A Appendix A

A.1 Model specification.

Here we provide all details about our model specification. The joint distribution for our model is

$$p(\boldsymbol{u}_{1:M}, \boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}}) = p(\boldsymbol{u}_{1:N}|\boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}})p(\boldsymbol{s}_{1:B}|\theta_{\text{dyn}})p(\theta_{\text{dyn}})p(\theta_{\text{dec}}). \tag{23}$$

Next, we specify each component in detail.

Parameter priors. The parameter priors are isotropic zero-mean multivariate normal distributions:

$$p(\theta_{\text{dyn}}) = \mathcal{N}(\theta_{\text{dyn}}|\mathbf{0}, I), \tag{24}$$

$$p(\theta_{\text{dec}}) = \mathcal{N}(\theta_{\text{dec}}|\mathbf{0}, I), \tag{25}$$

where $\mathcal{N}$ is the normal distribution, $\mathbf{0}$ is a zero vector, and I is the identity matrix, both have an appropriate dimensionality dependent on the number of encoder and dynamics parameters.

Continuity prior. We define the continuity prior as

$$p(\boldsymbol{s}_{1:B}|\theta_{\text{dyn}}) = p(\boldsymbol{s}_1)\prod_{b=2}^{B} p(\boldsymbol{s}_b|\boldsymbol{s}_{b-1}, \theta_{\text{dyn}}), \tag{26}$$

$$= \left[\prod_{j=1}^{N} p(\boldsymbol{s}_1^j)\right]\left[\prod_{b=2}^{B}\prod_{j=1}^{N} p(\boldsymbol{s}_b^j|\boldsymbol{s}_{b-1}, \theta_{\text{dyn}})\right], \tag{27}$$

$$= \left[\prod_{j=1}^{N} \mathcal{N}(\boldsymbol{s}_1^j|\mathbf{0}, I)\right]\left[\prod_{b=2}^{B}\prod_{j=1}^{N} \mathcal{N}\left(\boldsymbol{s}_b^j|\boldsymbol{z}(t_{[b]}, \boldsymbol{x}_j; t_{[b-1]}, \boldsymbol{s}_{b-1}, \theta_{\text{dyn}}), \sigma_c^2 I\right)\cdot\right], \tag{28}$$

where $\mathcal{N}$ is the normal distribution, $\mathbf{0} \in \mathbb{R}^d$ is a zero vector, $I \in \mathbb{R}^{d\times d}$ is the identity matrix, and $\sigma_c \in \mathbb{R}$ is the parameter controlling the strength of the prior. Smaller values of σ_c tend to produce smaller gaps between the sub-trajectories.

Observation model

$$p(\boldsymbol{u}_{1:N}|\boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}}) = \prod_{b=1}^{B}\prod_{i\in\mathcal{I}_b}\prod_{j=1}^{N} p(\boldsymbol{u}_i^j|\boldsymbol{s}_b, \theta_{\text{dyn}}, \theta_{\text{dec}}) \tag{29}$$

$$= \prod_{b=1}^{B}\prod_{i\in\mathcal{I}_b}\prod_{j=1}^{N} p(\boldsymbol{u}_i^j|g_{\theta_{\text{dec}}}(\boldsymbol{z}(t_i, \boldsymbol{x}_j; t_{[b]}, \boldsymbol{s}_b, \theta_{\text{dyn}}))) \tag{30}$$

$$= \prod_{b=1}^{B}\prod_{i\in\mathcal{I}_b}\prod_{j=1}^{N} \mathcal{N}(\boldsymbol{u}_i^j|g_{\theta_{\text{dec}}}(\boldsymbol{z}(t_i, \boldsymbol{x}_j; t_{[b]}, \boldsymbol{s}_b, \theta_{\text{dyn}})), \sigma_u^2 I), \tag{31}$$

where $\mathcal{N}$ is the normal distribution, σ_u^2 is the observation noise variance, and $I \in \mathbb{R}^{D\times D}$ is the identity matrix. Note again that $\boldsymbol{z}(t_i, \boldsymbol{x}_j; t_{[b]}, \boldsymbol{s}_b, \theta_{\text{dyn}})$ above equals the ODE forward solution $\text{ODESolve}(t_i; t_{[b]}, \boldsymbol{s}_b, \theta_{\text{dyn}})$ at grid location $\boldsymbol{x}_j$.

A.2 Approximate posterior specification.

Here we provide all details about the approximate posterior. We define the approximate posterior as

$$q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) = q(\theta_{\text{dyn}})q(\theta_{\text{dec}})q(\boldsymbol{s}_{1:B}) = q_{\boldsymbol{\psi}_{\text{dyn}}}(\theta_{\text{dyn}})q_{\boldsymbol{\psi}_{\text{dec}}}(\theta_{\text{dec}})\prod_{b=1}^{B}\prod_{j=1}^{N} q_{\boldsymbol{\psi}_b^j}(\boldsymbol{s}_b^j). \tag{32}$$

Next, we specify each component in detail.

Dynamics parameters posterior. We define $q_{\boldsymbol{\psi}_{\text{dyn}}}(\theta_{\text{dyn}})$ as

$$q_{\boldsymbol{\psi}_{\text{dyn}}}(\theta_{\text{dyn}}) = \mathcal{N}(\theta_{\text{dyn}} | \boldsymbol{\gamma}_{\text{dyn}}, \operatorname{diag}(\boldsymbol{\tau}^2_{\text{dyn}})), \tag{33}$$

where $\boldsymbol{\gamma}_{\text{dyn}}$ and $\boldsymbol{\tau}^2_{\text{dyn}}$ are vectors with an appropriate dimension (dependent on the number of dynamics parameters), and $\operatorname{diag}(\boldsymbol{\tau}^2_{\text{dyn}})$ is a matrix with $\boldsymbol{\tau}^2_{\text{dyn}}$ on the diagonal. We define the vector of variational parameters as $\boldsymbol{\psi}_{\text{dyn}} = (\boldsymbol{\gamma}_{\text{dyn}}, \boldsymbol{\tau}^2_{\text{dyn}})$. We optimize directly over $\boldsymbol{\psi}_{\text{dyn}}$ and initialize $\boldsymbol{\gamma}_{\text{dyn}}$ using Xavier (Glorot and Bengio, 2010) initialization, while $\boldsymbol{\tau}_{\text{dyn}}$ is initialized with each element equal to $9 \cdot 10^{-4}$.

Decoder parameters posterior. We define $q_{\boldsymbol{\psi}_{\text{dec}}}(\theta_{\text{dec}})$ as

$$q_{\boldsymbol{\psi}_{\text{dec}}}(\theta_{\text{dec}}) = \mathcal{N}(\theta_{\text{dec}} | \boldsymbol{\gamma}_{\text{dec}}, \operatorname{diag}(\boldsymbol{\tau}^2_{\text{dec}})), \tag{34}$$

where $\boldsymbol{\gamma}_{\text{dec}}$ and $\boldsymbol{\tau}^2_{\text{dec}}$ are vectors with an appropriate dimension (dependent on the number of decoder parameters), and $\operatorname{diag}(\boldsymbol{\tau}^2_{\text{dec}})$ is a matrix with $\boldsymbol{\tau}^2_{\text{dec}}$ on the diagonal. We define the vector of variational parameters as $\boldsymbol{\psi}_{\text{dec}} = (\boldsymbol{\gamma}_{\text{dec}}, \boldsymbol{\tau}^2_{\text{dec}})$. We optimize directly over $\boldsymbol{\psi}_{\text{dec}}$ and initialize $\boldsymbol{\gamma}_{\text{dec}}$ using Xavier (Glorot and Bengio, 2010) initialization, while $\boldsymbol{\tau}_{\text{dec}}$ is initialized with each element equal to $9 \cdot 10^{-4}$.

Shooting variables posterior. We define $q_{\boldsymbol{\psi}_b^j}(\boldsymbol{s}_b^j)$ as

$$q_{\boldsymbol{\psi}_b^j}(\boldsymbol{s}_b^j) = \mathcal{N}(\boldsymbol{s}_b^j | \boldsymbol{\gamma}_b^j, \operatorname{diag}([\boldsymbol{\tau}_b^j]^2))), \tag{35}$$

where the vectors $\boldsymbol{\gamma}_b^j, \boldsymbol{\tau}_b^j \in \mathbb{R}^d$ are returned by the encoder $h_{\theta_{\text{enc}}}$, and $\operatorname{diag}([\boldsymbol{\tau}_b^j]^2)$ is a matrix with $[\boldsymbol{\tau}_b^j]^2$ on the diagonal. We define the vector of variational parameters as $\boldsymbol{\psi}_b^j = (\boldsymbol{\gamma}_b^j, [\boldsymbol{\tau}_b^j])$. Because the variational inference for the shooting variables is amortized, our model is trained w.r.t. the parameters of the encoder network, θ_{enc}.

B Appendix B

B.1 Derivation of ELBO.

For our model and the choice of the approximate posterior the ELBO can be written as

$$\begin{aligned}
\mathcal{L} &= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{p(\boldsymbol{u}_{1:M}, \boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}})}{q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(36)\\
&= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{p(\boldsymbol{u}_{1:M} | \boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}}) p(\boldsymbol{s}_{1:B} | \theta_{\text{dyn}}) p(\theta_{\text{dyn}}) p(\theta_{\text{dec}})}{q(\boldsymbol{s}_{1:B}) q(\theta_{\text{dyn}}) q(\theta_{\text{dec}})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(37)\\
&= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln p(\boldsymbol{u}_{1:M} | \boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}}) d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(38)\\
&\quad - \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{q(\boldsymbol{s}_{1:B})}{p(\boldsymbol{s}_{1:B} | \theta_{\text{dyn}})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(39)\\
&\quad - \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{q(\theta_{\text{dyn}})}{p(\theta_{\text{dyn}})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(40)\\
&\quad - \int q(\theta_{\text{dec}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{q(\theta_{\text{dec}})}{p(\theta_{\text{dec}})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} &(41)\\
&= \mathcal{L}_1 - \mathcal{L}_2 - \mathcal{L}_3 - \mathcal{L}_4. &(42)
\end{aligned}$$

Next, we will look at each term $\mathcal{L}_i$ separately.

$$\mathcal{L}_1 = \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln p(\boldsymbol{u}_{1:M} | \boldsymbol{s}_{1:B}, \theta_{\text{dyn}}, \theta_{\text{dec}}) d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{43}$$

$$= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[\prod_{b=1}^{B} \prod_{i \in \mathcal{I}_b} \prod_{j=1}^{N} p(\boldsymbol{u}_i^j | \boldsymbol{s}_b, \theta_{\text{dyn}}, \theta_{\text{dec}}) \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{44}$$

$$= \sum_{b=1}^{B} \sum_{i \in \mathcal{I}_b} \sum_{j=1}^{N} \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[p(\boldsymbol{u}_i^j | \boldsymbol{s}_b, \theta_{\text{dyn}}, \theta_{\text{dec}}) \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{45}$$

$$= \sum_{b=1}^{B} \sum_{i \in \mathcal{I}_b} \sum_{j=1}^{N} \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_b) \ln \left[p(\boldsymbol{u}_i^j | \boldsymbol{s}_b, \theta_{\text{dyn}}, \theta_{\text{dec}}) \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_b \tag{46}$$

$$= \sum_{b=1}^{B} \sum_{i \in \mathcal{I}_b} \sum_{j=1}^{N} \mathbb{E}_{q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_b)} \ln \left[p(\boldsymbol{u}_i^j | \boldsymbol{s}_b, \theta_{\text{dyn}}, \theta_{\text{dec}}) \right]. \tag{47}$$

$$\mathcal{L}_2 = \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \frac{q(\boldsymbol{s}_{1:B})}{p(\boldsymbol{s}_{1:B} | \theta_{\text{dyn}})} d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{48}$$

$$= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[\frac{q(\boldsymbol{s}_1)}{p(\boldsymbol{s}_1)} \prod_{b=2}^{B} \frac{q(\boldsymbol{s}_b)}{p(\boldsymbol{s}_b | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}})} \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{49}$$

$$= \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[\prod_{j=1}^{N} \frac{q(\boldsymbol{s}_1^j)}{p(\boldsymbol{s}_1^j)} \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{50}$$

$$+ \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[\prod_{b=2}^{B} \prod_{j=1}^{N} \frac{q(\boldsymbol{s}_b^j)}{p(\boldsymbol{s}_b^j | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}})} \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{51}$$

$$= \sum_{j=1}^{N} \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \ln \left[\frac{q(\boldsymbol{s}_1^j)}{p(\boldsymbol{s}_1^j)} \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{52}$$

$$+ \sum_{b=2}^{B} \int q(\theta_{\text{dyn}}, \theta_{\text{dec}}, \boldsymbol{s}_{1:B}) \sum_{j=1}^{N} \ln \left[\frac{q(\boldsymbol{s}_b^j)}{p(\boldsymbol{s}_b^j | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}})} \right] d\theta_{\text{dyn}} d\theta_{\text{dec}} d\boldsymbol{s}_{1:B} \tag{53}$$

$$= \sum_{j=1}^{N} \int q(\boldsymbol{s}_1^j) \ln \left[\frac{q(\boldsymbol{s}_1^j)}{p(\boldsymbol{s}_1^j)} \right] d\boldsymbol{s}_1^j \tag{54}$$

$$+ \sum_{b=2}^{B} \int q(\theta_{\text{dyn}}, \boldsymbol{s}_{b-1}, \boldsymbol{s}_b) \sum_{j=1}^{N} \ln \left[\frac{q(\boldsymbol{s}_b^j)}{p(\boldsymbol{s}_b^j | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}})} \right] d\theta_{\text{dyn}} d\boldsymbol{s}_{b-1} d\boldsymbol{s}_b \tag{55}$$

$$= \sum_{j=1}^{N} \int q(\boldsymbol{s}_1^j) \ln \left[\frac{q(\boldsymbol{s}_1^j)}{p(\boldsymbol{s}_1^j)} \right] d\boldsymbol{s}_1^j \tag{56}$$

$$+ \sum_{b=2}^{B} \int q(\theta_{\text{dyn}}, \boldsymbol{s}_{b-1}) \sum_{j=1}^{N} \left[\int q(\boldsymbol{s}_b^j) \ln \frac{q(\boldsymbol{s}_b^j)}{p(\boldsymbol{s}_b^j | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}})} d\boldsymbol{s}_b^j \right] d\theta_{\text{dyn}} d\boldsymbol{s}_{b-1} \tag{57}$$

$$= \sum_{j=1}^{N} \text{KL} \left(q(\boldsymbol{s}_1^j) \| p(\boldsymbol{s}_1^j) \right) + \sum_{b=2}^{B} \mathbb{E}_{q(\theta_{\text{dyn}}, \boldsymbol{s}_{b-1})} \left[\sum_{j=1}^{N} \text{KL} \left(q(\boldsymbol{s}_b^j) \| p(\boldsymbol{s}_b^j | \boldsymbol{s}_{b-1}, \theta_{\text{dyn}}) \right) \right], \tag{58}$$

where KL is Kullback–Leibler (KL) divergence. Both of the KL divergences above have a closed form but the expectation w.r.t. $q(\theta_{\text{dyn}}, \boldsymbol{s}_{b-1})$ does not.

$$\mathcal{L}_3 = \text{KL}(q(\theta_{\text{dyn}}) \| p(\theta_{\text{dyn}})), \quad \mathcal{L}_4 = \text{KL}(q(\theta_{\text{dec}}) \| p(\theta_{\text{dec}})). \tag{59}$$

B.2 Computation of ELBO.

We compute the ELBO using the following algorithm:

1. Sample $\theta_{\text{dyn}}, \theta_{\text{dec}}$ from $q_{\boldsymbol{\psi}_{\text{dyn}}}(\theta_{\text{dyn}}), q_{\boldsymbol{\psi}_{\text{dec}}}(\theta_{\text{dec}})$.
2. Sample $\boldsymbol{s}_{1:B}$ by sampling each $\boldsymbol{s}_b^j$ from $q_{\boldsymbol{\psi}_b^j}(\boldsymbol{s}_b^j)$ with $\boldsymbol{\psi}_b^j = h_{\theta_{\text{enc}}}(\boldsymbol{u}[t_{[b]}, \boldsymbol{x}_j])$.
3. Compute $\boldsymbol{u}_{1:M}$ from $\boldsymbol{s}_{1:B}$ as in Equations 14-16.
4. Compute ELBO $\mathcal{L}$ (KL terms are computed in closed form, for expectations we use Monte Carlo integration with one sample).

Sampling is done using reparametrization to allow unbiased gradients w.r.t. the model parameters.

C Appendix C

C.1 Datasets.

Shallow Water. The shallow water equations are a system of partial differential equations (PDEs) that simulate the behavior of water in a shallow basin. These equations are effectively a depth-integrated version of the Navier-Stokes equations, assuming the horizontal length scale is significantly larger than the vertical length scale. Given these assumptions, they provide a model for water dynamics in a basin or similar environment, and are commonly utilized in predicting the propagation of water waves, tides, tsunamis, and coastal currents. The state of the system modeled by these equations consists of the wave height $h(t, x, y)$, velocity in the x-direction $u(t, x, y)$ and velocity in the y-direction $v(t, x, y)$. Given an initial state (h_0, u_0, v_0), we solve the PDEs on a spatial domain Ω over time interval $[0, T]$. The shallow water equations are defined as:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0, \tag{60}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g\frac{\partial h}{\partial x} = 0, \tag{61}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g\frac{\partial h}{\partial y} = 0, \tag{62}$$

where g is the gravitational constant.

We set the spatial domain Ω to be a unit square and use periodic boundary conditions. We set $T = 0.1$. The solution is evaluated at randomly selected spatial locations and time points. We use 1089 spatial locations and 25 time points. The spatial end temporal grids are the same for all trajectories. Since we are dealing with partially-observed cases, we assume that we observe only the wave height $h(t, x, y)$.

For each trajectory, we start with zero initial velocities and the initial height $h_0(x, y)$ generated as:

$$\tilde{h}_0(x, y) = \sum_{k,l=-N}^{N} \lambda_{kl} \cos(2\pi(kx + ly)) + \gamma_{kl} \sin(2\pi(kx + ly)), \tag{63}$$

$$h_0(x, y) = 1 + \frac{\tilde{h}_0(x, y) - \min(\tilde{h}_0)}{\max(\tilde{h}_0) - \min(\tilde{h}_0)}, \tag{64}$$

where $N = 3$ and $\lambda_{kl}, \gamma_{kl} \sim \mathcal{N}(0, 1)$.

The datasets used for training, validation, and testing contain 60, 20, and 20 trajectories, respectively.

We use scikit-fdiff (Cellier, 2019) to solve the PDEs.

Navier-Stokes. For this dataset we model the propagation of a scalar field (e.g., smoke concentration) in a fluid (e.g., air). The modeling is done by coupling the Navier-Stokes equations with the Boussinesq buoyancy term and the transport equation to model the propagation of the scalar field. The state of the system modeled by these equations consists of the scalar field $c(t, x, y)$, velocity in x-direction $u(t, x, y)$, velocity in y-direction $v(t, x, y)$, and pressure $p(t, x, y)$. Given an initial state

(c_0, u_0, v_0, p_0), we solve the PDEs on a spatial domain Ω over time interval $[0, T]$. The Navier-Stokes equations with the transport equation are defined as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{65}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{66}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + c, \tag{67}$$

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} - v\frac{\partial c}{\partial y} + \nu\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right), \tag{68}$$

where $\nu = 0.002$.

We set the spatial domain Ω to be a unit square and use periodic boundary conditions. We set $T = 2.0$, but drop the first 0.5 seconds due to slow dynamics during this time period. The solution is evaluated at randomly selected spatial locations and time points. We use 1089 spatial locations and 25 time points. The spatial and temporal grids are the same for all trajectories. Since we are dealing with partially-observed cases, we assume that we observe only the scalar field $c(t, x, y)$.

For each trajectory, we start with zero initial velocities and pressure, and the initial scalar field $c_0(x, y)$ is generated as:

$$\tilde{c}_0(x, y) = \sum_{k,l=-N}^{N} \lambda_{kl}\cos(2\pi(kx + ly)) + \gamma_{kl}\sin(2\pi(kx + ly)), \tag{69}$$

$$c_0(x, y) = \frac{\tilde{c}_0(x, y) - \min(\tilde{c}_0)}{\max(\tilde{c}_0) - \min(\tilde{c}_0)}, \tag{70}$$

where $N = 2$ and $\lambda_{kl}, \gamma_{kl} \sim \mathcal{N}(0, 1)$.

The datasets used for training, validation, and testing contain 60, 20, and 20 trajectories, respectively.

We use PhiFlow (Holl et al., 2020) to solve the PDEs.

Scalar Flow. This dataset, proposed by Eckert et al. (2019), consists of observations of smoke plumes rising in hot air. The observations are post-processed camera images of the smoke plumes taken from multiple views. For simplicity, we use only the front view. The dataset contains 104 trajectories, where each trajectory has 150 time points and each image has the resolution 1080 × 1920.

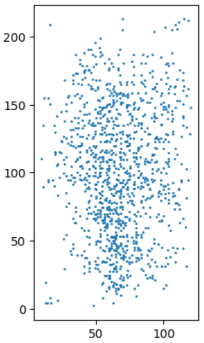


Figure 11: Spatial grid used for Scalar Flow dataset.

To reduce dimensionality of the observations we sub-sample the original spatial and temporal grids. For the temporal grid, we remove the first 50 time points, which leaves 100 time points, and then take every 4th time point, thus leaving 20 time points in total. The original 1080 × 1920 spatial grid is first down-sampled by a factor of 9 giving a new grid with resolution 120 × 213, and then the new grid is further sub-sampled based on the smoke density at each node. In particular, we compute the average smoke density at each node (averaged over time), and then sample the nodes without replacement with the probability proportional to the average smoke density (thus, nodes that have zero density most of the time are not selected). See example of a final grid in Figure 11. This gives a new grid with 1089 nodes.

We further smooth the observations by applying Gaussian smoothing with the standard deviation of 1.5 (assuming domain size 120 × 213).

We use the first 60 trajectories for training, next 20 for validation and next 20 for testing.

C.2 Model architecture and hyper-parameters.

Dynamics function. For all datasets we define $F_{\theta_{\text{dyn}}}$ as an MLP. For Shallow Water/Navier-Stokes/Scalar Flow we use 1/3/3 hidden layers with the size of 1024/512/512, respectively. We use ReLU nonlinearities.

Observation function. For all datasets we define $g_{\theta_{\text{dec}}}$ as a selector function which takes the latent state $\boldsymbol{z}(t, x) \in \mathbb{R}^d$ and returns its first component.

Encoder. Our encoder $h_{\theta_{\text{enc}}}$ consists of three function: $h_{\theta_{\text{spatial}}}$, $h_{\theta_{\text{temporal}}}$, and $h_{\theta_{\text{read}}}$. The spatial aggregation function $h_{\theta_{\text{spatial}}}$ is a linear mapping to $\mathbb{R}^{128}$. The temporal aggregation function $h_{\theta_{\text{temporal}}}$ is a stack of transformer layers with temporal attention and continuous relative positional encodings (Iakovlev et al., 2023). For all datasets, we set the number of transformer layers to 6. Finally, the variational parameter readout function $h_{\theta_{\text{read}}}$ is a mapping defined as

$$\boldsymbol{\psi}_b^j = h_{\theta_{\text{read}}}(\alpha^{\text{T}}_{[b]}) = \begin{pmatrix} \boldsymbol{\gamma}_b^j \\ \boldsymbol{\tau}_b^j \end{pmatrix} = \begin{pmatrix} \text{Linear}(\alpha^{\text{T}}_{[b]}) \\ \exp(\text{Linear}(\alpha^{\text{T}}_{[b]})) \end{pmatrix}, \tag{71}$$

where Linear is a linear layer (different for each line), and $\boldsymbol{\gamma}_b^j$ and $\boldsymbol{\tau}_b^j$ are the variational parameters discussed in Appendix A.

Spatial and temporal neighborhoods. We use the same spatial neighborhoods $\mathcal{N}_{\text{S}}(\boldsymbol{x})$ for both the encoder and the dynamics function. We define $\mathcal{N}_{\text{S}}(\boldsymbol{x})$ as the set of points consisting of the point $\boldsymbol{x}$ and points on two concentric circles centered at $\boldsymbol{x}$, with radii r and $r/2$, respectively. Each circle contains 8 points spaced 45 degrees apart (see Figure 12 (right)). The radius r is set to 0.1. For Shallow Water/Navier-Stokes/Scalar Flow the size of temporal neighborhood (δ_T) is set to 0.1/0.1/0.2, respectively.

Multiple Shooting. For Shallow Water/Navier-Stokes/Scalar Flow we split the full training trajectories into 4/4/19 sub-trajectories, or, equivalently, have the sub-trajectory length of 6/6/2.

C.3 Training, validation, and testing setup.

Data preprocessing. We scale the temporal grids, spatial grids, and observations to be within the interval $[0, 1]$.

Training. We train our model for 20000 iterations using Adam (Kingma and Ba, 2017) optimizer with constant learning rate 3e-4 and linear warmup for 200 iterations. The latent spatiotemporal dynamics are simulated using differentiable ODE solvers from the torchdiffeq package (Chen, 2018) (we use dopri5 with rtol=1e-3, atol=1e-4, no adjoint). The batch size is 1.

Validation. We use validation set to track the performance of our model during training and save the parameters that produce the best validation performance. As performance measure we use the mean absolute error at predicting the full validation trajectories given some number of initial observations. For Shallow Water/Navier-Stokes/Scalar Flow we use the first 5/5/10 observations. The predictions are made by taking one sample from the posterior predictive distribution (see Appendix C.4 for details).

Testing. Testing is done similarly to validation, except that as the prediction we use an estimate of the expected value of the posterior predictive distribution (see Appendix C.4 for details).

C.4 Forecasting.

Given initial observations $\tilde{\boldsymbol{u}}_{1:m}$ at time points $t_{1:m}$, we predict the future observation $\tilde{\boldsymbol{u}}_n$ at a time point $t_n > t_m$ as the expected value of the approximate posterior predictive distribution:

$$p(\tilde{\boldsymbol{u}}_n | \tilde{\boldsymbol{u}}_{1:m}, \boldsymbol{u}_{1:M}) \approx \int p(\tilde{\boldsymbol{u}}_n | \tilde{\boldsymbol{s}}_m, \theta_{\text{dyn}}, \theta_{\text{dec}}) q(\tilde{\boldsymbol{s}}_m) q(\theta_{\text{dyn}}) q(\theta_{\text{dec}}) d\tilde{\boldsymbol{s}}_m d\theta_{\text{dyn}} d\theta_{\text{dec}}. \tag{72}$$

The expected value is estimated via Monte Carlo integration, so the algorithm for predicting $\tilde{\boldsymbol{u}}_n$ is:

1. Sample $\theta_{\text{dyn}}, \theta_{\text{dec}}$ from $q(\theta_{\text{dyn}}), q(\theta_{\text{dec}})$.
2. Sample $\tilde{\boldsymbol{s}}_m$ from $q(\tilde{\boldsymbol{s}}_m) = \prod_{j=1}^{N} q_{\boldsymbol{\psi}_m^j}(\tilde{\boldsymbol{s}}_m^j)$, where the variational parameters $\boldsymbol{\psi}_m^j$ are given by the encoder $h_{\theta_{\text{enc}}}$ operating on the initial observations $\tilde{\boldsymbol{u}}_{1:m}$ as $\boldsymbol{\psi}_m^j = h_{\theta_{\text{enc}}}(\tilde{\boldsymbol{u}}[t_m, \boldsymbol{x}_j])$.
3. Compute the latent state $\tilde{\boldsymbol{z}}(t_n) = \boldsymbol{z}(t_n; t_m, \tilde{\boldsymbol{s}}_m, \theta_{\text{dyn}})$.
4. Sample $\tilde{\boldsymbol{u}}_n$ by sampling each $\tilde{\boldsymbol{u}}_n^j$ from $\mathcal{N}(\tilde{\boldsymbol{u}}_n^j | g_{\theta_{\text{dec}}}(\tilde{\boldsymbol{z}}(t_n, \boldsymbol{x}_j))), \sigma_u^2 I)$.
5. Repeat steps 1-4 n times and average the predictions (we use $n = 10$).

C.5 Model comparison setup.

DINo. We use the official implementation of DINo (Yin et al., 2023). The encoder is an MLP with 3 hidden layers, 512 neurons each, and Swish non-linearities. The code dimension is 100. The dynamics function is an MLP with 3 hidden layers, 512 neurons each, and Swish non-linearities. The decoder has 3 layers and 64 channels.

MAgNet. We use the official implementation of MAgNet (Boussif et al., 2022). We use the graph neural network variant of the model. The number of message-passing steps is 5. All MLPs have 4 layers with 128 neurons each in each layer. The latent state dimension is 128.

D Appendix D

D.1 Spatiotemporal neighborhood shapes and sizes.

Here we investigate the effect of changing the shape and size of spatial and temporal neighborhoods used by the encoder and dynamics functions. We use the default hyperparameters discussed in Appendix C and change only the neighborhood shape or size. A neighborhood size of zero implies no spatial/temporal aggregation.

Initially, we use the original circular neighborhood displayed in Figure 12 for both encoder and dynamics function and change only its size (radius). The results are presented in Figures 13a and 13b. In Figure 13a, it is surprising to see very little effect from changing the encoder's spatial neighborhood size. A potential explanation is that the dynamics function shares the spatial aggregation task with the encoder. However, the results in Figure 13b are more intuitive, displaying a U-shaped curve for the test MAE, indicating the importance of using spatial neighborhoods of appropriate size. Interestingly, the best results tend to be achieved with relatively large neighborhood sizes. Similarly, Figure 13c shows U-shaped curves for the encoder's temporal neighborhood size, suggesting that latent state inference benefits from utilizing local temporal information.

We then examine the effect of changing the shape of the dynamics function's spatial neighborhood. We use $n$circle neighborhoods, which consist of n equidistant concentric circular neighborhoods (see examples in Figure 12). Effectively, we maintain a fixed neighborhood size while altering its density. The results can be seen in Figure 14. We find that performance does not significantly improve when using denser (and presumably more informative) spatial neighborhoods, indicating that accurate predictions only require a relatively sparse neighborhood with appropriate size.

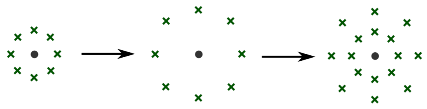
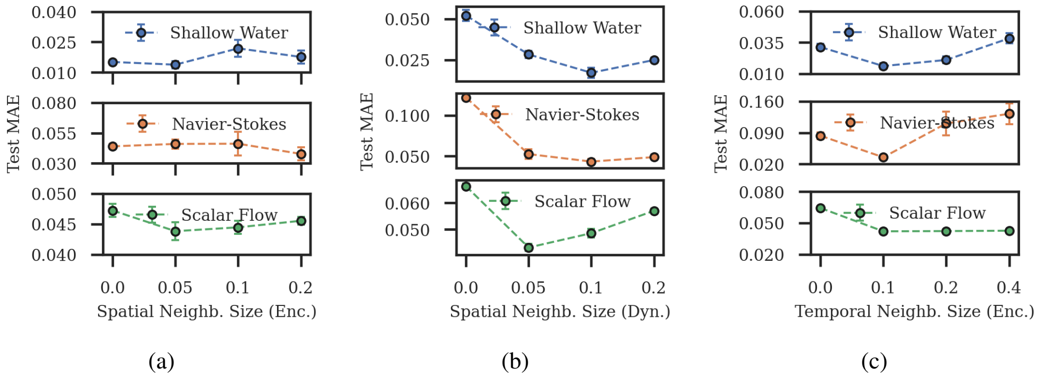


Figure 12: **Left:** original circular neighborhood (1circle). **Center:** circular neighborhood with increased size. **Right:** circular neighborhood of a different shape (2circle).

D.2 Multiple shooting.

Here we demonstrate the effect of using multiple shooting for model training. In Figure 15 (left), we vary the sub-trajectory length (longer sub-trajectories imply more difficult training) and plot the test errors for each sub-trajectory length. We observe that in all cases, the best results are achieved when the sub-trajectory length is considerably smaller than the full trajectory length. In Figure 15 (right)



(a) (b) (c)

Figure 13: **(a),(b):** Test MAE vs spatial neighborhood sizes of the encoder and dynamics function, respectively. **(c):** Test MAE vs temporal neighborhood size of the encoder. Note that the spatial and temporal domains are normalized, so their largest size in any dimension is 1.

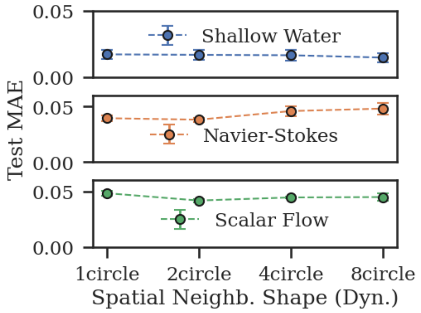


Figure 14: Test MAE vs spatial neighborhood shape.

we further show the training times, and as can be seen multiple shooting allows to noticeably reduce the training times.

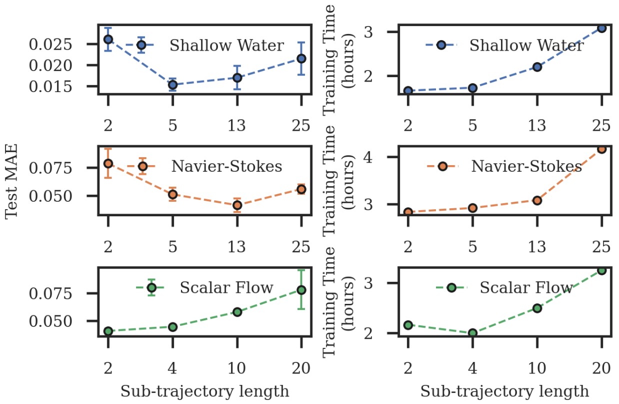


Figure 15: Test MAE vs training sub-trajectory length.

E Appendix E

Noisy Data. Here we show the effect of observation noise on our model and compare the results against other models. We train all models with data noise of various strengths, and then compute test MAE on noiseless data (we still use noisy data to infer the initial state at test time). Figure 16 shows that our model can manage noise strength up to 0.1 without significant drops in performance. Note that all observations are in the range $[0, 1]$.

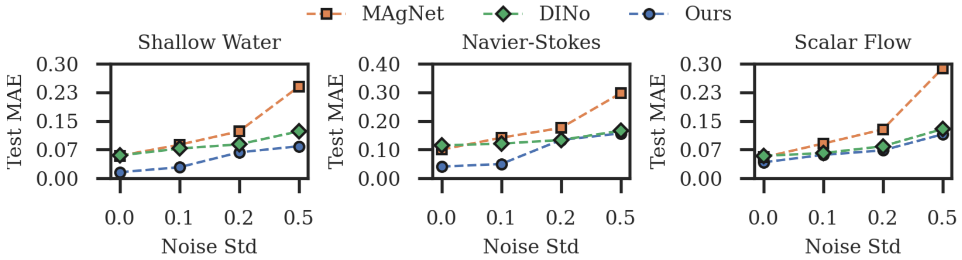


Figure 16: Test MAE vs observation noise σ_u.