)$
  end for
  $\bar{\theta} \leftarrow \theta$ *# Lagged θ to use for tSMC proposal*
  # Wake phase
  for $n = 1, \dots, N$ **do**
    for $b = 1, \dots, B$ **do**
      $(y^{(b)}_{1:K}, \tau^{(b)}_{1:K}) \sim \mathcal{D}_{\text{train}}$
      Simulate approx. posterior via $\{z^{(b,s)}_{[t_0:t_M]}, \bar{w}^{(b,s)}\}_{s=1}^S \leftarrow \text{tSMC}\left(y^{(b)}_{1:K}, \tau^{(b)}_{1:K}\right)$ with prior dynamics P_θ, potentials $G_{\cdot,\theta}$, twist function h^ψ and proposal $q^{\bar{\theta},\psi}$ from equation 12 (see Algorithm 3).
      $z^{(b)}_{[t_0:t_M]} \leftarrow z^{(b,s^\star)}_{[t_0:t_M]}$, where $s^\star \sim \text{Categorical}\left(\left\{ \bar{w}^{(b,s)} \right\}_{s=1}^S \right)$
      if `mc_loss` **then**
        $m \sim \mathcal{U}(\{0, \dots, M-1\})$, $k \sim \mathcal{U}(\{1, \dots, K\})$
        $\ell^{(b)}_{\text{wake}}(\theta) = -\log p^\theta_0(z^{(b)}_0) - M \sum_{i \in \mathcal{I}} \left[\Delta_{t_{m+1}} r^\theta_{i,t_m}(z^{i,(b)}_{t_m} \mid z^{(b)}_{t_m}) + \mathbf{1}[z^{i,(b)}_{t_m} \neq z^{i,(b)}_{t_{m+1}}] \log r^\theta_{i,t_m}(z^{i,(b)}_{t_{m+1}} \mid z^{(b)}_{t_m}) \right] - K \log G_{\tau_k,\theta}(z^{(b)}_{\tau_k})$.
      else
        $\ell^{(b)}_{\text{wake}}(\theta) = -\log p^\theta_0(z^{(b)}_0) - \sum_{m=0}^{M-1} \sum_{i \in \mathcal{I}} \left[\Delta_{t_{m+1}} r^\theta_{i,t_m}(z^{i,(b)}_{t_m} \mid z^{(b)}_{t_m}) + \mathbf{1}[z^{i,(b)}_{t_m} \neq z^{i,(b)}_{t_{m+1}}] \log r^\theta_{i,t_m}(z^{i,(b)}_{t_{m+1}} \mid z^{(b)}_{t_m}) \right] - \sum_{k=1}^K \log G_{\tau_k,\theta}(z^{(b)}_{\tau_k})$.
      end if
    end for
    $\theta \leftarrow \textsc{GradStep}_\theta \left(\nabla_\theta \frac{1}{B} \sum_{b=1}^B \ell^{(b)}_{\text{wake}}(\theta) \right)$
  end for
end for

(ii) In our experiments we resample at every step, hence particle weights are $1/S$. Due to compute constraints we use relatively few particles (dictated by high dimensionality), and often observe particle collapse. To reduce memory cost, in the wake phase we do not weight the loss across particles. Instead, we draw a single path $z^{(s^\star)}_{[t_0:t_M]}$ by importance resampling using the final normalized weights and compute the wake loss on that path only. This yields a consistent estimator under importance resampling and is effectively equivalent to using self-normalized importance weights (Chopin and Papaspiliopoulos, 2020).

(iii) Resampling is performed using systematic resampling (Chopin and Papaspiliopoulos, 2020).

Algorithm 3 tSMC

Inputs: $(y_{1:K}, \tau_{1:K})$; time grid $0 = t_0 < \cdots < t_M = T$; # of particles S; prior kernel p^θ and initial distribution p_0^θ; potential function $G_{\cdot,\theta}$; twist function h^ψ; proposal kernel $q^{\bar{\theta},\psi}$ and initial distribution q_0^ψ; ESS threshold τ_{ESS}

Output: $\{z^{(s)}_{[t_0:t_M]}, \bar{w}^{(s)}\}_{s=1}^S$

Initialize

$$z_{t_0}^{(s)} \sim q_0^\psi, \quad w_0^{(s)} \leftarrow \frac{p_0^\theta(z_0^{(s)}) h_0^\psi(z_0^{(s)}) G_{0,\theta}(z_0^{(s)})^{\mathbf{1}_{\{\tau_k\}}(0)}}{q_0^\psi(z_0^{(s)})}, \quad \text{for} \quad s = 1, \ldots, S.$$

for $m = 0, \ldots, M-1$ **do**
  for $s = 1, \ldots, S$ **do**
    if $\text{ESS}(\{w_m^{(s)}\}) < \tau_{\text{ESS}}$ **then**
      Resample $z_{t_0:t_m}^{(s)} \leftarrow z_{t_0:t_m}^{(a_s)}$ using ancestor $a_s \sim \text{Categorical}\left(\left\{w_m^{(s)} / \sum_{u=1}^S w_m^{(u)}\right\}\right)$.
      Reset weights to $w_m^{(s)} \leftarrow 1$
    end if
  end for
  for $s = 1, \ldots, S$ **do**
    Propose $z_{t_{m+1}}^{(s)} \sim q_{t_m,\Delta t}^{\bar{\theta},\psi}(\cdot \mid z_{t_m}^{(s)})$
    Update weights

$$w_{m+1}^{(s)} \leftarrow w_m^{(s)} \times \frac{p_{t_m,\Delta t}^\theta(z_{t_{m+1}}^{(s)} \mid z_{t_m}^{(s)})}{q_{t_m,\Delta t}^{\bar{\theta},\psi}(z_{t_{m+1}}^{(s)} \mid z_{t_m}^{(s)})} \times \frac{h_{t_{m+1}}^\psi(z_{t_{m+1}}^{(s)})}{h_{t_m}^\psi(z_{t_m}^{(s)})} \times G_{t_{m+1},\theta}(z_{t_{m+1}}^{(s)})^{\mathbf{1}_{\{\tau_k\}}(t_{m+1})}$$

  end for
end for
return paths $\{z_{[t_0:t_M]}^{(s)}\}$ and final normalized weights $\{w_M^{(s)} / \sum_{u=1}^S w_M^{(u)}\}$

D Experimental details

All code for the experiments can be found at the following link: https://github.com/giosueio/LatentIPS.

D.1 Baselines

We compare our method against alternatives that are scalable to high-dimensional systems, amenable to gradient-based optimization of neural models for the dynamics of the model, and do not require expensive operations such as backpropagation through time. This rules out most approaches based on a reverse KL objective, such as Naesseth et al. (2018); Lawson et al. (2022). We note that the NeuralMJP method in Seifner and Sánchez (2023) could, in principle, ease the memory cost of backpropagating through time by using the adjoint method (Chen et al., 2018). However, we failed to have this method converge to meaningful solutions in our setup, due to exploding gradients – possibly coupled with those gradients being biased, due to the repeated Gumbel-softmax approximations (Seifner and Sánchez, 2023; Jang et al., 2017). We note that in Seifner and Sánchez (2023) high-dimensional, interacting systems such as the ones considered in this work were not addressed. We also exclude methods that are not amenable to amortization, as this would make inference for neural models much more complex. This rules out the methods proposed in Opper and Sanguinetti (2007); Wildner and Koeppl (2019); Köhs et al. (2021); Eich et al. (2025).

The baselines we consider are:

- **Bootstrap particle filter** (BPF)(Doucet and Johansen, 2009): this method is an SMC algorithm using the filtering distributions as intermediate targets, and the prior transition probabilities as a proposal distribution.

- **NASMC and NAS-X** (Gu et al., 2015; Lawson et al., 2023): these are both SMC algorithms with an informative proposal learned with a forward KL loss. In Gu et al. (2015) the intermediate targets are the filtering distributions, while in Lawson et al. (2023) the intermediate targets are the twisted distributions, where the twist function is learned using a density ratio estimation loss. We parameterize the proposal

distribution by fitting a score network minimizing the forward KL loss in 20,

$$\log \frac{h_t(z^{i\to v})}{h_t(z)} \approx s_t^{\psi}(z)_{[i,v]},$$

where $s_t^{\psi} : \mathcal{Z} \to \mathbb{R}^{d\times V}$ and $s_t^{\psi}(z)_{[i,z^i]} = 1$ for $i \in \mathcal{I}$.

- **Taylor-approximated guidance** (TAG): to overcome the inefficiency of computing $d \times (V-1)+1$ forward passes of the twist function, Nisonoff et al. (2025) proposed to compute a first-order Taylor approximation of the log-twist evaluated at a specific value z, i.e.

$$\log h_t^{\psi}(z_t) \approx \log h_t^{\psi}(\mathbf{z}) + \mathbf{z}_t^{\top} \nabla_{\mathbf{z}} \log h_t^{\psi}(\mathbf{z}) \tag{58}$$

  where $\mathbf{z}, \mathbf{z}_t$ are one-hot encoded versions of z, z_t, enabling a single forward pass at z of the twist function. Note that backpropagation of the KL loss with respect to the score function implied by equation 58 would require computing a second derivative, on top of the spatial derivative with respect to $\mathbf{z}$. This can be extremely expensive for neural models, therefore we only consider a DRE loss for this method.

For all of our methods (except BPF), we also train a variational initial distribution $q_0^{\psi}(z_0)$ with a forward KL loss on generated trajectories (see the full objective in equation 20).

The density ratio estimation (DRE) loss (Lawson et al., 2022, 2023) used to learn the twist in NAS-X and TAG is

$$\hat{\mathcal{L}}_{\text{DRE}}(\psi) = \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{I}} \log \sigma(\log h_t^{\psi}(z_t^+; y_{\geq t}, \tau_{\geq t})) + \log(1 - \sigma(\log h_t^{\psi}(z_t^-; y_{\geq t}, \tau_{\geq t}))), \tag{59}$$

where $\sigma : \mathbb{R} \to [0,1]$ is the logistic function. Positive samples z_t^+ are generated by the forward model using ancestral sampling of $z_{[0,T]}^+ \sim P_\theta$ first, and then $y_{1:K}, \tau_{1:K}$. Negative samples $z_t^- \sim P_\theta$, and are hence uncoupled from $y_{1:K}, \tau_{1:K}$. Using this loss is equivalent to training a classifier to distinguish between coupled and uncoupled samples.

All of the methods were trained on a single NVIDIA RTX A5000 with 24GB of VRAM.

D.2 SIRS model

Data generation. We simulate SIRS epidemics on undirected graphs with d nodes. Graphs are sampled from an expected-degree model using `networkx.expected_degree_graph` (Hagberg et al., 2008), and all nodes have expected degree 5. Each node i has a feature vector $\xi_i \in \mathbb{R}^{16}$, included to make posterior inference more challenging. Ground truth paths on $[0,T] = [0,10]$ are drawn using Gillespie's algorithm (Gillespie, 1977; Wilkinson, 2018), with rate parameters in equation 25 fixed to $(\alpha_0, \alpha_1, \beta, \gamma) = (0.1, 1.0, 0.4, 0.05)$. We assign 50 trajectories for the training set, and 50 to the test set.

Observations model. Observations are sampled at $K = 10$ snapshot times $\tau_1 < \cdots < \tau_K$, sampled from a uniform distribution in $[0,10]$ for each trajectory. Observations are conditionally independent across nodes, given Z_{τ_k}. For each node $i \in \mathcal{I}$ we include an explicit mask token $\varnothing$ and use the node-factorized emission distribution

$$p(y_{\tau_k} \mid Z_{\tau_k} = z) \;=\; \prod_{i\in\mathcal{I}} g\big(y_{\tau_k}^i \mid z^i\big)\,,$$

with a masking probability set to $p_{\text{mask}} = \frac{1}{2}$ and small symmetric label noise $\delta > 0$, for numerical stability:

$$g(c \mid z^i) \;=\; \begin{cases} p_{\text{mask}}, & c = \varnothing, \\ (1-p_{\text{mask}})\big[(1-\delta(V-1))\,\mathbf{1}\{c = z^i\} + \delta\,\mathbf{1}\{c \neq z^i\}\big]\,, & c \in \mathcal{V}, \end{cases}$$

where $V = |\mathcal{V}|$.

Twist model. We parameterize the score networks and the encoder of the TwistNet with a Graph Transformer (GT) from Vignac et al. (2022). For all of our models we use 2 GT layers and 4 heads, with a node embedding dimension of 64, an edge embedding dimension of 8, and global information embedding of 32, for a total of 485,827 parameters. We did not tune any of these hyperparameters and we keep them fixed throughout our experiments. For any time $t \in [0, T]$, we feed as input the feature vector, future observation $y_{\geq t}$ and observation times $\tau_{\geq t}$, and graph statistics computed on the adjacency matrix as in (Vignac et al., 2022). The score network also takes as input the current state z_t, while in the TwistNet this is only considered when aggregating encoder outputs. For the TwistNet, we let the last layer be a two-layer MLP with $m = 64$.

Training. For optimization of the TwistNet and score models, we use the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001 in the latent trajectory inference task, and 0.0003 for parameter learning. Parameters of the rates in equation 25 are optimized using Adam with learning rate 0.005. All of the other hyperparameters are set to their default in PyTorch [1]. We did not tune these values. For our latent trajectory inference experiment, we trained using the forward KL loss on the twist and score models for 1,000 steps, with a batch size of 32 and $\Delta_t = 0.1$. We note that all of the methods converged. Covariates ξ to sample from the prior in the sleep phase were sampled from the test set. For the parameter learning task, we initialize all parameters of the rates at 0.2. We then train with batch size 16, using $\Delta_t = 0.05$, $S = 10$ tSMC particles for each sample in the wake phase, and we resample at every step (i.e., $\tau_{\text{ESS}} = 1$ in Algorithm 3). We found it helpful to begin the training loop by optimizing ψ with sleep steps (see Algorithm 2) until convergence, which could be achieved in roughly 2,500 steps, and then alternating wake and sleep steps. We reuse each simulated mini-batch for multiple inner updates to reduce simulator calls, using 25 optimizer steps per batch before resampling fresh trajectories and employing a Monte Carlo approximation of the time summation in the objective. We use 25 steps in each phase for 25 global iterations, i.e. for a total of 625 steps (not counting the repeated steps for each batch). In the wake phase we compute the loss on a *single* trajectory sampled by importance resampling from the final particle weights, rather than a weight-averaged objective, as explained in Section C.

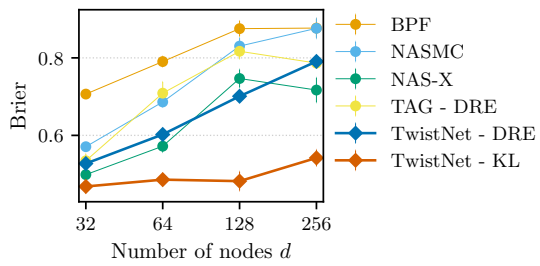

Figure 6: Latent trajectory reconstruction of each inference method, measured by Brier score of true latent trajectories with respect to the posterior approximations. Methods using TwistNet are highlighted by a thicker line. Error bars correspond to two standard errors.

Evaluation. In our *latent trajectory inference* experiment we are interested in understanding whether our method can be used to perform posterior inference given a set of observations and a prescribed forward model, with no parameter learning. From tSMC particles we form per-time nodewise marginals $\hat{p}_t(z) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}(z_t^{(s)} = z)$, where weights are uniform because we resampled at every step (i.e. we let $\tau_{\text{ESS}} = 1$. in Algorithm 3). For each method we consider 25 particles ($S = 25$), except for BPF for which we take 250 particles. We also add a small uniform weight to avoid numerical issues when support is scarce:

$$\tilde{p}_t \;=\; (1-\epsilon)\,\hat{p}_t \;+\; \epsilon\,\text{Unif}(\mathcal{V}).$$

Let $z_t^\star \in \mathcal{Z}$ be the ground-truth state of the latent trajectory at time $t \in [0, T]$. We report an average of the following metrics over a test set of 50 trajectories:

$$\text{CE} = -\frac{1}{M}\sum_{m=1}^{M} \log \tilde{p}_{t_m}(z_{t_m}^\star), \quad \text{Brier} = \frac{1}{M}\sum_{m=1}^{M} \|\tilde{p}_{t_m} - \mathbf{z}_{t_m}^\star\|_2^2.$$

where $0 = t_1 < \cdots < t_M = T$ is the set of discretized time indices, and $\mathbf{z}_t^\star$ is the one-hot encoded $z_t^\star$. The CE loss over dimensions is displayed in Figure 3, and the Brier score in Figure 6.

In our *parameter learning* experiment, we keep track of individual parameter estimates and total relative parameter error $\sum_j |\hat{\theta}_j - \theta_j|/|\theta_j|$ over the four parameters in equation 25.

[1] `https://docs.pytorch.org/docs/stable/generated/torch.optim.Adam.html`

D.3 Wildfires trajectories

Dataset. As mentioned in Section 4.2, we consider a subset of the trajectories in the WildfireSpreadTS dataset (Gerard et al., 2023). We filter them based on the following criteria:

- We take a 64×64 crop at the center of the image at the day corresponding to the starting date in GlobFire (Artés et al., 2019), and consider trajectories that are a single week long. Note that images are created based on the final shape of the fire (Gerard et al., 2023), hence the center point does not necessarily correspond to the point of ignition.
- Since many of these crops do not contain any active fire pixel, we filter trajectories so that the 64×64 grid at the first day contains at least one fire pixel, and there are at least two of the other six days with at least one pixel of active fire.

All covariates except for VIIRS reflectance channels are excluded. We decided to keep these variables because Gerard et al. (2023) showed empirically that they are the most predictive, in terms of cross-validation error.

Rates model. Local rates are computed via a neural network that takes as input the VIIRS channels $x_{1:7} \in \mathbb{R}^{64\times64\times3\times7}$ for the entire week (i.e., we are performing prediction conditioning on covariates that occur at future time points) and the current one-hot encoded state $\mathbf{z}_t \in \{0,1\}^{64\times64\times3}$, concatenated to VIIRS inputs along the channel dimension after repeating 7 times along the time dimension. The model is a UTAE network (Garnot and Landrieu, 2021), selected for its good performance in Gerard et al. (2023). This model resembles a UNET, with temporal self-attention (Garnot and Landrieu, 2021). We set the number of channels at each layer to be $[64, 64, 64, 128]$ for upsampling steps and $[128, 64, 32, 32, 3]$ for downsampling steps, and the embedding dimension for self-attention to 128, for a total of $1,574,725$ parameters. We did not tune these hyperparameters. We let the output be a field in $\mathbb{R}^{64\times64\times3}$, and exponentiate to ensure positivity. Before turning these into outflow rates, we mask values so that transitions from unburned (U) to burned (B) have rates zero, and set the rates of transitioning to active (A) to zero for all pixels with no Moore neighbors being active. Finally, we fill the diagonal of the outflow rates with the negative sum of the off-diagonals. We let the initial distribution be a UTAE with channels $[32, 32, 64]$ for upsampling and $[64, 32, 32]$ for downsampling, conditioned on VIIRS channels.

Observations model. Observations are binarized fields of active fire pixels, taken directly from (Gerard et al., 2023). These are obtained once a day by aggregating two overpasses from the VIIRS satellite, hence are not exactly an instantaneous snapshot. For simplicity, we treat it as such and model the emission distribution using a simple scalar value: $p\big(y_k^i = 1 \mid Z_{\tau_k}^i = A\big) = \sigma(\theta_{\text{detect}})$, where θ_{detect} is a scalar logit and σ the logistic function. We set $p\big(y_k^i = 1 \mid Z_{\tau_k}^i \in \{U, B\}\big) = \delta$ with $\delta = 0.001$ for numerical stability. In other words, if a fire is active in a single pixel we have probability $\sigma(\theta_{\text{detect}})$ of observing it, and if a fire is not active we will most likely not detect it (probability of $1 - \delta$ of the observed pixel being zero). This choice is scientifically motivated: pixels are likely to be obscured by clouds and smoke, and we assume contamination does not occur in the other direction (i.e. very few false negatives).

Twist model. We use the same UTAE model for the context encoder of the twist and the score networks, with channel multiplicities of $[64, 64, 64, 128]$ during upsampling and $[128, 64, 32, 32]$ during downsampling, and embedding dimension of 64 for self-attention. We let the context encoder depend on the set of weekly observations $y_{1:7} \in \{0,1\}^{64\times64\times1\times7}$ and VIIRS covariates $x_{1:7} \in \mathbb{R}^{64\times64\times3\times7}$, concatenated along the channel dimension. We also let the model depend on the scalar time t, by applying adaptive normalization layers Perez et al. (2018) to intermediate activations, following the implementation in Peebles and Xie (2023). The score network (used in NASMC and NAS-X) is identical, but it also takes as input the current one-hot encoded state $\mathbf{z}_t \in \{0,1\}^{64\times64\times3}$ by repeating it 7 times and concatenating it to the other inputs along channel dimensions. For both the context encoder and the score we use an additional convolutional layer to map to the desired output channel dimension, corresponding to 3 for the score net and $3 \times m$ for the TwistNet, where $m = 256$. In the TwistNet, these embeddings are then passed through a two-layer MLP to produce the logarithm of twist values. We let the posterior initial distribution be a UTAE with channels $[32, 32, 64]$ for upsampling and $[64, 32, 32]$ for downsampling, conditioned on VIIRS channels and future observations. We also train an encoder conditioned only on VIIRS channels and the first observation. NAS-X makes use of an additional model producing the twist value. We let this be identical to the score network, and perform mean pooling over the output to get a scalar value. This

Table 3: Parameter estimates and relative parameter error (RPE) for the SIRS model with 64 nodes, mean ± 2 standard deviations across 10 random seeds.

	α_0	α_1	β	γ	RPE
Ground truth	0.1	1.0	0.4	0.05	–
TwistNet - KL	0.117 ± 0.021	**0.888** ± 0.115	**0.396** ± 0.030	0.047 ± 0.007	**0.388** ± 0.222
TwistNet - DRE	0.066 ± 0.022	0.842 ± 0.136	0.328 ± 0.039	**0.049** ± 0.014	0.797 ± 0.319
TAG - DRE	0.067 ± 0.085	0.723 ± 0.271	0.307 ± 0.104	0.096 ± 0.130	1.891 ± 3.075
NAS-X	**0.092** ± 0.024	0.777 ± 0.123	0.359 ± 0.044	**0.049** ± 0.009	0.500 ± 0.293
NASMC	0.015 ± 0.017	0.640 ± 0.326	0.329 ± 0.105	0.035 ± 0.021	1.728 ± 0.981

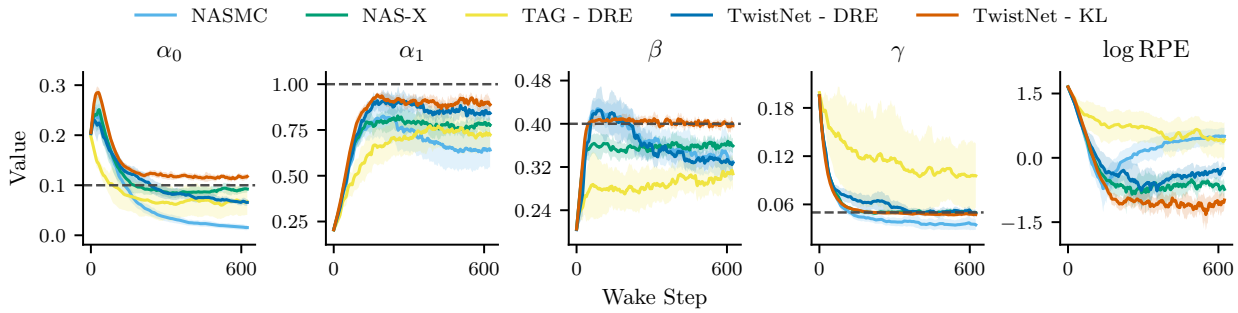

Figure 7: Evolution of the parameters and log relative parameter error (log RPE) through wake steps updating the parameters θ, for a SIRS model on a graph with 64 nodes.

makes the number of parameters for NAS-X and TwistNet roughly equal: $2,845,961 = 1,268,355 \times 2$ for the score and twist networks in NAS-X, and $2,808,420 = 1,709,664 + 789,505$ for the TwistNet, combining context encoder and aggregator.

Training. We follow the wake-sleep routine detailed in Algorithm 2. We start by training the h^{ψ}, the initial distribution, and the first-observation encoder using the Adam optimizer (learning rate 5×10^{-4}, PyTorch defaults otherwise) with batch size 16 for 150 steps. We employ a Monte Carlo approximation of the time summation at each training step, and the time grid is built by sampling Δ_t uniformly from $\{0.02, 0.05, 0.1\}$. We found that including finer grids helps the twist model learn how to interpolate between observations where multiple transitions occur, which is typical in this dataset given the rapid spread of wildfires. We then alternate blocks of wake updates for θ and sleep updates for ψ for G=15 outer iterations. We amortize computation by performing 25 gradient updates per simulated batch. During wake steps, we use the Adam optimizer on θ with the same learning rate of 5×10^{-4}, batch size of 8, and 40 updates per outer iteration. Each update uses tSMC with proposal $q^{\bar{\theta},\psi}$, M=5 particles, systematic resampling at every step (ESS threshold = 1), and Δ_t sampled uniformly from $\{0.05, 0.1\}$. Because we resample at every step, we avoid weighting the wake loss across particles. Instead, we draw one path by importance resampling using the final normalized weights and evaluate the wake loss on that path.

Evaluation. We consider two tasks:

- *Reconstruction*: we generate trajectories conditioning on the full observation sequence $y_{1:7}$, and evaluate how well we recover the active fire areas. We run tSMC (Algorithm 3) (or SMC for NASMC) with S=16 particles, Δ_t=0.02, and resample at every step (uniform weights at the daily grid). We found that the performance of every method could be improved in this task by introducing a temperature parameter α, and modifying both the twist and the score by raising them to the power of α. To set the temperature, we grid-search $\alpha \in \{0.05, 0.10, 0.25, 0.50, 1.0\}$ on the first three training batches and fix the best $\alpha^{\star}$ for each method for evaluation.
- *Prediction*: after conditioning on the first observation, we run the prior model for $S = 16$ samples per set of observations and see how well the resulting empirical distribution can predict future trajectories.

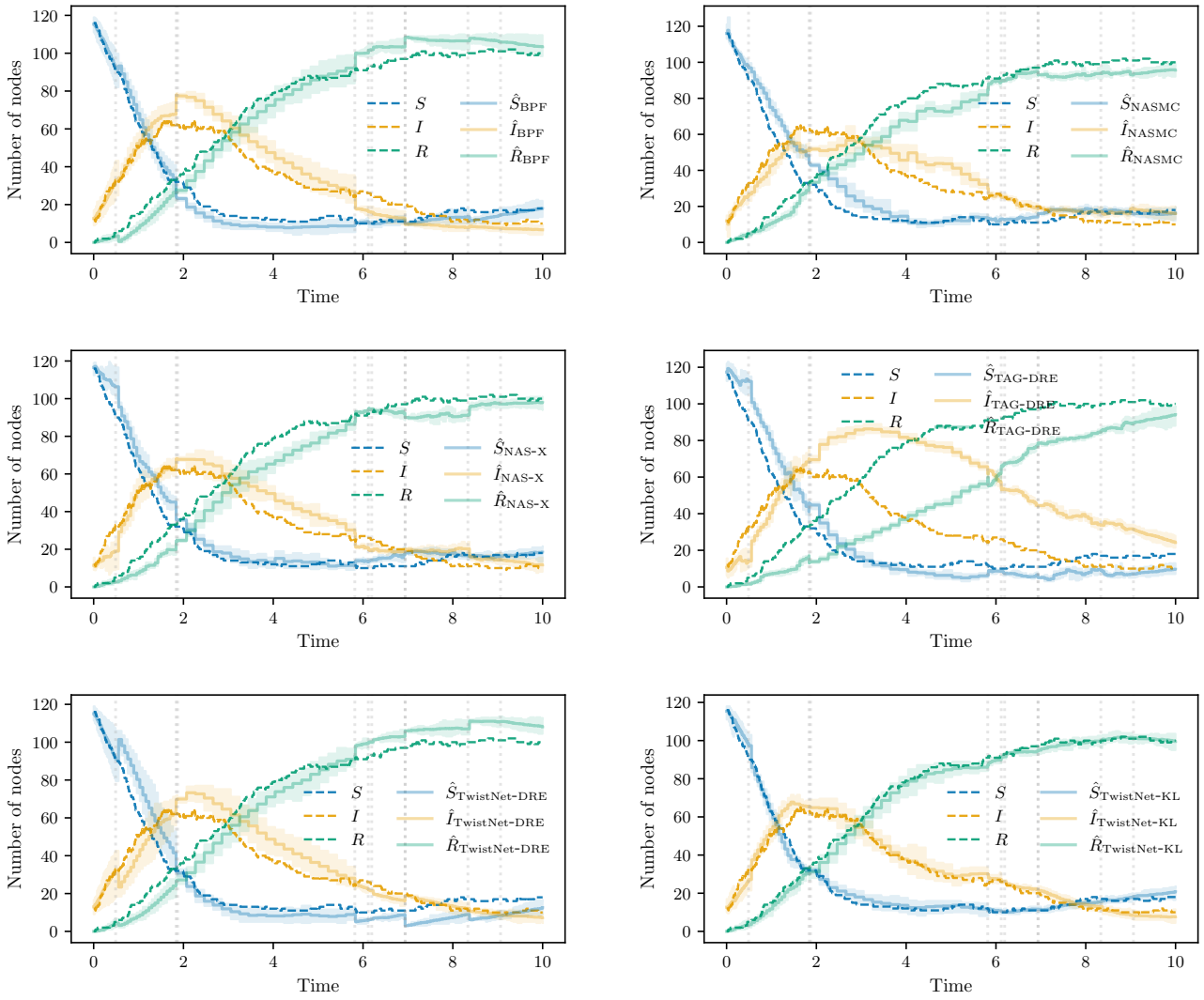

Figure 8: First example of latent trajectories, with counts of each state over the graph at each timestep.

For both of these tasks, the metric we use is the binary cross-entropy of the empirical distribution from the particles against the observed active-fire map, averaging over pixels and days per trajectory. We report mean ± 2 standard errors across 26 test trajectories in Table 2.

E Additional Results

E.1 SIRS model

We replicate the parameter learning experiment for graphs with $d = 64$ nodes using 10 random seeds, and display the results in Table 3 and the values at each step in Figure 7. By comparing it with the results for 32 nodes, we can notice that the gap between the TwistNet with KL loss and the other methods widens as dimensionality increases, in line with the experiment on latent trajectory inference.

In Figures 8 and 9, we show a summary of generated latent trajectories for each method for different graphs with $d = 128$ nodes, displaying a count of nodes in each state. In line with the evaluation in Figures 3 and 6, the TwistNet-KL method seems to display much better performance than the alternatives in terms of closeness to the ground truth. In Figure 10 and Figure 11 we display a segment between two observations for graphs with 32 and 64 nodes respectively, and highlight the mismatch of the latent trajectories with the (non-masked) observations.

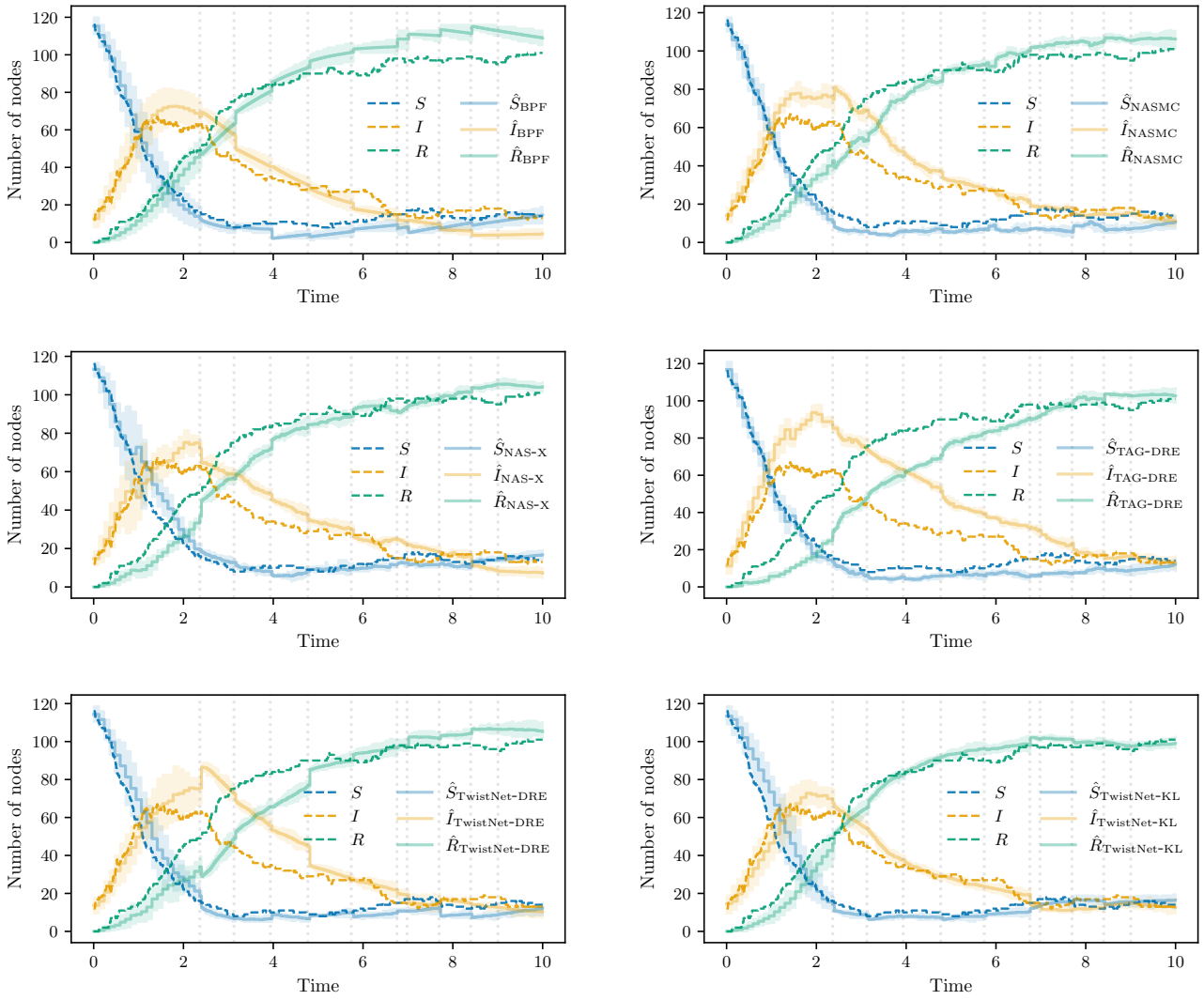

Figure 9: Second example of latent trajectories, with counts of each state over the graph at each timestep.

While twisted SMC methods shows huge improvements in performance over traditional schemes for high-dimensional problems, this comes at a cost: the need to perform a forward pass of the neural network at each timestep. This significantly increases the runtime of this family of methods, and this is possibly the biggest limitation of these methods when applied to large datasets.

E.2 Wildfires trajectories

In Figure 12 we show the VIIRS channels and empirical prior marginals (for $S = 16$ samples and $\Delta_t = 0.02$) for the trajectories displayed analyzed in Figure 5. Prior samples are obtained after encoding the first observation. We note that, other than TwistNet with a KL loss, none of the baselines seems to have learned a meaningful model for the prior dynamics, despite their losses having converged.

Additional examples of ground truth, inputs, prior predictions and posterior reconstructions on the test set are displayed in Figure 13 and Figure 14, all using $S = 16$ samples and $\Delta_t = 0.02$. For the prior we display the empirical distribution of active fire, whereas for the posterior we display a single (non-cherry-picked) sample. A particularly challenging scenario is displayed in Figure 14 on the right: the fire has a single burning pixel at the start but develops into a large fire at days 6 and 7, and VIIRS channels are missing at two timesteps. TwistNet with a KL loss tries to smoothly interpolate, slightly accelerating between the fifth and the sixth day. All the other methods fail, and none of the predictions from the prior model seems to capture the possibility of a rapid

spread.

Figure 10: Samples in between observations at times $\tau_3 = 1.6$ and $\tau_4 = 1.9$, for a graph with 32 nodes. Mismatches between unmasked observations and samples are highlighted in red.

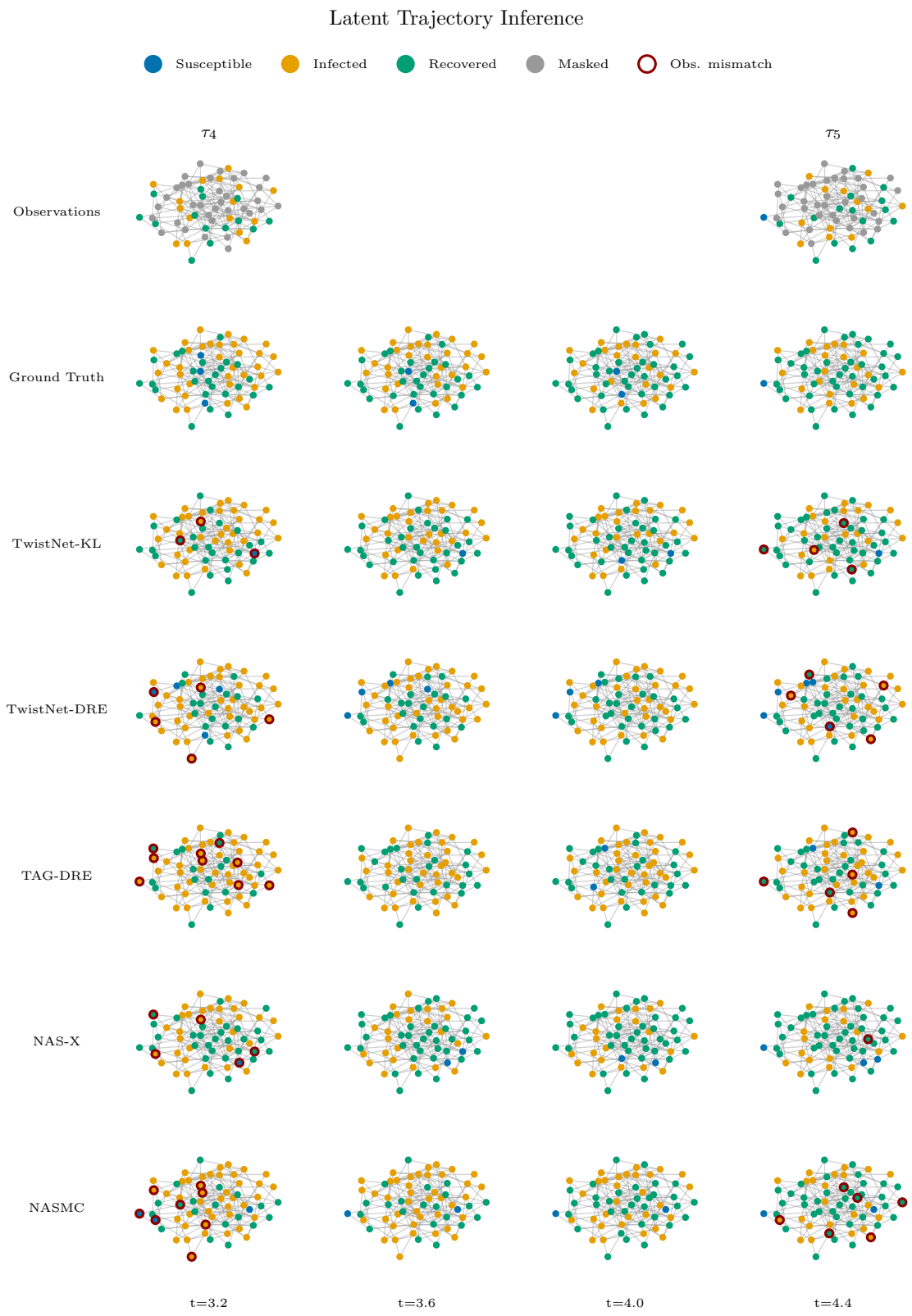

Figure 11: Samples in between observations at times $\tau_4 = 3.2$ and $\tau_5 = 4.4$, for a graph with 64 nodes. Mismatches between unmasked observations and samples are highlighted in red.

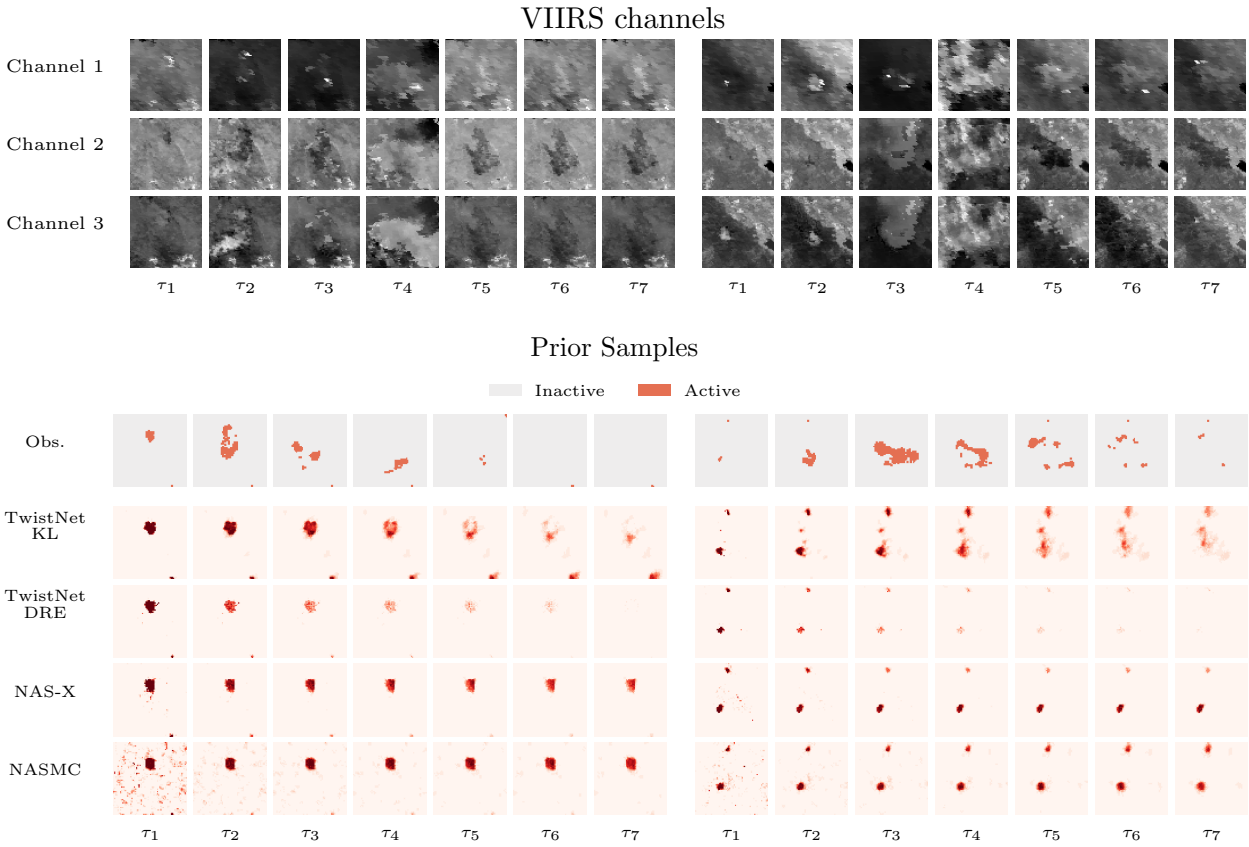

Figure 12: VIIRS covariates and empirical distribution of active fires obtained from samples of the prior dynamics, for the examples presented in Figure 5.

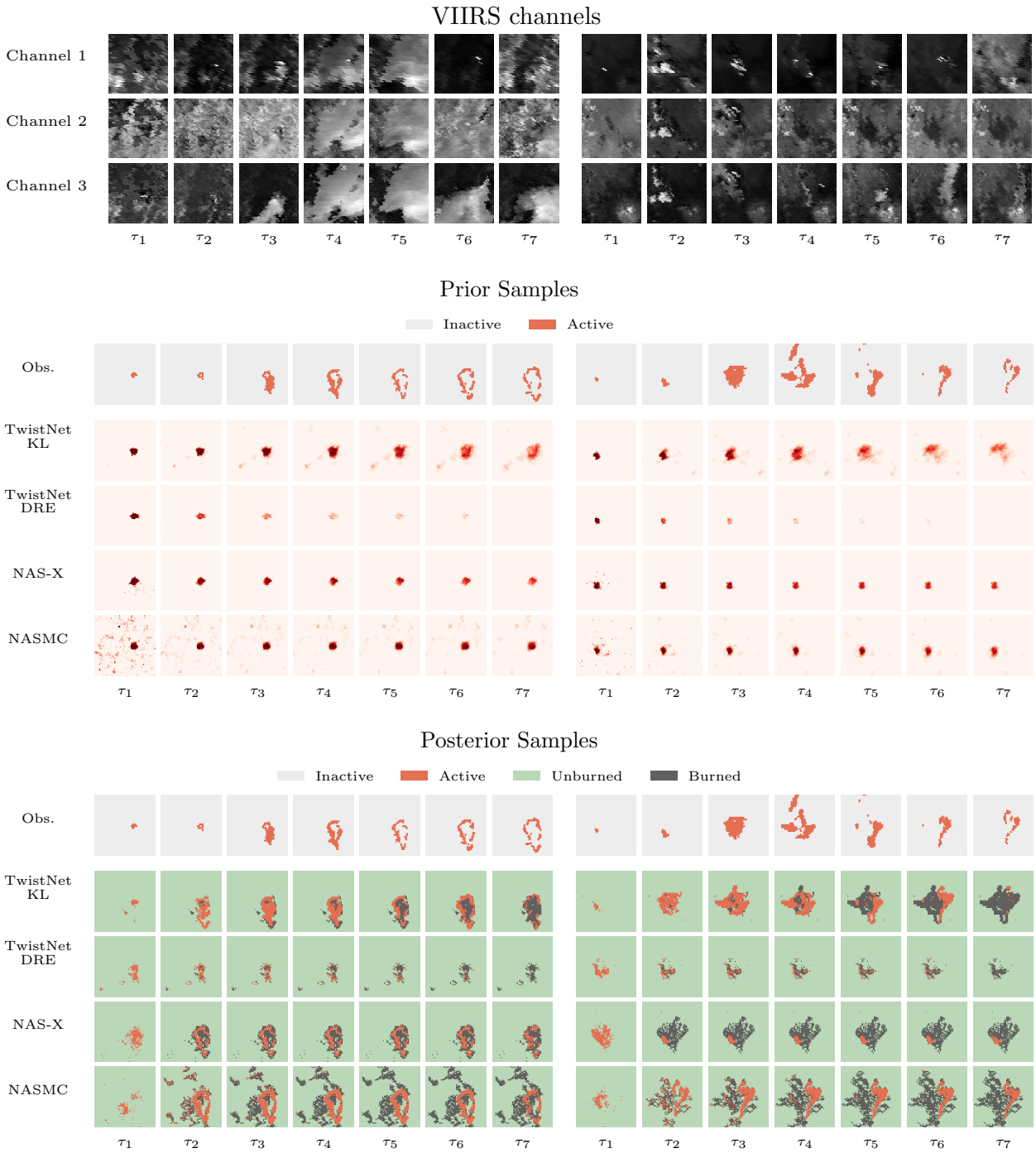

Figure 13: Third and fourth examples of wildfire trajectories.

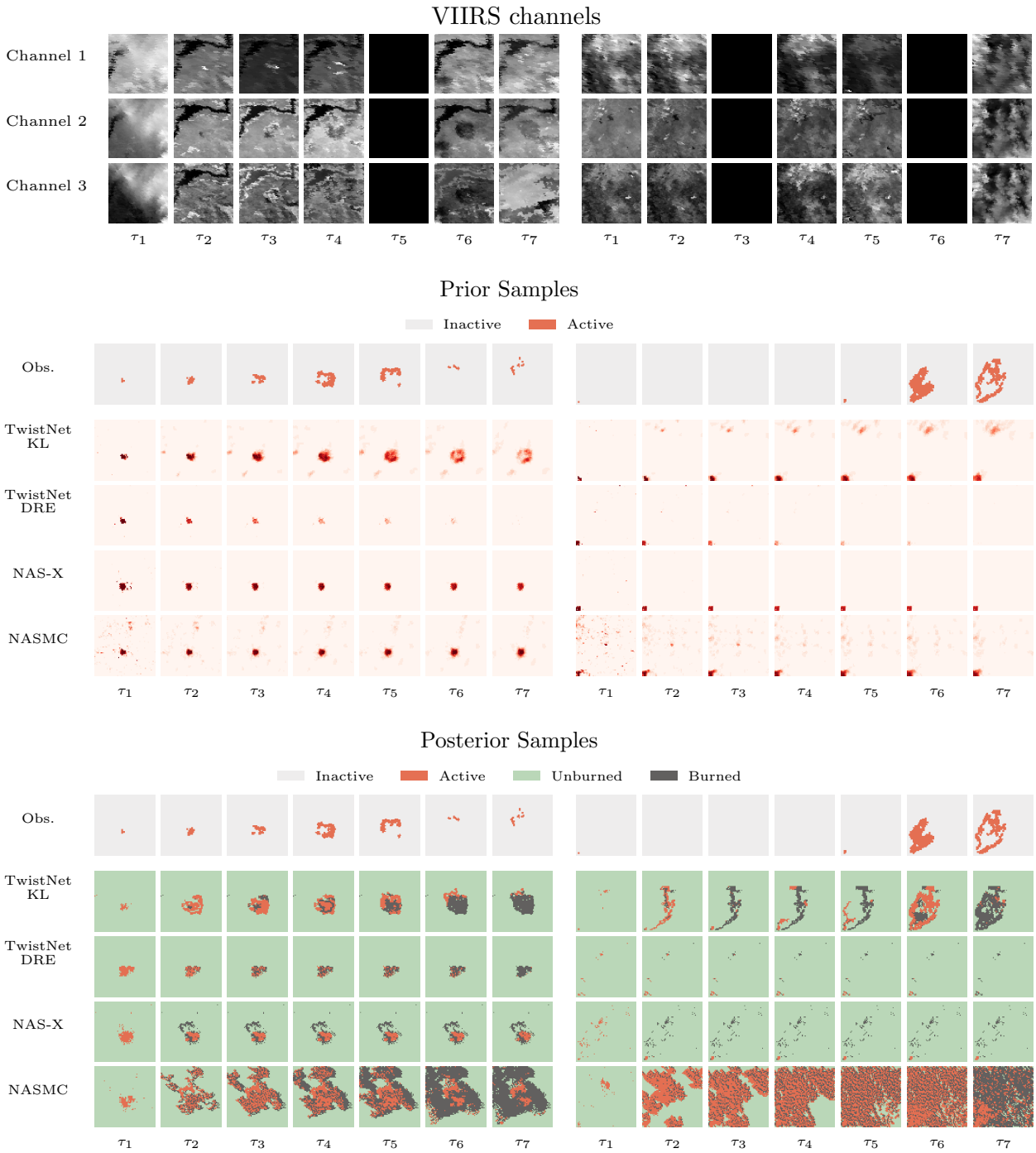

Figure 14: Fifth and sixth examples of wildfire trajectories.