PAIRWISE CONSISTENCY MODELS FOR BAYESIAN TRANSPORT AND VARIATIONAL DATA ASSIMILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian inference in high-dimensional dynamical systems is traditionally approached through either score-based diffusion models or variational data assimilation. Although developed independently, both rely on the numerical integration of continuous Bayesian transport dynamics, making inference computationally expensive. We introduce a unified pairwise consistency framework that learns finite-time Bayesian transport operators between arbitrary solver times instead of infinitesimal transport fields. By enforcing pairwise consistency, the proposed approach preserves the semigroup structure of the underlying flow while avoiding explicit integration of the transport dynamics. This perspective naturally generalizes consistency models beyond diffusion processes and extends to variational inference without requiring an explicit forward diffusion model. The resulting framework encompasses supervised diffusion learning, self-consistent variational optimization, and hybrid diffusion–physical transport within a common mathematical formulation. Experiments on synthetic and geophysical inverse problems demonstrate efficient few-step reconstruction, sequential forecasting, and uncertainty-aware posterior sampling, showing that finite-time transport learning provides an effective alternative to conventional score-based integration.

1 INTRODUCTION

Generative diffusion models and variational data assimilation (4DVar) are often viewed as distinct approaches to Bayesian inference in dynamical systems. Diffusion models generate posterior samples by reversing a stochastic diffusion process (Ho et al., 2020; Song et al., 2021), whereas variational data assimilation estimates the most likely trajectory by minimizing a Bayesian objective (Courtier et al., 1994; Rabier et al., 2000; Asch et al., 2016). Despite these different formulations, both aim at approximating the posterior distribution of the hidden state conditioned on noisy observations.

Consider the state-space model

$$\begin{cases} \mathbf{x}_{k+1} = \mathcal{M}(\mathbf{x}_k) + \boldsymbol{\eta}_k, \\ \mathbf{y}_k = \mathcal{H}(\mathbf{x}_k) + \boldsymbol{\varepsilon}_k, \end{cases}$$

where $\mathbf{x}_k$ and $\mathbf{y}_k$ denote the latent state and observations at physical time k, $\mathcal{M}$ and $\mathcal{H}$ the dynamical and observation operators, and $\boldsymbol{\eta}_k$ and $\boldsymbol{\varepsilon}_k$ the corresponding model and observation errors. Bayesian inference then aims to characterize

$$p(\mathbf{x}_{0:N} \mid \mathbf{y}_{0:N}),$$

rather than only a single optimal trajectory.

A convenient viewpoint is to interpret Bayesian inference as learning a transport from a simple reference distribution to the posterior (Dashti and Stuart, 2017). Let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and let

$$(T_\theta)_{\#}\mathcal{N}(\mathbf{0}, \mathbf{I}) \approx p(\mathbf{x}_{0:N} \mid \mathbf{y}_{0:N}), \tag{1}$$

where $(\cdot)_{\#}$ denotes the pushforward operator. Once T_θ is learned, posterior expectations, uncertainty estimates and conditional simulations follow directly.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Modern generative models differ mainly in how they parameterize this transport. Diffusion models, Continuous Normalizing Flows and Flow Matching learn a continuous transport field (Ho et al., 2020; Song et al., 2021; Chen et al., 2018; Lipman et al., 2023; Albergo et al., 2023)

$$\frac{d\mathbf{x}^{(t)}}{dt} = V_\theta\left(\mathbf{x}^{(t)}, t, \mathbf{y}_{0:N}\right), \qquad t \in [0,1], \tag{2}$$

whose associated density satisfies the continuity equation

$$\frac{\partial p_t}{\partial t} + \nabla_{\mathbf{x}} \cdot (V_\theta p_t) = 0. \tag{3}$$

Inference is then obtained by numerically integrating this vector field. Although highly successful, this still requires tens to hundreds of solver evaluations.

The key idea of this paper is to learn finite-time Bayesian transport operators instead of infinitesimal transport fields. Let

$$\Phi_{t\to t'}(\mathbf{x}) = \mathbf{x}^{(t)} + \int_t^{t'} V(\mathbf{x}^{(s)}, s)\, ds \tag{4}$$

denote the exact flow map. Rather than approximating the local velocity V_θ, we directly learn

$$g_\theta(\mathbf{x}^{(t)}, t, t') \approx \Phi_{t\to t'}(\mathbf{x}^{(t)}),$$

that is, the finite-time transport between arbitrary solver times. Since the exact flow satisfies the semigroup property

$$\Phi_{t\to t''} = \Phi_{t'\to t''} \circ \Phi_{t\to t'}, \tag{5}$$

our objective enforces the same property on the learned transport operators. This pairwise consistency formulation generalizes recent consistency models from endpoint prediction to arbitrary finite-time Bayesian transport.

Interestingly, the same transport viewpoint naturally arises in variational data assimilation. Under the Bayesian formulation of inverse problems (Tarantola, 2005; Stuart, 2010; Dashti and Stuart, 2017), inference can be formulated as the minimization of the variational objective

$$\mathcal{J}(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{x} \mid \mathbf{y}),$$

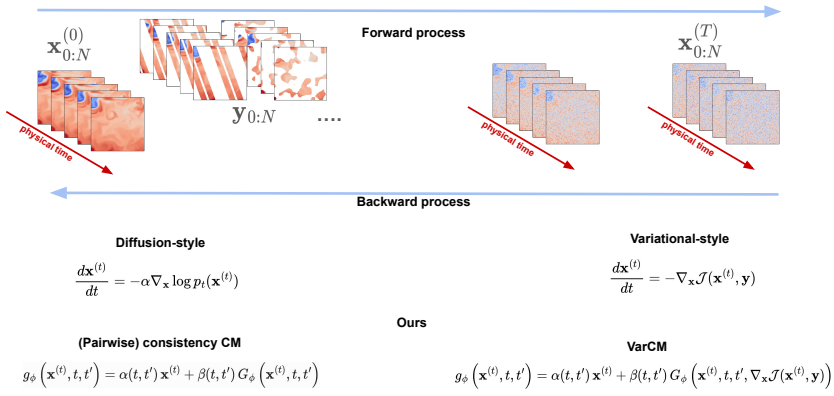
whose negative gradient defines the posterior transport direction,

$$V(\mathbf{x}) = -\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}, \mathbf{y}).$$

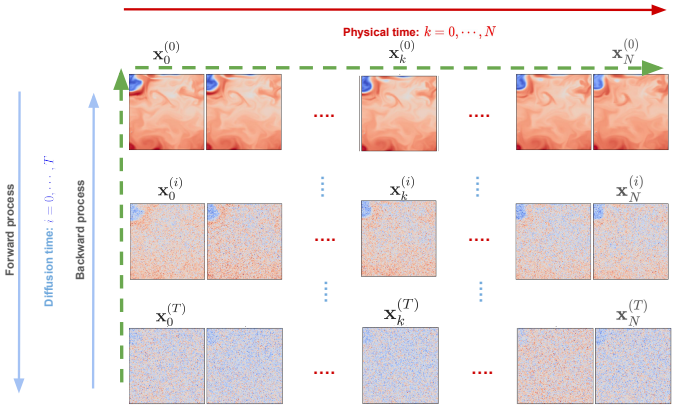
Classical 4DVar algorithms recover the posterior mode by iteratively integrating this gradient field through successive optimization updates (Courtier et al., 1994; Rabier et al., 2000; Asch et al., 2016). Likewise, diffusion models recover the target distribution by integrating the reverse-time score dynamics (Song et al., 2021), while Flow Matching integrates a learned velocity field (Lipman et al., 2023). Although originating from different communities, these approaches therefore share the same computational bottleneck: posterior inference relies on the numerical integration of a continuous transport field.

Recent methods attempt to accelerate this iterative process from complementary perspectives. Consistency Models learn finite-time approximations of the reverse diffusion dynamics (Song et al., 2023; Song and Dhariwal, 2024; Heek et al., 2024), whereas 4DVarNet learns neural corrections that accelerate variational optimization while preserving the structure of the underlying Bayesian inverse problem (Fablet et al., 2021). Our starting point is that both approaches can be interpreted as different approximations of the same underlying Bayesian transport dynamics.

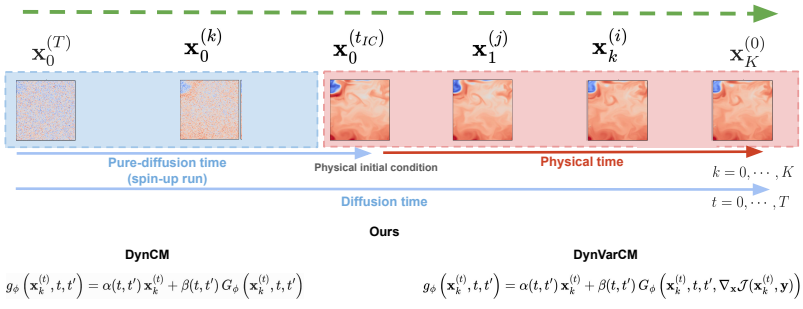
In this work, we develop a unified framework for Bayesian posterior transport that bridges diffusion models, consistency models, variational data assimilation, and physical dynamical systems. Rather than learning infinitesimal transport fields, we directly learn finite-time transport operators between arbitrary solver times. This perspective provides a common formulation for deterministic optimization, posterior sampling, uncertainty quantification, and physical forecasting.



(a) Connection between diffusion models and variational optimization



(b) Diffusion time interpretation and solver-time alignment.



(c) Physical-time interpretation of the learned consistency regime.

Figure 1: Interpretation of diffusion time as solver and physical time. The green trajectory illustrates how diffusion over a full spatio-temporal tensor (a) is reformulated as a sequential process (b): reconstruction of a physically consistent initial state followed by physical time integration.

Our contributions are fourfold:

- **Pairwise Consistency Models (Pairwise CM).** Building on the two-time formulation of Consistency Trajectory Models (Kim et al., 2024) and related finite-time flow-map perspectives (Boffi et al., 2024), we formulate pairwise consistency through the lens of Bayesian transport. The operator
$$g_\theta(\mathbf{x}^{(t)}, t, t', \mathbf{y}_{0:N})$$
represents finite-time conditional transport between arbitrary solver times rather than an infinitesimal transport field. This provides the common transport primitive on which the following formulations are built.
- **Variational Consistency Models (VarCM).** We extend pairwise consistency beyond prescribed diffusion trajectories by removing the requirement for an explicit forward diffusion process. Instead, we jointly learn
$$g_\theta\left(\mathbf{x}^{(t)}, t, t', \nabla_{\mathbf{x}} \mathcal{J}_\phi\left(\mathbf{x}^{(t)}, \mathbf{y}_{0:N}\right)\right),$$
where $\mathcal{J}_\phi$ defines a learned variational objective and g_θ approximates the finite-time transport induced by its gradient flow through self-consistency constraints. This establishes a direct bridge between consistency learning and learned variational data assimilation.
- **Dynamic Consistency Models (DynCM).** We introduce a unified diffusion–physical timeline in which solver time is aligned with physical time after an initial spin-up phase. The learned transport operator
$$g_\theta\left(\mathbf{x}_k^{(t)}, t, t', \mathbf{y}_{0:N}\right)$$
therefore models posterior initialization and physical propagation within a single continuous transport framework.
- **Dynamic Variational Consistency Models (DynVarCM).** Finally, we combine variational and dynamic consistency by learning
$$g_\theta\left(\mathbf{x}_k^{(t)}, t, t', \nabla_{\mathbf{x}} \mathcal{J}_\phi\left(\mathbf{x}_k^{(t)}, \mathbf{y}_{0:N}\right)\right)$$
along the unified solver–physical timeline. This formulation jointly learns the posterior geometry and its associated finite-time transport, enabling deterministic reconstruction, conditional posterior simulation, uncertainty quantification, and few-step inference within a common framework.

Taken together, these formulations extend finite-time consistency transport from prescribed generative trajectories to learned variational and physical dynamics. They provide a common perspective connecting score-based diffusion models (Ho et al., 2020; Song et al., 2021), Flow Matching (Lipman et al., 2023; Albergo et al., 2023), Consistency Models and finite-time trajectory models (Song et al., 2023; Song and Dhariwal, 2024; Heek et al., 2024; Kim et al., 2024; Boffi et al., 2024), and variational data assimilation (Courtier et al., 1994; Rabier et al., 2000; Fablet et al., 2021).

Experiments on generative modeling and geophysical inverse problems evaluate these formulations in terms of posterior reconstruction, uncertainty estimation, conditional simulation, and few-step inference.

2 Background

Both score-based diffusion models and variational data assimilation perform Bayesian inference by integrating a transport field toward a target posterior distribution. Although they originate from different communities, they differ primarily in the score that drives the dynamics.

Score-based diffusion. Diffusion models generate samples by integrating the reverse-time dynamics of a stochastic diffusion process (Ho et al., 2020; Song et al., 2021). For the probability-flow formulation, inference follows the conditional score field

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)} \mid \mathbf{y}),$$

which transports a simple reference distribution toward the posterior distribution conditioned on the observations.

Variational data assimilation. Variational data assimilation estimates the posterior mode by minimizing

$$\mathcal{J}(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{x} \mid \mathbf{y}),$$

leading to the gradient flow

$$\frac{d\mathbf{x}^{(t)}}{dt} = -\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}^{(t)}, \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}^{(t)} \mid \mathbf{y}).$$

Classical 4DVar and learned solvers such as 4DVarNet (Courtier et al., 1994; Fablet et al., 2021) therefore integrate the same posterior-score field through successive solver iterations.

Unified viewpoint. Diffusion models follow a sequence of marginal scores $\nabla_{\mathbf{x}} \log p_t$, whereas variational solvers directly follow the conditional posterior score $\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y})$. Both nevertheless rely on repeated numerical integration of a Bayesian transport field (Figure 1a). This shared structure motivates the proposed pairwise consistency framework, which learns finite-time transport operators instead of infinitesimal score updates.

Recent work on generative drifting provides a complementary perspective, evolving the model push-forward distribution directly during training (Deng et al., 2026). A subsequent flow-map interpretation showed that drifting can be recovered from a semigroup-consistent long–short transport factorization (Li and Zhu, 2026), further highlighting the connection between local generative fields and finite-time transport maps.

3 Pairwise Consistency Model (Pairwise CM)

We adopt the two-time consistency formulation introduced by Consistency Trajectory Models (CTM) (Kim et al., 2024), which generalizes endpoint consistency (Song et al., 2023; Song and Dhariwal, 2024; Heek et al., 2024), to finite-time transport between arbitrary points along a diffusion trajectory. For two states $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t')}$ on the same trajectory, we parameterize

$$g_\phi(\mathbf{x}^{(t)}, t, t') = \alpha(t, t')\mathbf{x}^{(t)} + \beta(t, t')G_\phi(\mathbf{x}^{(t)}, t, t'), \tag{6}$$

where α, β enforce boundary conditions and G_ϕ is a UNet-like network.

Standard consistency models are recovered for $t' = 0$, while arbitrary t' corresponds to the trajectory-level generalization introduced by CTM (Kim et al., 2024).

This pairwise formulation serves as the common transport primitive for the framework developed below. Our contribution is to extend it beyond prescribed diffusion trajectories to learned variational dynamics and to physical state-space evolution.

Besides naturally extending consistency learning to variational inference and physical data assimilation, this formulation also provides additional flexibility during inference, allowing adaptive solver schedules and multi-step integration while preserving the consistency properties of the learned transport.

4 Pairwise Self-Consistency in Variational Dynamics (VarCM)

Pairwise CM relies on supervised state pairs generated from a prescribed forward diffusion process. Given a clean sample $\mathbf{x}_0$, states at arbitrary solver times are obtained as

$$\mathbf{x}^{(t)} = \alpha(t)\mathbf{x}_0 + \sigma(t)\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

providing reference pairs $(\mathbf{x}^{(t)}, \mathbf{x}^{(t')})$ along the same diffusion trajectory.

Variational assimilation provides no analogous forward process. Instead, transport is induced by the posterior geometry encoded by the variational cost $\mathcal{J}_\Phi$, potentially jointly learned (Fablet et al., 2021), through the gradient flow

$$\frac{d\mathbf{x}^{(t)}}{dt} = -\nabla_{\mathbf{x}} \mathcal{J}_\Phi(\mathbf{x}^{(t)}, \mathbf{y}). \tag{7}$$

This flow defines trajectories toward the variational analysis, but its intermediate states are not available as supervised training targets. rucially, unlike diffusion-based consistency training, VarCM does not rely on a prescribed forward corruption process or on reference trajectories: both the variational geometry and its finite-time transport are learned jointly from the assimilation objective.

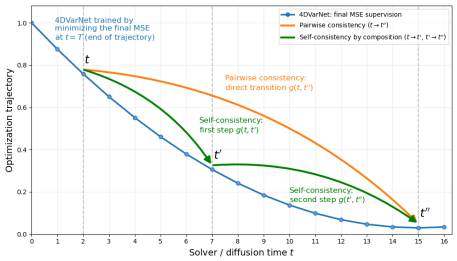


Figure 2: From endpoint supervision in end-to-end variational DA (blue line) to pairwise consistency in CM and self-consistency in VarCM.

This is the key difficulty addressed by VarCM. As illustrated in Fig. 2, learned solvers like 4DVarNet-derived strategies supervises the final reconstruction, whereas pairwise consistency additionally constrains transport between intermediate solver states. Since no reference trajectories are available, these transports cannot be supervised directly.
We therefore replace trajectory supervision by a teacher-free self-consistency objective, inspired by Song et al. (2023): different compositions of the learned pairwise operator are required to produce consistent states. VarCM thereby learns finite-time transport induced by $\mathcal{J}_\Phi$ jointly with the variational geometry itself, without requiring an explicit forward diffusion process.

Recent analyses have also connected generative drifting to score matching and Wasserstein gradient flows (Turan and Ovsjanikov, 2026), providing a complementary distribution-space variational interpretation. VarCM instead learns finite-time transports induced by a conditional variational geometry in state space.

Consistency of the variational flow. Let $\Phi_{t,t'}$ denote the finite-time flow induced by $-\nabla_{\mathbf{x}} \mathcal{J}_\phi$. If $\nabla_{\mathbf{x}} \mathcal{J}_\phi$ is Lipschitz continuous in $\mathbf{x}$, the corresponding flow is uniquely defined and satisfies

$$\Phi_{t,t''} = \Phi_{t',t''} \circ \Phi_{t,t'}, \qquad \Phi_{t,t} = \mathrm{Id}.$$

Hence, provided sufficient model capacity, $g_\theta = \Phi$ is a zero-loss solution of the pairwise composition and identity constraints used by VarCM. Composition consistency alone, however, does not uniquely identify this transport and may admit degenerate solutions. Long- and short-range reconstruction anchors are therefore used to constrain the learned semigroup toward the variational analysis. The resulting loss and training algorithm are detailed in Appendix B. Establishing convergence and identifiability of the jointly learned nonlinear variational objective and transport remains an open problem.

5 Physical and Learned Consistency Regimes

The previous sections considered Bayesian transport over a latent state representing the complete unknown variable. In many geophysical inverse problems, however, the objective is not to reconstruct a single state but an entire spatio-temporal trajectory conditioned on observations: $\mathbf{y}_{0:N}$:

$$\mathbf{x}_{0:N} = (\mathbf{x}_0, \ldots, \mathbf{x}_N),$$

Current diffusion-based approaches typically address this problem by treating the complete trajectory as a single high-dimensional tensor and learning a transport directly in this joint space (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023). Although conceptually simple, this formulation rapidly becomes computationally demanding as the temporal dimension increases, since both memory consumption and computational cost scale with the size of the complete space-time tensor.

Rather than treating diffusion time as an artificial denoising variable, we identify it with the progression of the inference procedure itself, establishing a continuous correspondence between solver time and physical time. Unlike DYffusion and Dynamical Diffusion, which couple diffusion and physical time within diffusion-based forecasting formulations (Rühling Cachay et al., 2023; Guo et al., 2025), our formulation uses the same finite-time consistency operator for observation-conditioned posterior initialization and subsequent physical propagation, without a separate assimilation and forecasting stage.

As illustrated in Figure 1, the proposed framework replaces diffusion over a full spatio-temporal tensor by a sequential transport. The solver first reconstructs a dynamically consistent initial condition (spin-up), then propagates this state along the physical dynamics using the same pairwise transport operator. This yields a unified formulation combining denoising, variational optimization and physical forecasting within a single continuous transport framework.

5.1 Pairwise consistency for space-time dynamics (DynCM)

We parameterize inference by a single continuous solver time $t \in [0, 1]$. The state at solver time t is denoted $\mathbf{x}_k^{(t)}$, where $k \in \{0, \dots, N\}$ indexes the corresponding physical state.

Unlike standard diffusion models, the solver timeline combines two successive regimes. Starting from a noisy initialization,

$$\mathbf{x}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

the trajectory first reconstructs a dynamically consistent initial condition (spin-up) for $0 \le t < t_{\mathrm{IC}}$, and subsequently follows the physical evolution for $t_{\mathrm{IC}} \le t \le 1$, where the physical regime is discretized as

$$t_{\mathrm{IC}} = t_0 < t_1 < \cdots < t_N = 1, \qquad \mathbf{x}^{(t_{\mathrm{IC}})} \approx \mathbf{x}_0, \qquad \mathbf{x}^{(t)} \approx \mathbf{x}_{k(t)}, \quad t \ge t_{\mathrm{IC}}.$$

and where the solver-to-physical-time correspondence is defined by

$$k(t) = \begin{cases} 0, & 0 \le t < t_{\mathrm{IC}}, \\ i, & t_i \le t < t_{i+1}, \quad i = 0, \dots, N-1, \\ N, & t = 1, \end{cases} \qquad \mathbf{x}^{(t)} \approx \mathbf{x}_{k(t)}.$$

A single pairwise transport operator is learned over the whole solver timeline,

$$g_\theta\left(\mathbf{x}_k^{(t)}, t, t'\right) = \alpha(t, t')\, \mathbf{x}_k^{(t)} + \beta(t, t')\, G_\theta\left(\mathbf{x}_k^{(t)}, t, t'\right), \tag{8}$$

and is conditioned on the complete observation sequence,

$$g_\theta\left(\mathbf{x}_k^{(t)}, t, t', \mathbf{y}_{0:N}\right).$$

The same transport operator therefore performs spin-up, physical propagation, and transitions between both regimes. Since all observations are available at every solver time, inference naturally behaves as a smoother, with each transport conditioned on both past and future observations.

5.2 Pairwise Self-Consistency in Variational Space-Time Dynamics (DynVarCM)

Unlike DynCM, no forward diffusion process or reference interpolant is available. Instead, both the posterior geometry and the associated transport dynamics are learned jointly from observations.

DynVarCM extends DynCM by learning pairwise transports induced by a variational gradient flow. Along the unified diffusion–physical timeline, the variational objective naturally transitions from initial-condition reconstruction during spin-up to physical trajectory propagation after t_{IC}.

The dynamics are therefore defined by

$$\frac{d\mathbf{x}_{k(t)}^{(t)}}{dt} = -\nabla_{\mathbf{x}} \mathcal{J}_\phi \left(\mathbf{x}_{k(t)}^{(t)}, t; \mathbf{y}_{0:N} \right), \tag{9}$$

where the mapping $k(t)$ between unified diffusion time and physical time is identical to that introduced in DynCM. The posterior score

$$\nabla_{\mathbf{x}} \log p \left(\mathbf{x}_{k(t)}^{(t)} \mid \mathbf{y}_{0:N} \right) = -\nabla_{\mathbf{x}} \mathcal{J}_\phi \left(\mathbf{x}_{k(t)}^{(t)}, t; \mathbf{y}_{0:N} \right) \tag{10}$$

is therefore learned implicitly through the variational cost function.

Following VarCM, the transport operator is not conditioned directly on the observations. Instead, they enter through the local variational gradient, which characterizes the local posterior geometry along the unified space–time trajectory.

The pairwise transport operator is then written as

$$g_\theta \left(\mathbf{x}_{k(t)}^{(t)}, t, t', \mathbf{d}_{k(t)}^{(t)} \right), \qquad t < t',$$

where $\mathbf{d}$ denotes the gradient of the variational cost. This operator approximates the finite-time integration of the learned gradient flow between two arbitrary points of the unified timeline.

Unlike DynCM, no reference transitions are available for supervision. Instead, the transport operator is learned entirely through pairwise self-consistency by enforcing the semigroup property

$$g_\theta \left(\mathbf{x}, t, t'', \mathbf{d}_{k(t)}^{(t)} \right) \approx g_\theta \left(g_\theta \left(\mathbf{x}, t, t', \mathbf{d}_{k(t)}^{(t)} \right), t', t'', \mathbf{d}_{k(t')}^{(t')} \right), \qquad t < t' < t''. \tag{11}$$

DynVarCM retains the unified diffusion–physical timeline of DynCM while replacing the explicit diffusion dynamics by a learned variational cost. The gradient of $\mathcal{J}_\phi$ defines the local posterior geometry, and the pairwise operator G_θ learns its finite-time transport. DynCM and DynVarCM therefore share the same space–time transport framework, differing only in how the local transport geometry is constructed. The linear–Gaussian setting provides an exactly solvable instance of this dynamic formulation. As shown in Appendix F, the corresponding finite-time conditional transports recover the classical Gaussian smoothing operators, providing an analytical consistency check for DynVarCM.

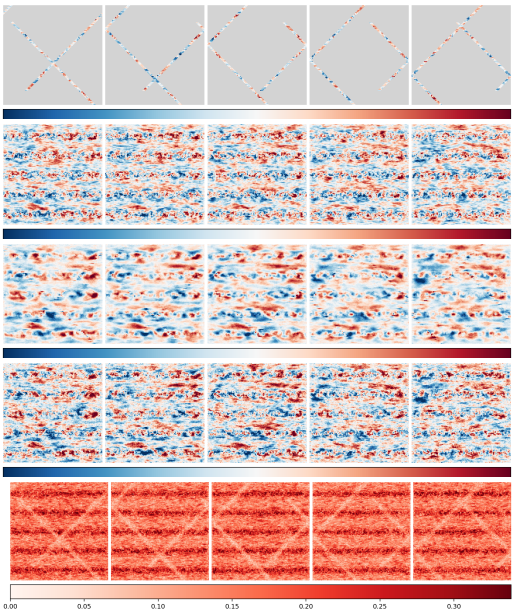
6 Results

We evaluate the proposed pairwise consistency framework on three representative Bayesian inference problems, illustrating the successive developments introduced throughout the paper:

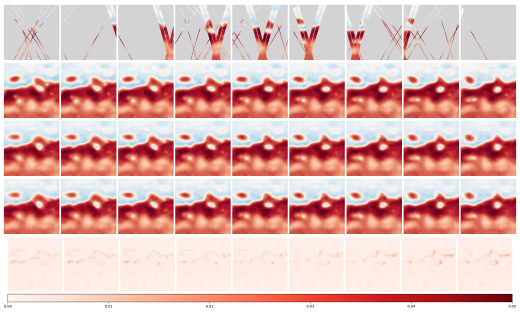
- **Idealized Gaussian Process (GP):** a linear Gaussian inverse problem with known posterior, allowing quantitative evaluation of deterministic reconstruction and posterior sampling.
- **Sea Surface Height (SSH) reconstruction:** a realistic ocean data assimilation experiment from satellite altimetry observations, assessing large-scale variational reconstruction.
- **Sea-Ice Concentration (SIC) super-resolution:** a high-resolution reconstruction problem in the Arctic marginal ice zone, demonstrating the ability of the framework to recover fine-scale spatial structures.

We evaluate all methods using RMSE, effective spatial resolutions (λ_x), and CRPS, computed from ensembles obtained by independently sampling the initial latent state and propagating each member through the deterministic pairwise transport, when applicable. Across the three benchmarks, the proposed consistency-based models achieve reconstruction performance comparable to Flow Matching approaches, while retaining the few-step inference and flexible transport structure of the proposed framework. To quantify inference efficiency, we evaluate RMSE and CRPS as a function of the number of neural-network function evaluations (NFE) on the SSH experiment, comparing Pairwise

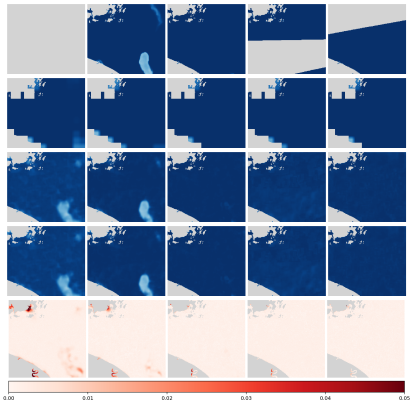
CM with Flow Matching (FM), the full accuracy–cost and calibration–cost trade-offs are reported in Appendix C, see Fig 7.



(a) GP



(b) SSH



(c) SIC

Figure 3: Comparison of the observations, ground truth, ensemble mean, a representative ensemble member, and ensemble spread for the GP, SSH_GF, and SIC experiments.

	UNet			4DVarNet			FM			DynFM		
Dataset	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓
GP	0.249	50.4	–	0.228	61.6	–	**0.198**	**50.2**	**0.016**	0.229	54.9	0.030
SSH	0.186	153.4	–	0.075	1.35	–	0.062	1.28	0.036	0.070	1.42	0.043
SIC	0.173	21.8	–	0.151	18.5	–	0.126	13.2	0.061	0.134	15.6	0.067
	CM			**VarCM**			**DynCM**			**DynVarCM**		
Dataset	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓	RMSE ↓	λ_x ↓	CRPS ↓
GP	0.230	50.6	0.109	0.230	52.1	0.029	0.226	52.0	0.048	0.229	51.2	0.018
SSH	**0.055**	**1.27**	**0.033**	0.064	2.10	0.039	0.062	2.35	0.036	0.069	1.34	0.050
SIC	0.141	17.1	0.073	0.157	19.4	0.081	0.119	12.6	0.058	**0.108**	**10.9**	**0.051**

Table 1: Comparison of baseline, Flow Matching, and pairwise consistency models on GP, SSH, and SIC. We report RMSE, spatial resolution λ_x, and CRPS; CRPS is reported only for ensemble-generating methods.

7 Conclusion

We introduced a unified pairwise consistency framework for Bayesian inference that learns finite-time transport operators rather than infinitesimal fields, connecting diffusion models, variational data assimilation, and physics-aware sequential inference. We derived four complementary formulations: supervised Pairwise CM, self-consistent VarCM, hybrid diffusion–physical DynCM, and fully variational space–time DynVarCM. Experiments on synthetic and geophysical benchmarks demonstrated accurate few-step reconstruction, sequential forecasting, and uncertainty-aware posterior sampling with substantially fewer solver evaluations. The framework opens new directions for scalable Bayesian transport in high-dimensional dynamical systems, including Earth-system foundation models and probabilistic physical emulation.

ACKNOWLEDMENTS

This work was partially supported by the French Agence Nationale de la Recherche (ANR) through the OceaniX project (ANR-19-CHIA-0016). The authors acknowledge the high-performance computing and GPU resources provided by the CPER AIDA GPU cluster, supported by the Regional Council of Brittany, Brest Métropole and the European Regional Development Fund (FEDER). This work was granted access to the HPC resources of TGCC operated by GENCI (A0190116867) and IDRIS (AD010117657).

REPRODUCIBILITY STATEMENT

We provide an anonymized source-code archive as supplementary material, including implementations of all proposed methods and experimental baselines. The appendix documents model architectures, training and sampling procedures, hyperparameters, data preprocessing, and evaluation protocols. Together with the provided code and configuration files, these materials enable reproduction of the experiments reported in this work.

AI USE STATEMENT

Generative AI tools were used to assist with scientific writing and editing, translation, and the refinement of presentation. AI-assisted suggestions were critically reviewed, verified, and revised by the authors. The scientific methodology, experiments, results, and conclusions were developed and validated by the authors, who take full responsibility for the content of this work.

REFERENCES

M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. In *International Conference on Learning Representations*, 2023.

B. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. doi: 10.1016/0304-4149(82)90051-5.

M. Asch, M. Bocquet, and M. Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016. doi: 10.1137/1.9781611974546.

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. Flow map matching, 2024.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120 (519):1367–1387, 1994. doi: 10.1002/qj.49712051912.

M. Dashti and A. M. Stuart. The bayesian approach to inverse problems. In R. Ghanem, D. Higdon, and H. Owhadi, editors, *Handbook of Uncertainty Quantification*, pages 311–428. Springer, Cham, 2017. doi: 10.1007/978-3-319-12385-1_7.

M. Deng, H. Li, T. Li, Y. Du, and K. He. Generative modeling via drifting, 2026. URL https://arxiv.org/abs/2602.04770.

EUMETSAT Ocean and Sea Ice SAF. Osi saf global sea ice concentration (osi-401-d). 2023. doi: 10.48670/moi-00134. Operational sea ice concentration product.

R. Fablet, B. Chapron, L. Drumetz, E. Mémin, O. Pannekoucke, and F. Rousseau. Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, 13 (10):e2021MS002572, 2021. doi: 10.1029/2021MS002572.

X. Guo, Y. Zhang, B. Chen, H. Xu, J. Wang, and M. Long. Dynamical diffusion: Learning temporal dynamics with diffusion models. In *International Conference on Learning Representations*, 2025.

J. Heek, E. Hoogeboom, and T. Salimans. Multistep consistency models. In *International Conference on Learning Representations*, 2024.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

T. Lavergne, A. M. Sørensen, S. Kern, R. Tonboe, D. Notz, S. Aaboe, L. Bell, G. Dybkjær, S. Eastwood, C. Gabarro, G. Heygster, M. A. Killie, M. Brandt Kreiner, J. Lavelle, R. Saldo, S. Sandven, and L. T. Pedersen. Version 2 of the eumetsat osi saf and esa cci sea-ice concentration climate data records. *The Cryosphere*, 13:49–78, 2019. doi: 10.5194/tc-13-49-2019.

Z. Li and B. Zhu. A long-short flow-map perspective for drifting models, 2026. URL `https://arxiv.org/abs/2602.20463`.

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

F. Rabier, H. Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons. The ECMWF operational implementation of four-dimensional variational assimilation. i: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143–1170, 2000. doi: 10.1002/qj.49712656415.

S. Rühling Cachay, B. Zhao, H. Joren, and R. Yu. DYffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, 2023.

Y. Song and P. Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252, 2023.

A. M. Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005. doi: 10.1137/1.9780898717921.

E. Turan and M. Ovsjanikov. Generative drifting is secretly score matching: A spectral and variational perspective, 2026. URL `https://arxiv.org/abs/2603.09936`.

T. Wulf, J. Buus-Hinkler, S. Singha, H. Shi, and M. B. Kreiner. Pan-arctic sea ice concentration from sar and passive microwave. *The Cryosphere*, 18:5277–5300, 2024. doi: 10.5194/tc-18-5277-2024.

A BACKGROUND

A.1 DIFFUSION AND REVERSE-TIME SDE.

Score-based diffusion models construct a stochastic process that gradually transforms data samples into Gaussian noise through a forward stochastic differential equation (SDE), while generation is performed by integrating the associated reverse-time dynamics (Ho et al., 2020; Song et al., 2021). Starting from a data sample $\mathbf{x}_0$, the forward process is defined by the Itô SDE

$$d\mathbf{x}^{(t)} = f(\mathbf{x}^{(t)}, t)\, dt + g(t)\, d\mathbf{w}^{(t)}, \qquad t \in [0, 1], \tag{12}$$

where f and g denote the drift and diffusion coefficients, respectively, and $\mathbf{w}^{(t)}$ is a standard Wiener process.

For the Variance-Exploding SDE (VE-SDE) introduced in Song et al. (2021), the reverse-time dynamics satisfy

$$\begin{cases} d\mathbf{x}^{(t)} = \Big(f(\mathbf{x}^{(t)}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)}) \Big)\, dt + g(t)\, d\mathbf{w}^{(t)}, \\ f(\mathbf{x}^{(t)}, t) = 0, \\ g(t) = \sigma'(t) \sqrt{2 \log \sigma(t)}, \end{cases}$$

where $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)})$ denotes the score function of the perturbed distribution.

In many applications, including Bayesian inverse problems and geophysical data assimilation, the objective is not to sample from the unconditional distribution $p_t(\mathbf{x})$, but from the posterior distribution conditioned on partial or noisy observations $\mathbf{y}$. This naturally leads to the conditional score

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)} \mid \mathbf{y}),$$

which modifies the reverse-time dynamics into a data-consistent stochastic transport. By Bayes' rule,

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)} \mid \mathbf{y}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)}) + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x}^{(t)}),$$

where the second term corresponds to the likelihood gradient and acts as a data-fidelity correction that guides the reverse dynamics toward the posterior distribution.

This formulation naturally connects score-based diffusion models with Bayesian inverse problems, inpainting, and data assimilation. For example, observations may correspond to partial measurements of the latent state,

$$\mathbf{y} = \mathbf{H}\mathbf{x}_0 + \boldsymbol{\varepsilon},$$

where $\mathbf{H}$ denotes an observation operator. The conditional score then simultaneously enforces consistency with the available observations while reconstructing the unobserved components through the learned prior represented by the unconditional score.

In practice, conditioning can be incorporated either explicitly through the likelihood gradient $\nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x}^{(t)})$, or implicitly by learning a score network conditioned on the observations. In both cases, the reverse-time dynamics become guided toward regions of high posterior probability, thereby jointly performing denoising, reconstruction, and posterior sampling.

A.2 VARIATIONAL DATA ASSIMILATION AND LEARNED OPTIMIZERS.

Variational data assimilation seeks to reconstruct the latent state $\mathbf{x} \in \mathbb{R}^n$ from partial and noisy observations $\mathbf{y} \in \mathbb{R}^m$ by minimizing a Bayesian objective that balances observational consistency

and prior information (Courtier et al., 1994; Rabier et al., 2000; Asch et al., 2016). The analysis state is defined as

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \mathcal{J}(\mathbf{x}, \mathbf{y}), \tag{13}$$

where the variational cost typically takes the form

$$\mathcal{J}(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2_{R^{-1}} + \|\mathbf{x} - \mathbf{x}_b\|^2_{B^{-1}},$$

with $\mathbf{x}_b$ the background state and B and R the prior and observation error covariance matrices, respectively.

Under the Bayesian interpretation,

$$\mathcal{J}(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{x} \mid \mathbf{y}),$$

the optimization naturally defines a continuous gradient flow,

$$\frac{d\mathbf{x}^{(t)}}{dt} = -\nabla_{\mathbf{x}} \mathcal{J}\left(\mathbf{x}^{(t)}, \mathbf{y}\right), \tag{14}$$

whose stationary point corresponds to the maximum a posteriori (MAP) estimate. A standard time discretization of this gradient flow yields the classical 4DVar iterations

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \alpha_i \nabla_{\mathbf{x}} \mathcal{J}\left(\mathbf{x}^{(i)}, \mathbf{y}\right),$$

where i denotes the solver iteration and α_i the corresponding step size.

Building upon this formulation, Fablet et al. (2021) proposed to replace the hand-crafted optimization procedure by a learnable architecture, referred to as *4DVarNet*. Rather than modifying the variational objective itself, 4DVarNet learns the optimization dynamics in an end-to-end manner, jointly estimating the prior representation and the iterative update rule.

More precisely, the classical gradient-descent iteration is augmented with a learned correction,

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \alpha_i \nabla_{\mathbf{x}} \mathcal{J}\left(\mathbf{x}^{(i)}, \mathbf{y}\right) + u_\phi\left(\mathbf{x}^{(i)}, \mathbf{y}\right),$$

where u_ϕ is a neural network, typically implemented as a convolutional recurrent architecture. This learned correction acts as an adaptive, data-driven preconditioner of the gradient flow, accelerating convergence while compensating for model errors and approximations in the variational formulation.

From the perspective developed in this paper, both classical 4DVar and 4DVarNet can therefore be interpreted as numerical integrators of the continuous Bayesian transport defined by Equation equation 14. This observation provides the conceptual bridge with the proposed consistency formulations, which approximate the same transport through learned finite-time operators rather than through successive gradient-descent iterations.

A.3 Unrolled variational solvers as score-driven dynamics

We consider the generic unrolled architecture

$$\begin{cases} \mathbf{x}^{(0)} \sim \mathcal{N}\left(\mathbf{0}, \sigma_0^2 \mathbf{I}\right), \\ \mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \mathcal{R}_\phi\left(\mathbf{x}^{(i)}, \mathbf{y}, i\right), \qquad i = 0, \ldots, K-1, \end{cases} \tag{15}$$

where i denotes the solver iteration and K the total number of unrolled updates. Such an iterative architecture can be interpreted as the time discretization of a dynamical system evolving in state space.

In a variational data assimilation framework, the update is driven by the gradient of a Bayesian cost functional.

When

$$\mathcal{J}(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{x} \mid \mathbf{y}) + C(\mathbf{y}), \tag{16}$$

where $C(\mathbf{y})$ is independent of $\mathbf{x}$, its gradient satisfies

$$-\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}). \tag{17}$$

A standard gradient-descent update can therefore be written as

$$\mathcal{R}\left(\mathbf{x}^{(i)}, \mathbf{y}, i\right) = \alpha_i \nabla_{\mathbf{x}} \log p\left(\mathbf{x}^{(i)} \mid \mathbf{y}\right), \tag{18}$$

where $\alpha_i > 0$ denotes the step size. The residual update thus follows a discretized posterior-score field and progressively transports the state toward a stationary point of the posterior density. Under suitable regularity and local convexity assumptions, this stationary point corresponds to the maximum a posteriori estimator (Stuart, 2010; Dashti and Stuart, 2017).

Learned variational solvers generalize this update by parameterizing the residual operator itself,

$$\mathcal{R}_\phi\left(\mathbf{x}^{(i)}, \mathbf{y}, i\right) = \Psi_\phi\left(\mathbf{x}^{(i)}, \nabla_{\mathbf{x}} \mathcal{J}\left(\mathbf{x}^{(i)}, \mathbf{y}\right), \mathbf{y}, i\right), \tag{19}$$

where Ψ_ϕ learns how to transform the current variational gradient into an effective state update. This formulation includes 4DVarNet, which learns both the prior structure entering the variational cost and a recurrent solver optimized end-to-end from data (Fablet et al., 2021). The resulting architecture may therefore be viewed as a learned integrator of the posterior-score dynamics rather than as a fixed gradient-descent method.

The key distinction lies in the probability distribution whose score drives the dynamics. Score-based diffusion models follow the marginal score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}^{(t)})$ along a noise-annealed path toward the data distribution (Song et al., 2021), whereas 4DVar and 4DVarNet follow the conditional posterior score $\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y})$ defined by an explicit Bayesian inverse problem.

From this viewpoint, unrolled variational solvers should be interpreted as deterministic posterior-score optimizers rather than posterior samplers: without an additional stochastic component, their trajectories converge toward a posterior mode instead of representing the full posterior distribution. Nevertheless, diffusion models and learned variational solvers share a common computational structure, as both integrate score-driven dynamics through successive state-space updates. This common structure motivates the finite-time transport formulations introduced in the following sections.

B Additional details for training and sampling strategies of Pairwise Consistency Models

B.1 Pairwise Consistency Models (CM)

B.1.1 Training the Pairwise Consistency Model

Training relies on the pairwise consistency property introduced in Section 3. Let $t > t' > t''$ denote three solver times sampled from the diffusion schedule. Since all three states belong to the same reverse trajectory, transporting either directly from t to t'', or through the intermediate time t', should produce identical predictions.

The corresponding objective is

$$\mathcal{L}_{\mathrm{CM}} = \mathbb{E}_{\mathbf{x}_0,\mathbf{z},t,t',t''}\left[\left\| g_\phi\left(\mathbf{x}^{(t)}, t, t''\right) - g_\phi\left(\mathbf{x}^{(t')}, t', t''\right)\right\|^2\right], \tag{20}$$

where

$$\mathbf{x}^{(t)} = \alpha(t)\mathbf{x}_0 + \sigma(t)\mathbf{z},$$

and the same latent noise realization $\mathbf{z}$ is used to generate all diffusion states along the trajectory.

This objective enforces the semigroup structure of the learned transport: irrespective of the starting diffusion time, all states lying on the same reverse trajectory are mapped consistently toward the same target state at solver time t''.

Algorithm 1 Training of the Pairwise Consistency Model

1: **Input:** data distribution $p_{\mathrm{data}}(\mathbf{x}_0)$, diffusion schedule $\mathcal{T}$, model G_ϕ
2: Initialize parameters ϕ
3: **while** not converged **do**
4: Sample $\mathbf{x}_0 \sim p_{\mathrm{data}}$
5: Sample latent noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: Sample solver times $t > t' > t''$
7: Generate diffusion states

$$\mathbf{x}^{(t)} = \alpha(t)\mathbf{x}_0 + \sigma(t)\mathbf{z},$$

$$\mathbf{x}^{(t')} = \alpha(t')\mathbf{x}_0 + \sigma(t')\mathbf{z},$$

8: Predict pairwise transports

$$\widehat{\mathbf{x}}^{(t'')} = g_\phi\left(\mathbf{x}^{(t)}, t, t''\right),$$

$$\widehat{\mathbf{x}}'^{(t'')} = g_\phi\left(\mathbf{x}^{(t')}, t', t''\right),$$

9: Compute

$$\mathcal{L}_{\mathrm{CM}} = \left\|\widehat{\mathbf{x}}^{(t'')} - \widehat{\mathbf{x}}'^{(t'')}\right\|^2$$

10: Update $\phi \leftarrow \phi - \eta\nabla_\phi\mathcal{L}_{\mathrm{CM}}$
11: **end while**

B.1.2 Sampling

Once trained, the Pairwise Consistency Model defines a finite-time transport operator that can be iteratively applied to generate samples. In the continuous limit, this transport approximates the probability-flow ODE associated with the reverse diffusion process,

$$\frac{d\mathbf{x}^{(t)}}{dt} = -D_\phi\left(\mathbf{x}^{(t)}, t\right), \tag{21}$$

where D_ϕ denotes the learned transport field. Rather than explicitly integrating this ODE through many small solver steps, Pairwise CM directly approximates finite-time transports between arbitrary solver times. Sampling therefore consists of repeatedly applying the learned pairwise operator along a prescribed solver schedule,

Algorithm 2 Sampling with the Pairwise Consistency Model

1: **Input:** trained model G_ϕ, solver-time schedule $\{t_0 = 0, \ldots, t_K = 1\}$
2: Initialize

$$\mathbf{x}^{(t_0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

3: **for** $k = 0, \ldots, K-1$ **do**
4: Apply the learned pairwise transport

$$\mathbf{x}^{(t_{k+1})} = \alpha(t_k, t_{k+1})\mathbf{x}^{(t_k)} + \beta(t_k, t_{k+1})G_\phi\left(\mathbf{x}^{(t_k)}, t_k, t_{k+1}\right)$$

5: **end for**
6: **return** $\mathbf{x}^{(t_K)}$

B.2 VarCM

B.2.1 Training

As for Pairwise CM, the transport operator is parameterized as

$$g_\phi\left(\mathbf{x}^{(t)}, t, t'\right) = \alpha(t, t')\,\mathbf{x}^{(t)} + \beta(t, t')\,G_\phi\left(\mathbf{x}^{(t)}, t, t'\right). \tag{22}$$

Unlike diffusion models, no supervised transport pairs are available. We therefore learn the transport operator solely through self-consistency and reconstruction constraints.

Given an initial noisy state

$$\mathbf{x}^{(1)} = \sigma_{\max}\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

and three solver times

$$0 < t'' < t' < t < 1,$$

we define the composed transport

$$(g \circ g)\,(1, t, t'') = g_\phi\left(g_\phi\left(\mathbf{x}^{(1)}, 1, t\right), t, t''\right).$$

The training objective is

$$\mathcal{L}_{\text{VarCM}} = \lambda_{\text{pair}}\,\mathcal{L}_{\text{pair}} + \lambda_{\text{long}}\,\mathcal{L}_{\text{long}} + \lambda_{\text{short}}\,\mathcal{L}_{\text{short}}, \tag{23}$$

$$\mathcal{L}_{\text{pair}} = \mathbb{E}\left[\left\|(g \circ g)(1, t, t'') - (g \circ g)(1, t', t'')\right\|^2\right], \tag{24}$$

$$\mathcal{L}_{\text{long}} = \mathbb{E}\left[\left\|(g \circ g)(1, t, 0) - \mathbf{x}_0\right\|^2\right], \tag{25}$$

$$\mathcal{L}_{\text{short}} = \mathbb{E}\left[\left\|(g \circ g)(t, t', 0) - \mathbf{x}_0\right\|^2\right]. \tag{26}$$

The objective combines three complementary constraints. The pairwise term enforces the semigroup property of the learned transport operator by requiring direct and composed transports to agree. In the absence of supervised diffusion trajectories, the two reconstruction terms act as anchoring constraints that connect the learned transport to the data distribution. The long reconstruction supervises trajectories originating from the noisy initialization, whereas the short reconstruction constrains intermediate states to remain on the same posterior transport manifold.

Algorithm 3 Training of VarCM

1: **Input:** $p_{\mathrm{data}}(\mathbf{x}_0)$, variational objective J_Φ, transport model G_ϕ
2: Initialize parameters ϕ
3: **while** not converged **do**
4: Sample $\mathbf{x}_0 \sim p_{\mathrm{data}}$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $0 < t'' < t' < t < 1$
5: Set

$$\mathbf{x}^{(1)} = \sigma_{\max}\mathbf{z}$$

6: Compute intermediate states

$$\widehat{\mathbf{x}}^{(t)} = g_\phi(\mathbf{x}^{(1)}, 1, t), \qquad \widehat{\mathbf{x}}^{(t')} = g_\phi(\mathbf{x}^{(1)}, 1, t')$$

7: Compute pairwise predictions

$$\widehat{\mathbf{x}}^{(t'')}_{\mathrm{dir}} = g_\phi(\mathbf{x}^{(1)}, 1, t''), \qquad \widehat{\mathbf{x}}^{(t'')}_{\mathrm{comp}} = g_\phi(\widehat{\mathbf{x}}^{(t)}, t, t'')$$

8: Compute endpoint predictions

$$\widehat{\mathbf{x}}^{(0)}_{\mathrm{long}} = g_\phi(\widehat{\mathbf{x}}^{(t)}, t, 0), \qquad \widehat{\mathbf{x}}^{(0)}_{\mathrm{short}} = g_\phi(\widehat{\mathbf{x}}^{(t')}, t', 0)$$

9: Compute

$$\mathcal{L}_{\mathrm{VarCM}} = \lambda_{\mathrm{pair}} \left\| \widehat{\mathbf{x}}^{(t'')}_{\mathrm{dir}} - \widehat{\mathbf{x}}^{(t'')}_{\mathrm{comp}} \right\|^2 + \lambda_{\mathrm{long}} \left\| \widehat{\mathbf{x}}^{(0)}_{\mathrm{long}} - \mathbf{x}_0 \right\|^2 + \lambda_{\mathrm{short}} \left\| \widehat{\mathbf{x}}^{(0)}_{\mathrm{short}} - \mathbf{x}_0 \right\|^2$$

10: Update

$$\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{\mathrm{VarCM}}$$

11: **end while**

Together, these three terms learn a finite-time transport operator solely from self-consistency and endpoint supervision, without requiring any prescribed forward diffusion process or teacher model.

From the transport perspective developed throughout this paper, VarCM learns finite-time approximations of the posterior gradient flow associated with the variational objective. Following the 4DVarNet paradigm (Fablet et al., 2021), the variational objective itself—and in particular its learned prior operator—can be optimized jointly with the transport operator. Consequently, the learned transport becomes implicitly conditioned on the geometry of the posterior distribution, providing a unified framework that combines consistency learning, learned variational optimization and Bayesian transport.

B.2.2 Sampling

Once trained, VarCM acts as a learned variational solver. Starting from a noisy initialization, the state is progressively transported toward the posterior mode by repeatedly applying the learned pairwise transport operator conditioned on the current variational gradient.

At each solver step, the transport is given by

$$\mathbf{x}^{(t_{k+1})} = \alpha(t_k, t_{k+1})\mathbf{x}^{(t_k)} + \beta(t_k, t_{k+1}) G_\phi \left(\mathbf{x}^{(t_k)}, \nabla_{\mathbf{x}} J_\Phi \left(\mathbf{x}^{(t_k)}, \mathbf{y} \right), t_k, t_{k+1} \right), \tag{27}$$

where J_Φ denotes the learned variational objective and $\nabla_{\mathbf{x}} J_\Phi$ provides the local posterior descent direction. Unlike classical gradient descent, the learned transport operator performs finite-time updates between arbitrary solver times, allowing significantly fewer solver evaluations.

Algorithm 4 Sampling with the Variational Pairwise Consistency Model (VarCM)

1: **Input:** trained transport model G_ϕ, learned variational objective J_Φ, observations $\mathbf{y}$, solver-time schedule $\{t_0 = 0, \ldots, t_K = 1\}$
2: Initialize

$$\mathbf{x}^{(t_0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

3: **for** $k = 0, \ldots, K-1$ **do**
4: Compute the variational gradient

$$\mathbf{d}_k = \nabla_{\mathbf{x}} J_\Phi \left(\mathbf{x}^{(t_k)}, \mathbf{y} \right)$$

5: Apply the learned pairwise transport

$$\mathbf{x}^{(t_{k+1})} = \alpha(t_k, t_{k+1}) \mathbf{x}^{(t_k)} + \beta(t_k, t_{k+1}) G_\phi \left(\mathbf{x}^{(t_k)}, \mathbf{d}_k, t_k, t_{k+1} \right)$$

6: **end for**
7: **return** $\mathbf{x}^{(t_K)}$

B.3 DynCM

B.3.1 Training

Algorithm 5 Training of the Dynamic Pairwise Consistency Model (DynCM)

1: **Input:** trajectory distribution $p_{\text{data}}(\mathbf{x}_{0:N})$, observation sequence $\mathbf{y}_{0:N}$, transition time t_{IC}, pairwise model G_θ
2: Initialize the model parameters θ
3: **while** not converged **do**
4: Sample a physical trajectory

$$\mathbf{x}_{0:N} = (\mathbf{x}_0, \ldots, \mathbf{x}_N) \sim p_{\text{data}}$$

5: Sample two unified times

$$0 \le t < t' \le 1$$

6: Construct the corresponding reference states

$$\bar{\mathbf{x}}^{(t)}_{k(t)} \quad \text{and} \quad \bar{\mathbf{x}}^{(t')}_{k(t')}$$

7: Predict the finite-time transition

$$\widehat{\mathbf{x}}^{(t')}_{k(t')} = \alpha(t, t')\, \bar{\mathbf{x}}^{(t)}_{k(t)} + \beta(t, t')\, G_\theta \left(\bar{\mathbf{x}}^{(t)}_{k(t)}, t, t', \mathbf{y}_{0:N} \right)$$

8: Compute the pairwise transport loss

$$\mathcal{L}_{\text{DynCM}} = \left\| \widehat{\mathbf{x}}^{(t')}_{k(t')} - \bar{\mathbf{x}}^{(t')}_{k(t')} \right\|_2^2$$

9: Update the model parameters

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{DynCM}}$$

10: **end while**

The DynCM operator is trained over the full unified diffusion–physical timeline. Each training pair may therefore belong to the spin-up regime, the physical-evolution regime, or span the transition

between the two. The model learns the corresponding finite-time transport directly between any two times $0 \le t < t' \le 1$.

For notational consistency, we set $k(t) = 0$ for $t < t_{\mathrm{IC}}$. The reference state at any unified time t can then be written as $\bar{\mathbf{x}}^{(t)}_{k(t)}$, whether t lies in the spin-up or physical regime.

B.3.2 SAMPLING

Sampling follows the same unified diffusion–physical timeline used during training. Starting from a Gaussian state at $t = 0$, the model first performs the spin-up toward the initial physical condition. At $t = t_{\mathrm{IC}}$, the generated state is identified with the physical state at $k = 0$. The model then propagates this state along the physical trajectory until $k = N$.

Inference is performed over a numerical discretization of diffusion time,

$$0 = t_0 < t_1 < \cdots < t_K = 1,$$

which may include additional intermediate diffusion steps beyond the physical times, where the mapping $k(t)$ is defined in Section 5. This discretization may contain only a few direct transitions or additional intermediate steps between the physical times. The physical index associated with each time is determined by the mapping $k(t)$ introduced above, with $k(t) = 0$ throughout the spin-up regime.

Algorithm 6 Sampling with the Dynamic Pairwise Consistency Model (DynCM)

1: **Input:** trained pairwise model G_θ, observation sequence $\mathbf{y}_{0:N}$, unified-time discretization $\{t_0, \ldots, t_K\}$
2: Initialize the state at $t_0 = 0$

$$\widehat{\mathbf{x}}^{(t_0)}_{k(t_0)} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2_{\max}\mathbf{I}\right), \qquad k(t_0) = 0$$

3: **for** $i = 0, \ldots, K-1$ **do**
4: Predict the transition from t_i to t_{i+1}

$$\widehat{\mathbf{x}}^{(t_{i+1})}_{k(t_{i+1})} = \alpha(t_i, t_{i+1})\, \widehat{\mathbf{x}}^{(t_i)}_{k(t_i)} + \beta(t_i, t_{i+1})\, G_\theta\left(\widehat{\mathbf{x}}^{(t_i)}_{k(t_i)}, t_i, t_{i+1}, \mathbf{y}_{0:N}\right)$$

5: **end for**
6: Extract the generated states at the physical times and return

$$\widehat{\mathbf{x}}_{0:N} = (\widehat{\mathbf{x}}_0, \ldots, \widehat{\mathbf{x}}_N)$$

B.4 DYNVARCM

B.4.1 TRAINING

The model jointly learns the variational cost function $\mathcal{J}_\phi$, which defines the posterior geometry, and the pairwise transport operator G_θ, which approximates the associated finite-time gradient flow over the unified diffusion–physical timeline. Since no reference trajectories are available, supervision relies entirely on pairwise self-consistency together with anchoring constraints linking the learned dynamics to the observed physical trajectory.

Algorithm 7 Training of the Variational Dynamic Pairwise Consistency Model (DynVarCM)

1: **Input:** trajectory distribution $p_{\mathrm{data}}(\mathbf{x}_{0:N})$, observation sequence $\mathbf{y}_{0:N}$, variational cost $\mathcal{J}_\phi$, pairwise transport model G_θ
2: Initialize the model parameters
$$(\phi, \theta)$$
3: **while** not converged **do**
4: Sample a physical trajectory
$$\mathbf{x}_{0:N} \sim p_{\mathrm{data}}$$
5: Sample three unified times
$$0 \le t < t' < t'' \le 1$$
6: Construct the corresponding unified states
$$\mathbf{x}^{(t)}_{k(t)}, \qquad \mathbf{x}^{(t')}_{k(t')}, \qquad \mathbf{x}^{(t'')}_{k(t'')}$$
7: Update the posterior geometry by computing the local variational gradients
$$\mathbf{d}^{(t)}_{k(t)} = \nabla_{\mathbf{x}} \mathcal{J}_\phi \left(\mathbf{x}^{(t)}_{k(t)}, t; \mathbf{y}_{0:N} \right),$$
$$\mathbf{d}^{(t')}_{k(t')} = \nabla_{\mathbf{x}} \mathcal{J}_\phi \left(\mathbf{x}^{(t')}_{k(t')}, t'; \mathbf{y}_{0:N} \right).$$
8: Predict the direct finite-time transport
$$\widehat{\mathbf{x}}_{\mathrm{dir}} = G_\theta \left(\mathbf{x}^{(t)}_{k(t)}, t, t'', \mathbf{d}^{(t)}_{k(t)} \right)$$
9: Predict the composed transport
$$\widehat{\mathbf{x}}_{\mathrm{comp}} = G_\theta \left(G_\theta \left(\mathbf{x}^{(t)}_{k(t)}, t, t', \mathbf{d}^{(t)}_{k(t)} \right), t', t'', \mathbf{d}^{(t')}_{k(t')} \right)$$
10: Compute the training objective
$$\mathcal{L}_{\mathrm{DynVarCM}} = \mathcal{L}_{\mathrm{comp}} + \lambda_{\mathrm{IC}} \mathcal{L}_{\mathrm{IC}} + \lambda_{\mathrm{phys}} \mathcal{L}_{\mathrm{phys}},$$
where
$$\mathcal{L}_{\mathrm{comp}} = \| \widehat{\mathbf{x}}_{\mathrm{dir}} - \widehat{\mathbf{x}}_{\mathrm{comp}} \|_2^2,$$
and $\mathcal{L}_{\mathrm{IC}}$ and $\mathcal{L}_{\mathrm{phys}}$ denote the anchoring losses defined below.
11: Update the variational cost and transport operator
$$(\phi, \theta) \leftarrow (\phi, \theta) - \eta \nabla_{(\phi,\theta)} \mathcal{L}_{\mathrm{DynVarCM}}$$
12: **end while**

B.4.2 SAMPLING

Sampling follows the same unified diffusion–physical timeline introduced for DynCM. Starting from a Gaussian initialization, the model first reconstructs the initial physical condition during the spin-up phase before propagating it along the physical trajectory. Unlike DynCM, where the transport operator is conditioned directly on the observations, DynVarCM updates the local posterior geometry at every unified time by recomputing the variational gradient before applying the learned finite-time transport.

Inference is performed over a unified-time discretization

$$0 = t_0 < t_1 < \cdots < t_K = 1,$$

which may include additional intermediate diffusion steps beyond the physical times, while the mapping $k(t)$ defines the corresponding physical index.

Algorithm 8 Sampling with the Variational Dynamic Pairwise Consistency Model (DynVarCM)

1: **Input:** trained variational cost $\mathcal{J}_\phi$, trained pairwise transport operator G_θ, observation sequence $\mathbf{y}_{0:N}$, unified-time discretization $\{t_0, \dots, t_K\}$
2: Initialize

$$\widehat{\mathbf{x}}^{(t_0)}_{k(t_0)} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\max}\mathbf{I}), \qquad k(t_0) = 0$$

3: **for** $i = 0, \dots, K-1$ **do**
4: Update the local posterior geometry

$$\mathbf{d}^{(t_i)}_{k(t_i)} = \nabla_{\mathbf{x}} \mathcal{J}_\phi \left(\widehat{\mathbf{x}}^{(t_i)}_{k(t_i)}, t_i; \mathbf{y}_{0:N} \right)$$

5: Predict the finite-time transport

$$\widehat{\mathbf{x}}^{(t_{i+1})}_{k(t_{i+1})} = G_\theta \left(\widehat{\mathbf{x}}^{(t_i)}_{k(t_i)}, t_i, t_{i+1}, \mathbf{d}^{(t_i)}_{k(t_i)} \right)$$

6: **end for**
7: Extract the states corresponding to the physical timeline and return

$$\widehat{\mathbf{x}}_{0:N} = (\widehat{\mathbf{x}}_0, \dots, \widehat{\mathbf{x}}_N)\,.$$

C ADDITIONAL DESCRIPTION AND RESULTS FOR THE EXPERIMENTS

C.1 SPDE-BASED DATA ASSIMILATION: EXPERIMENTAL FRAMEWORK

We consider a linear data assimilation (DA) framework where observations $\mathbf{y}$ are related to the state $\mathbf{x}$ through

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}),$$

with $\mathbf{H}$ the observation operator and $\mathbf{R}$ the observation error covariance matrix. Assuming a Gaussian prior $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\mathbf{xx}})$, the posterior mean (Best Linear Unbiased Estimator) is given by the Optimal Interpolation (OI) formula:

$$\mathbf{x}^\star = \mathbf{P}_{\mathbf{xx}}\mathbf{H}^{\mathrm{T}} \left(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right)^{-1} \mathbf{y}. \tag{28}$$

In large-scale spatio-temporal applications, directly manipulating the dense covariance matrix $\mathbf{P}_{\mathbf{xx}}$ becomes computationally prohibitive. An alternative consists in working with the precision matrix $\mathbf{Q}_{\mathbf{xx}} = \mathbf{P}_{\mathbf{xx}}^{-1}$, which can be sparse under appropriate modeling assumptions. Rewriting the OI problem in precision form leads to

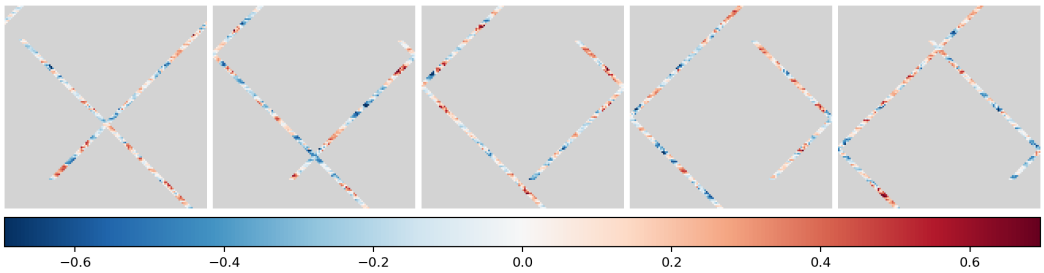
$$\mathbf{x}^\star = \left(\mathbf{Q}_{\mathbf{xx}} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1} \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}, \tag{29}$$

with posterior precision $\mathbf{Q}(\mathbf{x}|\mathbf{y}) = \mathbf{Q}_{\mathbf{xx}} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}$. When $\mathbf{Q}_{\mathbf{xx}}$ is sparse, sparse Cholesky factorizations can be used, significantly reducing computational complexity compared to dense formulations.

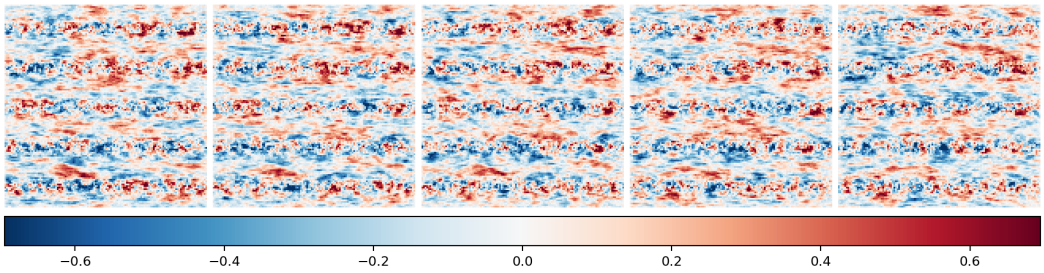
To obtain sparse precision matrices while retaining physically meaningful spatial structures, we rely on the SPDE–Gaussian process link. In its isotropic form, a Matérn Gaussian field X can be defined as the solution of

$$(\kappa^2 - \Delta)^{\alpha/2} X = \tau Z, \tag{30}$$

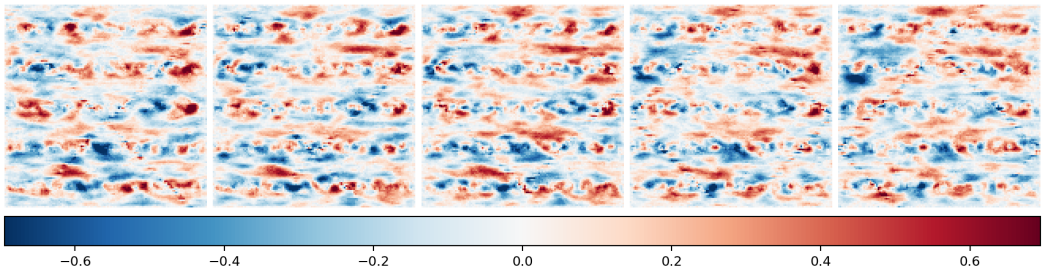
where Δ is the Laplacian operator, Z a Gaussian white noise, κ controls the spatial range, $\alpha = \nu + d/2$ the smoothness, and τ a scaling parameter related to the marginal variance. Discretization of this operator yields a sparse representation of the prior precision matrix $\mathbf{Q}_{\mathbf{xx}}$.



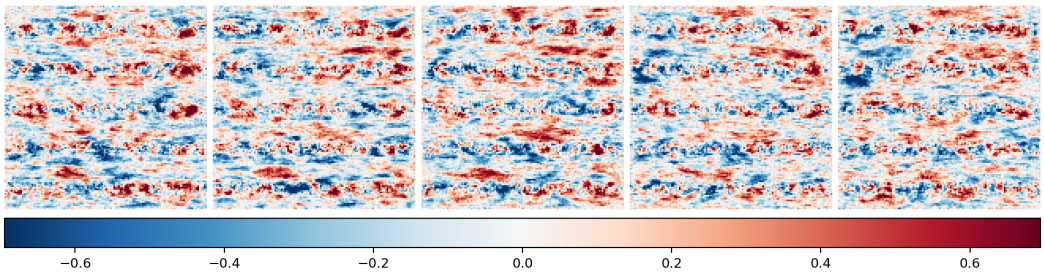
(a) Observations



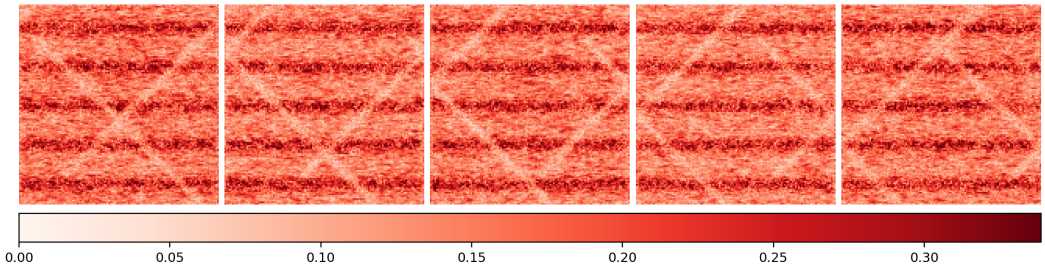
(b) Ground truth



(c) Ensemble mean

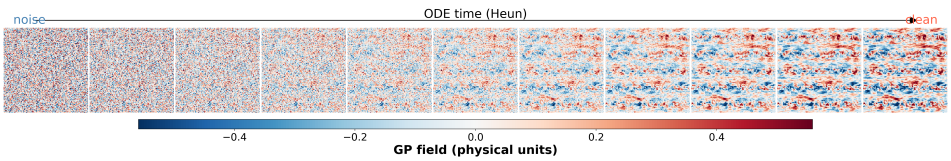


(d) Ensemble member 0

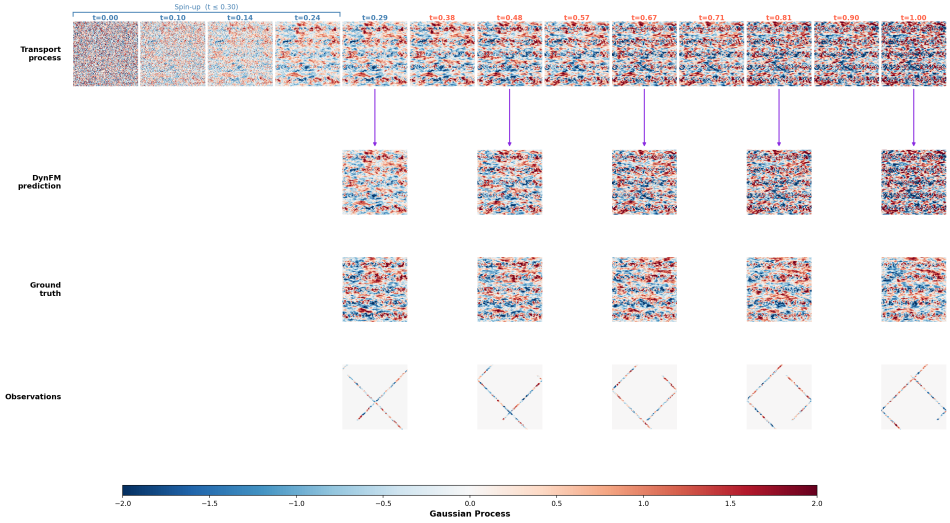


(e) Ensemble spread

Figure 4: Comparison of the observations, ground truth, ensemble mean, a representative ensemble member, and the ensemble spread for the SPDE-based calibration experiment. The ensemble mean reproduces the large-scale spatial patterns, while the ensemble spread highlights regions of greater uncertainty.



(a) FM transport



(b) DynFM transport

Figure 5: Comparison of transport estimates obtained using the FM (a) and DynFM (b) approaches. The figure highlights the differences in the spatial structure and variability of the inferred transport fields between the two methods.

In this work, we restrict to integer values of $\alpha/2$, which simplifies discretization and numerical implementation. The SPDE formulation can be extended to non-stationary and anisotropic processes by replacing the Laplacian with more general linear differential operators, such as advection–diffusion operators.

In our first application, we simulate 500 realizations of a Gaussian process driven by the following anisotropic diffusion SPDE:

$$\left\{\kappa^2 - \nabla \cdot \mathbf{H}(\mathbf{s})\nabla\right\}^{\alpha/2} \mathbf{x}(\mathbf{s}, t) = \tau \mathbf{z}(\mathbf{s}, t),$$

where $\mathbf{z}$ is a spatio-temporal Gaussian white noise. The range parameter is fixed to $\kappa = 0.33$ and the smoothness parameter to $\alpha = 4$, ensuring sufficiently smooth realizations.

The diffusion tensor $\mathbf{H}(\mathbf{s})$ introduces local anisotropy and is defined as

$$\mathbf{H}(\mathbf{s}) = \gamma \mathbf{I}_2 + \beta\, \mathbf{v}(\mathbf{s})^{\mathrm{T}} \mathbf{v}(\mathbf{s}),$$

with $\gamma = 1$, $\beta = 25$, and $\mathbf{v}(\mathbf{s}) = (v_1(\mathbf{s}), v_2(\mathbf{s}))^{\mathrm{T}}$ constructed from periodic spatial components. This parameterization generates non-stationary fields exhibiting eddy-like spatial patterns.

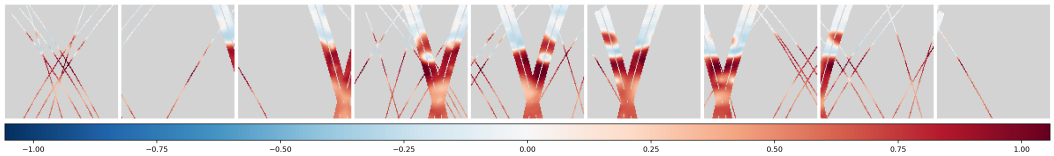
The spatial domain is $\mathcal{D} = [0, 100] \times [0, 100]$ and the temporal domain $\mathcal{T} = [0, 500]$. Both are discretized on a uniform Cartesian grid with $\Delta x = \Delta y = \Delta t = 1$. The SPDE is solved using finite differences in space and an implicit Euler scheme in time, yielding a sparse discretized operator and thus a sparse prior precision matrix $\mathbf{Q}_{\mathbf{xx}}$.

Synthetic observations are generated by subsampling the simulated states through a linear operator $\mathbf{H}$ and adding Gaussian noise with covariance $\mathbf{R}$. The SPDE-induced precision matrix is then used in the precision-based OI formulation equation 29 to produce posterior estimates.

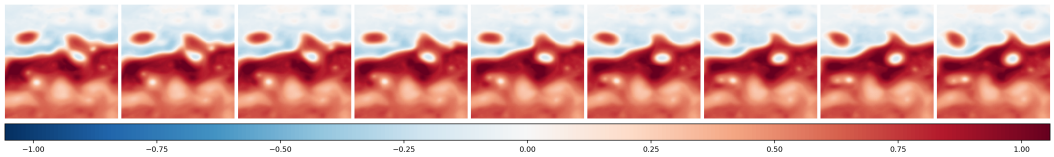
C.2 SUPERVISED TRAINING: BENCHMARKING ON SATELLITE ALTIMETRY DATA

We first consider the sea surface height (SSH) mapping benchmark from the 2020 Ocean Data Challenge (https://github.com/ocean-data-challenges/2020a_SSH_mapping_NATL60). The dataset is built from the high-resolution NATL60-CJM165 simulation, a realistic (1/60°) North Atlantic configuration of the NEMO ocean model, and provides realistic along-track satellite altimetry observations within an Observation System Simulation Experiment (OSSE) framework. The study domain covers the Gulf Stream region ((65°W)–(55°W), (33°N)–(43°N)), where mesoscale variability is particularly energetic. The benchmark consists of daily SSH fields sampled over a 63-day period. The first 21 days are used as a spin-up period, while the subsequent 42 days (corresponding to two SWOT repeat cycles) are used for evaluation. Satellite observations are generated using realistic nadir altimeter sampling (Jason-1, Envisat, Geosat Follow-On and Topex/Poseidon), optionally complemented by SWOT observations, without additional instrumental noise.

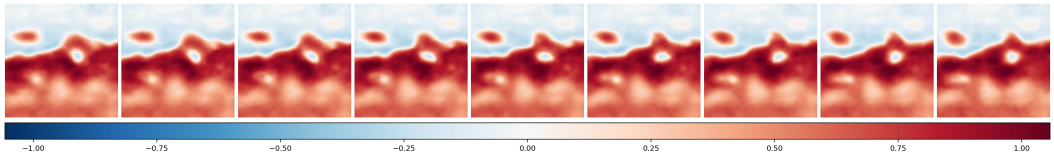
The objective is to reconstruct the daily SSH fields from these sparse along-track observations.



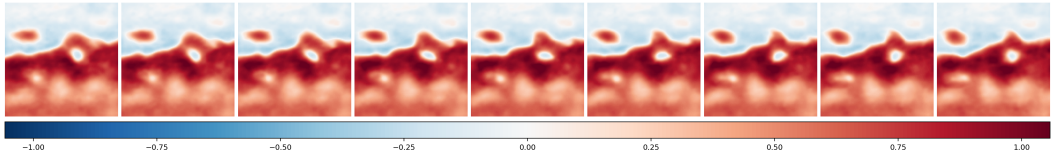
(a) Observations



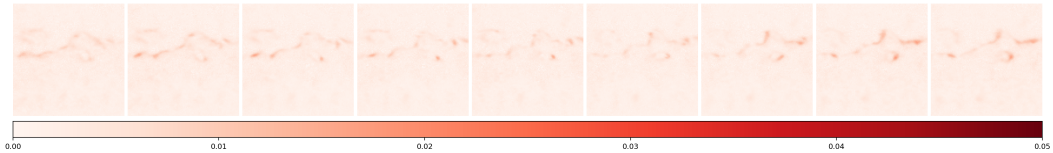
(b) Ground truth



(c) Ensemble mean



(d) Ensemble member 0



(e) Ensemble spread

Figure 6: Comparison of the observations, ground truth, ensemble mean, a representative ensemble member, and the ensemble spread for the SPDE-based calibration experiment. The ensemble mean reproduces the large-scale spatial patterns, while the ensemble spread highlights regions of greater uncertainty.

We further assess inference efficiency on the SSH Gulf Stream experiment by varying the number of neural-network function evaluations (NFE), defined as the number of network forward calls required

to generate one sample. Figure 7 compares Pairwise CM and Flow Matching (FM) in terms of reconstruction accuracy and ensemble calibration.

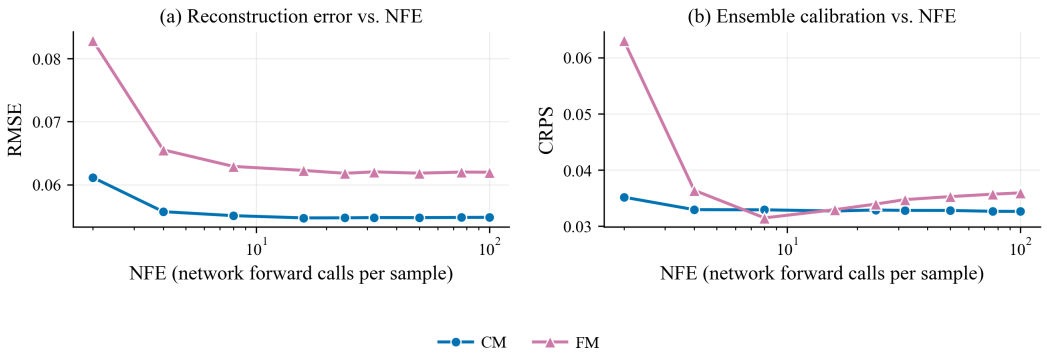
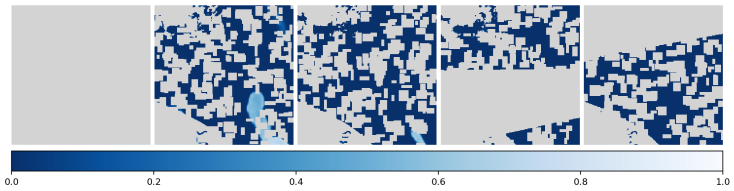


Figure 7: Inference efficiency on the SSH Gulf Stream experiment. (a) Reconstruction RMSE and (b) ensemble CRPS as a function of the number of neural-network function evaluations (NFE), defined as network forward calls per generated sample. Pairwise CM reaches stable reconstruction accuracy and ensemble calibration within only a few evaluations, whereas Flow Matching (FM) requires more evaluations to approach its best performance.

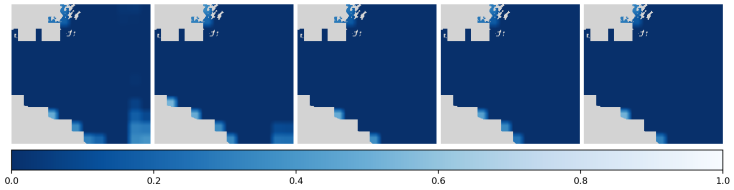
Pairwise CM reaches near-optimal reconstruction accuracy within only a few network evaluations: the RMSE decreases rapidly at low NFE and subsequently remains nearly unchanged as additional transport steps are introduced. In contrast, FM improves more progressively with increasing NFE and remains less accurate over the range considered here. This behavior illustrates the main computational motivation for learning finite-time pairwise transports: accurate state reconstruction can be achieved directly through a small number of learned transitions rather than through many successive evaluations of a local transport field.

The probabilistic behavior shows a similar robustness. The CRPS of Pairwise CM quickly stabilizes and remains nearly constant as NFE increases. FM exhibits a strong improvement at low NFE, reaches its lowest CRPS at an intermediate number of evaluations, and then slightly degrades for larger NFE. These results indicate that, on this experiment, increasing the number of Pairwise CM transport steps provides little additional benefit once the few-step regime is reached, while maintaining stable ensemble calibration.

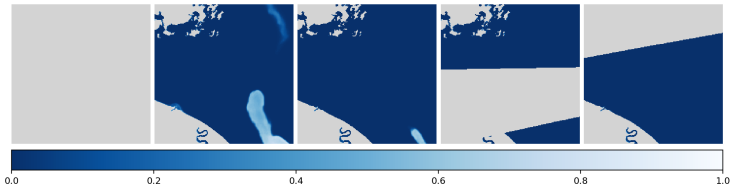
C.3 SELF-SUPERVISED TRAINING: BENCHMARKING ON SEA-ICE CONCENTRATION DATASET



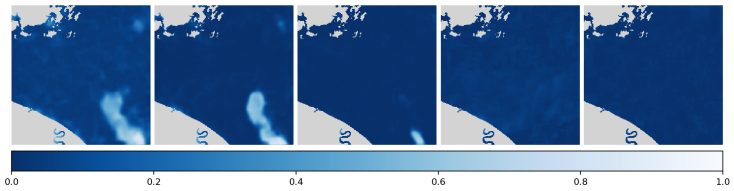
(a) Observations



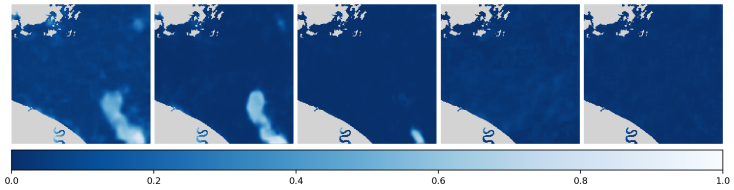
(b) Coarse observations



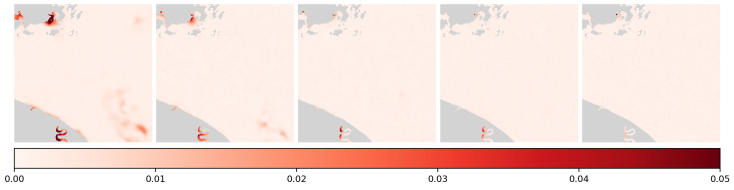
(c) Ground truth



(d) Ensemble mean



(e) Ensemble member 0



(f) Ensemble spread

Figure 8: Comparison of the high res and coarse res observations, ground truth, ensemble mean, a representative ensemble member, and the ensemble spread for the SIC-based calibration experiment

Our next benchmark considers the reconstruction and super-resolution of Arctic sea-ice concentration (SIC) from heterogeneous satellite observations. We use the high-resolution ASIP (Arctic Sea Ice Products) dataset developed at the Danish Meteorological Institute (DMI), which provides SIC

maps at 500 m spatial resolution derived from Sentinel-1 SAR imagery (Wulf et al., 2024). Owing to the limited swath of SAR acquisitions, these observations are spatially sparse and contain large gaps. As complementary large-scale observations, we use the operational OSI SAF sea-ice concentration product, which provides daily gap-free SIC fields at 12.5 km resolution derived from passive microwave observations (Lavergne et al., 2019; EUMETSAT Ocean and Sea Ice SAF, 2023). The learning problem therefore consists in reconstructing high-resolution, gap-free SIC fields by combining the accurate but sparse SAR observations with the complete but much coarser passive microwave product.

This dataset provides a realistic self-supervised super-resolution and data-fusion benchmark, representative of operational sea-ice monitoring.

D STOCHASTIC PAIRWISE CONSISTENCY MODELS

The deterministic formulations developed in the main paper learn finite-time transport operators associated with deterministic Bayesian dynamics. In particular, they approximate the probability-flow ODE underlying score-based diffusion models or the gradient flow associated with variational optimization. In both cases, the learned transport is deterministic and posterior sampling originates solely from the random initialization

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

which is subsequently transported toward the target posterior through successive applications of the learned pairwise operator.

An important empirical observation is that increasing the number of pairwise consistency steps progressively produces more stable and concentrated trajectories. This behavior is expected whenever the learned transport approximates a contractive continuous flow. For instance, when the transport field corresponds to the posterior gradient

$$D(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}),$$

and the posterior is locally log-concave, the associated dynamics satisfy

$$\frac{d\mathbf{x}}{dt} = -\mathbf{H}(\mathbf{x} - \mathbf{x}^\star), \qquad \mathbf{H} \succ 0,$$

leading to exponential convergence toward the MAP estimator. Consequently, repeated composition of deterministic finite-time transport operators naturally contracts nearby trajectories and progressively suppresses their dispersion.

Importantly, this behavior is not a limitation of pairwise consistency itself, but rather of the class of transport operators being approximated. Throughout the main paper, the learned operator approximates deterministic probability-flow dynamics. However, many Bayesian sampling algorithms are instead based on stochastic transports, including reverse-time stochastic differential equations, Langevin dynamics, or other diffusion-based samplers. In these settings, the finite-time transition between two solver times is itself stochastic and cannot, in general, be represented exactly by a deterministic mapping.

This motivates extending the pairwise framework from deterministic transport maps to stochastic transition kernels. Rather than predicting a single transported state, the learned operator now parameterizes the conditional distribution

$$p_\phi\left(\mathbf{x}^{(t')} \mid \mathbf{x}^{(t)}\right),$$

through its conditional moments. Concretely, we model

$$G_\phi\left(\mathbf{x}^{(t)}, t, t'\right) = \left(\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi\right),$$

and generate the transported state according to

$$\mathbf{x}^{(t')} = \alpha(t, t')\mathbf{x}^{(t)} + \beta(t, t')\boldsymbol{\mu}_\phi + \mathbf{L}_\phi \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

with

$$\mathbf{L}_\phi \mathbf{L}_\phi^\top = \boldsymbol{\Sigma}_\phi.$$

Within this formulation, pairwise consistency is enforced at the level of conditional distributions rather than deterministic trajectories, allowing the learned transport to approximate finite-time transitions of reverse-time SDEs while preserving the probabilistic interpretation of Bayesian transport.

D.1 Pairwise CM sampling: the reverse SDE case

The deterministic Pairwise CM can be naturally extended to stochastic Bayesian transport by approximating the reverse-time SDE

$$d\mathbf{x}^{(t)} = -D_\phi(\mathbf{x}^{(t)}, t)\, dt + \mathbf{\Gamma}_\phi^{1/2}(\mathbf{x}^{(t)}, t)\, dW_t, \tag{31}$$

where D_ϕ denotes the learned drift and $\mathbf{\Gamma}_\phi$ the diffusion tensor governing posterior uncertainty.

Rather than discretizing this SDE through infinitesimal updates, the pairwise formulation directly learns finite-time stochastic transitions between arbitrary solver times. The transport operator therefore predicts the conditional moments

$$G_\phi = (\boldsymbol{\mu}_\phi, \mathbf{\Sigma}_\phi),$$

and sampling is performed through

$$\mathbf{x}^{(t')} = \alpha(t,t')\,\mathbf{x}^{(t)} + \beta(t,t')\,\boldsymbol{\mu}_\phi(\mathbf{x}^{(t)}, t, t') + \mathbf{L}_\phi(\mathbf{x}^{(t)}, t, t')\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{32}$$

with $\mathbf{L}_\phi \mathbf{L}_\phi^\top = \mathbf{\Sigma}_\phi$.

Unlike the deterministic formulation, this stochastic transition preserves trajectory dispersion and approximates the finite-time transition kernel of the reverse SDE, enabling posterior sampling rather than only probability-flow integration.

D.2 Stochastic Variational Consistency Model

We use the increasing solver-time convention

$$0 = t_0 < t_1 < \cdots < t_K = 1,$$

where $t = 0$ denotes the noisy initialization and $t = 1$ the reconstructed state. The deterministic VarCM transition is

$$g_\phi^{\det}\left(\mathbf{x}^{(t)}, \mathbf{c}^{(t)}, t, t'\right) = c_{\text{skip}}(t,t')\,\mathbf{x}^{(t)} + c_{\text{out}}(t,t')\,\boldsymbol{\mu}_\phi\left(c_{\text{in}}(t)\mathbf{x}^{(t)}, \mathbf{d}^{(t)}, t, t'\right),$$

with conditioning

$$\mathbf{d}^{(t)} = -\nabla_{\mathbf{x}} \mathcal{J}_\Phi\left(\mathbf{x}^{(t)}, \mathbf{y}\right), \tag{33}$$

or, alternatively, direct observation conditioning.

For the linear interpolant used in our implementation,

$$\begin{cases} c_{\text{in}}(t) = \left[\sigma_{\text{data}}^2 t^2 + \sigma_{\text{noise}}^2 (1-t)^2\right]^{-1/2}, \\ c_{\text{skip}}(t,t') = \dfrac{1-t'}{1-t}, \\ c_{\text{out}}(t,t') = t' - c_{\text{skip}}(t,t')\,t. \end{cases}$$

The stochastic transition is obtained by adding a learned isotropic perturbation,

$$\mathbf{x}^{(t')} = g_\phi^{\det}\left(\mathbf{x}^{(t)}, \mathbf{c}^{(t)}, t, t'\right) + \sigma_\phi\left(\mathbf{x}^{(t)}, t\right)\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right).$$

Thus, the model predicts the deterministic transport together with a state-dependent uncertainty profile.

D.2.1 POWER-LAW UNCERTAINTY PROFILE

We parameterize the stochastic amplitude as

$$\sigma_\phi\left(\mathbf{x}^{(t)}, t\right) = \sigma_{\text{noise}}(1-t)^{a_\phi(\mathbf{x}^{(t)})}, \tag{34}$$

where

$$a_\phi(\mathbf{x}) = \text{softplus}\left(a_{\phi,\text{raw}}(\mathbf{x})\right) + a_{\min}.$$

The raw exponent is predicted from pooled decoder features through a small multilayer perceptron. This parameterization enforces

$$\sigma_\phi(\mathbf{x}^{(0)}, 0) = \sigma_{\text{noise}}, \qquad \sigma_\phi(\mathbf{x}^{(1)}, 1) = 0, \tag{35}$$

independently of the network output. The exponent controls the decay of uncertainty: $a_\phi < 1$ preserves stochasticity until late solver times, $a_\phi = 1$ yields a linear decay, and $a_\phi > 1$ concentrates uncertainty near the noisy initialization.

D.2.2 DECOUPLED MEAN AND UNCERTAINTY LEARNING

Joint Gaussian likelihood training may weaken the gradient applied to the predicted mean, since

$$\frac{\partial \mathcal{L}_{\text{NLL}}}{\partial \boldsymbol{\mu}} = -\frac{\mathbf{x}^\star - \boldsymbol{\mu}}{\sigma^2}.$$

Large predicted variances may therefore reduce the mean-learning signal and allow the model to improve the likelihood by inflating uncertainty rather than refining the transport.

We instead retain the deterministic VarCM losses for the mean and use a separate likelihood term to calibrate the uncertainty branch:

$$\mathcal{L}_{\text{StochVarCM}} = \lambda_{\text{pair}}\mathcal{L}_{\text{pair}} + \lambda_{\text{long}}\mathcal{L}_{\text{long}} + \lambda_{\text{short}}\mathcal{L}_{\text{short}} + \lambda_{\text{AE}}\mathcal{L}_{\text{AE}} + \lambda_\sigma\mathcal{L}_\sigma. \tag{36}$$

The first four terms are identical to VarCM and provide full gradients to the deterministic transport. The uncertainty loss is

$$\begin{aligned}\mathcal{L}_\sigma = \mathcal{L}_{\text{NLL}}&\left(\mathbf{x}_0, \text{sg}\left[\widehat{\mathbf{x}}_{\text{long}}\right], \sigma_{\text{long}}\right) \\ &+ \mathcal{L}_{\text{NLL}}\left(\text{sg}\left[\mathbf{x}_{\text{target}}\right], \text{sg}\left[\widehat{\mathbf{x}}_{\text{short}}\right], \sigma_{\text{short}}\right),\end{aligned}$$

where $\text{sg}[\cdot]$ denotes stop-gradient. For a state of dimension d,

$$\mathcal{L}_{\text{NLL}}\left(\mathbf{x}^\star, \boldsymbol{\mu}, \sigma\right) = \frac{\left\|\mathbf{x}^\star - \boldsymbol{\mu}\right\|^2}{2\sigma^2} + \frac{d}{2}\log\sigma^2.$$

At the optimum,

$$\sigma_\star^2 = \frac{1}{d}\mathbb{E}\left[\left\|\mathbf{x}^\star - \boldsymbol{\mu}\right\|^2\right],$$

so that the learned variance matches the effective quadratic error of the deterministic prediction.

D.2.3 STOCHASTIC SAMPLING

Starting from

$$\mathbf{x}^{(t_0)} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\text{noise}}^2 \mathbf{I}\right),$$

sampling follows

$$\begin{aligned}\mathbf{x}^{(t_{k+1})} = {} & c_{\text{skip}}\left(t_k, t_{k+1}\right) \mathbf{x}^{(t_k)} \\ & + c_{\text{out}}\left(t_k, t_{k+1}\right) \boldsymbol{\mu}_\phi\left(\mathbf{x}^{(t_k)}, \mathbf{c}^{(t_k)}, t_k, t_{k+1}\right) \\ & + \sigma_\phi\left(\mathbf{x}^{(t_k)}, t_k\right) \boldsymbol{\varepsilon}_k, \qquad \boldsymbol{\varepsilon}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right).\end{aligned}$$

Setting the stochastic term to zero recovers deterministic VarCM, whose ensemble diversity arises only from the initial latent state. StochVarCM additionally injects independent perturbations at each solver step.

D.2.4 ALTERNATIVE COVARIANCE PARAMETERIZATIONS

The implemented power-law model corresponds to the isotropic covariance

$$\boldsymbol{\Sigma}_\phi\left(\mathbf{x}^{(t)}, t\right) = \sigma_\phi^2\left(\mathbf{x}^{(t)}, t\right) \mathbf{I}. \tag{37}$$

It is inexpensive and imposes meaningful boundary conditions, but more expressive uncertainty structures may be introduced within the same framework.

Table 2: Possible covariance parameterizations for stochastic pairwise transport.

Type	**Form**	**Learned quantity**	**Status**
Power-law isotropic	$\sigma_{\text{noise}}^2(1-t)^{2a_\phi(\mathbf{x})}\mathbf{I}$	scalar exponent	presented here
Free isotropic	$\sigma_\phi^2(\mathbf{x}, t, t')\mathbf{I}$	scalar log-variance	alternative
Diagonal	$\text{diag}(\sigma_{\phi,1}^2, \ldots, \sigma_{\phi,d}^2)$	component-wise variances	extension
Structured precision	$\boldsymbol{\Sigma}_\phi^{-1} = a_\phi(t, t')\mathcal{L}_\phi$	precision operator	extension

A free isotropic model predicts one unconstrained variance per transition and is trained using

$$\mathcal{L}_{\text{iso}} = \mathbb{E}\left[\frac{\left\|\mathbf{x}^{(t')} - \boldsymbol{\mu}_\phi\right\|^2}{2\sigma_\phi^2} + \frac{d}{2}\log\sigma_\phi^2\right]. \tag{38}$$

A diagonal covariance,

$$\boldsymbol{\Sigma}_\phi = \text{diag}\left(\sigma_{\phi,1}^2, \ldots, \sigma_{\phi,d}^2\right), \tag{39}$$

captures heterogeneous uncertainty but neglects cross-correlations.

Finally, correlated uncertainty may be represented through a structured precision operator,

$$\boldsymbol{\Sigma}_\phi^{-1} = a_\phi(t, t')\mathcal{L}_\phi, \tag{40}$$

where $\mathcal{L}_\phi$ is a positive-definite differential or convolutional operator. This formulation is consistent with SPDE-based priors in variational data assimilation and may locally approximate the inverse Hessian of the variational objective,

$$\boldsymbol{\Sigma} \approx \left[\nabla^2_{\mathbf{x}} \mathcal{J}_\Phi \left(\mathbf{x}^\star, \mathbf{y}\right)\right]^{-1}. \tag{41}$$

These extensions provide a path toward spatially correlated posterior uncertainty while preserving the finite-time pairwise transport formulation.

D.3 Stochastic extensions of DynCM and DynVarCM

The stochastic extension of the dynamic models depends on whether an explicit forward diffusion process is available. In both cases, the state is denoted by $\mathbf{x}_k^{(t)}$, where $t \in [0,1]$ is solver time and k is the associated physical-time index. The interval $[0, t_{\mathrm{IC}}]$ corresponds to the reconstruction of the initial condition, while $[t_{\mathrm{IC}}, 1]$ represents the subsequent physical evolution.

For DynCM the spin-up regime is generated from a prescribed diffusion process. The standard reverse-time SDE, or its probability-flow ODE counterpart, may therefore be retained on $[0, t_{\mathrm{IC}}]$. Only the stochasticity associated with the physical regime must be learned. The hybrid dynamics may be summarized as

$$\mathbf{x}_{k'}^{(t')} = \begin{cases} \mathcal{S}^{\mathrm{rev}}_{t \to t'} \left(\mathbf{x}_0^{(t)}\right), & 0 \le t < t' \le t_{\mathrm{IC}}, \\ g_\theta^{\mathrm{det}} \left(\mathbf{x}_k^{(t)}, \mathbf{c}_k^{(t)}, t, t'\right) + \mathbf{L}_\psi \left(\mathbf{x}_k^{(t)}, t, t'\right) \boldsymbol{\varepsilon}, & t_{\mathrm{IC}} \le t < t' \le 1, \end{cases} \tag{42}$$

where $\mathcal{S}^{\mathrm{rev}}_{t \to t'}$ denotes the prescribed reverse-diffusion transition and

$$\boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right), \qquad \mathbf{L}_\psi \mathbf{L}_\psi^\top = \boldsymbol{\Sigma}_\psi. \tag{43}$$

Thus, the spin-up uncertainty follows the known diffusion model, whereas the covariance of the physical transport is learned from the variability of the physical targets.

For DynVarCM, neither the spin-up flow nor the physical transport is prescribed. The complete diffusion–physical timeline must therefore be learned through variational self-consistency. The stochastic transition is defined over the full interval as

$$\mathbf{x}_{k'}^{(t')} = g_\theta^{\mathrm{det}} \left(\mathbf{x}_k^{(t)}, \nabla_{\mathbf{x}} \mathcal{J}_\Phi \left(\mathbf{x}_k^{(t)}, t; \mathbf{y}_{0:N}\right), t, t'\right) + \mathbf{L}_\psi \left(\mathbf{x}_k^{(t)}, t, t'\right) \boldsymbol{\varepsilon}, \qquad 0 \le t < t' \le 1, \tag{44}$$

In this case, both the deterministic mean transport and its uncertainty are learned jointly over the spin-up and physical regimes. Composition and anchoring constraints retain the same structure as in deterministic DynVarCM.

For both models, mean estimation and uncertainty calibration are decoupled. The deterministic objective remains

$$\mathcal{L}^{\mathrm{dyn}}_{\mathrm{mean}} = \lambda_{\mathrm{comp}} \mathcal{L}^{\mathrm{dyn}}_{\mathrm{comp}} + \lambda_{\mathrm{IC}} \mathcal{L}_{\mathrm{IC}} + \lambda_{\mathrm{phys}} \mathcal{L}_{\mathrm{phys}}, \tag{45}$$

while a separate likelihood term calibrates the covariance with stop-gradient through the predicted mean,

$$\mathcal{L}_{\mathrm{StochDyn}} = \mathcal{L}^{\mathrm{dyn}}_{\mathrm{mean}} + \lambda_\Sigma \mathcal{L}^{\mathrm{dyn}}_\Sigma. \tag{46}$$

For DynCM, $\mathcal{L}^{\mathrm{dyn}}_\Sigma$ is required only for transitions in the physical regime, since spin-up stochasticity is prescribed by the reverse SDE. For DynVarCM, it is evaluated throughout the complete solver interval because both stochastic regimes are learned self-consistently.

This distinction yields two complementary extensions: StochDynCM combines a prescribed stochastic spin-up with learned stochastic physical propagation, whereas StochDynVarCM learns the complete stochastic diffusion–physical transport without relying on any explicit forward process.

E NORMALIZATION AND IMPLEMENTATION DETAILS

We detail here the normalization and preconditioning used in the implementations. All consistency-based models rely on a pairwise operator $g_\theta(\mathbf{x}^{(t)}, t, t')$ mapping a state at solver time t to a state at $t' \leq t$ along the same trajectory. CM, VarCM, DynCM, and DynVarCM share the same algebraic construction and differ only in their solver-time schedule, the spin-up/physical-time decomposition, and the conditioning of the transport network.

E.1 A SINGLE DETERMINISTIC TRANSPORT IDENTITY

Let $\mathbf{x}^{(0)}$ denote the clean state and $\mathbf{x}^{(1)}$ the fully corrupted state. For a generic solver time $t \in [0, 1]$ ($t = 0$: clean, $t = 1$: corrupted), we define the interpolant

$$\mathbf{x}^{(t)} = a(t)\mathbf{x}^{(1)} + b(t)\mathbf{x}^{(0)}, \qquad a(0) = 0, \quad b(0) = 1.$$

Evaluating the same trajectory at $t' \leq t$ gives the exact deterministic transport

$$\mathbf{x}^{(t')} = \alpha(t, t')\,\mathbf{x}^{(t)} + \beta(t, t')\,\mathbf{x}^{(0)}, \qquad \alpha(t, t') = \frac{a(t')}{a(t)}, \qquad \beta(t, t') = b(t') - \frac{a(t')}{a(t)}\,b(t). \quad (47)$$

Since $\mathbf{x}^{(0)}$ is unknown at inference, it is replaced by the learned prediction,

$$g_\theta(\mathbf{x}^{(t)}, t, t') = \alpha(t, t')\,\mathbf{x}^{(t)} + \beta(t, t')\,G_\theta(\mathbf{x}^{(t)}, \mathbf{c}, t, t'), \quad (48)$$

where $\mathbf{c}$ denotes an optional conditioning signal. All consistency-based models below correspond to this same pairwise transport evaluated with different schedules and conditioning strategies.

E.2 CM: NOISE-LEVEL SCHEDULE

CM uses the common solver coordinate t, parameterized through the Karras noise schedule $\sigma(t)$. With the Variance-Exploding interpolant

$$\mathbf{x}^{(t)} = \mathbf{x}^{(0)} + \sigma(t)\boldsymbol{\varepsilon},$$

Eq. equation 47 gives

$$\alpha(t, t') = \frac{\sigma(t')}{\sigma(t)}, \qquad \beta(t, t') = 1 - \alpha(t, t'). \quad (49)$$

E.3 VARCM: GENERAL SCHEDULE WITH VARIATIONAL CONDITIONING

VarCM uses normalized solver time $u = t \in [0, 1]$ and allows a general (a, b) interpolant. We consider either the linear schedule $a(t) = t,\ b(t) = 1 - t$, yielding $\alpha(t, t') = t'/t$, or the trigonometric schedule $a(t) = \sin(\pi t/2),\ b(t) = \cos(\pi t/2)$. The corresponding $\alpha(t, t')$ and $\beta(t, t')$ follow directly from Eq. equation 47.

As described in the main text, the transport can additionally be conditioned by the learned variational geometry associated with

$$\mathcal{J}_\phi(\mathbf{x}, \mathbf{y}) = \mathcal{J}_{\text{obs}}(\mathbf{x}, \mathbf{y}) + \lambda\,\mathcal{J}_{\text{prior},\phi}(\mathbf{x}).$$

The conditioning $\mathbf{c}$ contains either $\nabla_{\mathbf{x}}\mathcal{J}_\phi$, local observations and masks, or both. It is projected independently and added to the state representation, ensuring that the variational signal remains active even when its magnitude is small.

E.4 DYNCM: REGIME-ADAPTIVE SCHEDULE FOR DYNAMICAL SYSTEMS

DynCM extends the same transport to a sequence of C physical states by splitting the solver timeline at a boundary s_b. In the implementation, the solver coordinate follows the CM convention ($t = 1$: noisy initialization, $t = 0$: terminal state), so spin-up and physical evolution correspond to $t > s_b$ and $t \leq s_b$, respectively:

$$\alpha(t, t') = \begin{cases} \sigma_{\text{sp}}(t')/\sigma_{\text{sp}}(t), & t > s_b \quad \text{(spin-up)}, \\ t'/t, & t \leq s_b \quad \text{(physical)}, \end{cases} \qquad \beta(t, t') = 1 - \alpha(t, t'). \quad (50)$$

Here σ_{sp} maps the spin-up interval $[s_b, 1]$ to $[\sigma_{\min}, \sigma_{\max}]$ through the Karras schedule, while the physical branch uses the linear pairwise schedule. Thus, a single pairwise operator implements the unified spin-up–physical transport introduced in the main text. No variational conditioning is used.

E.5 DynVarCM: regime-adaptive schedule with variational conditioning

DynVarCM combines the dynamical and variational extensions. It retains the DynCM physical branch but uses a boundary-relative solver coordinate during spin-up,

$$\alpha(t,t') = \begin{cases} \dfrac{t'-s_b}{t-s_b}, & t > s_b \quad \text{(spin-up)}, \\ t'/t, & t \le s_b \quad \text{(physical)}, \end{cases} \qquad \beta(t,t') = 1-\alpha(t,t'). \tag{51}$$

This corresponds to Eq. equation 47 with $a(t) = t - s_b$ on $[s_b, 1]$ and telescopes exactly to zero at the spin-up/physical boundary, independently of the number of sampling steps. Variational conditioning $\mathbf{c}$ is activated in the physical regime, where it represents the local posterior geometry, and set to zero during spin-up.

E.6 FM: linear interpolant velocity regression

For comparison, Flow Matching directly learns the local velocity rather than a finite-time pairwise transport. It uses the opposite time convention, $t = 0$ for noise and $t = 1$ for data. For $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I})$,

$$\mathbf{x}_t = t\,\mathbf{x}_1 + (1-t)\boldsymbol{\varepsilon}, \qquad \mathbf{v}_{\mathrm{tgt}} = \mathbf{x}_1 - \boldsymbol{\varepsilon} = \frac{d\mathbf{x}_t}{dt}. \tag{52}$$

The network directly regresses $G_\theta(\mathbf{x}_t, \mathbf{y}, t) \approx \mathbf{v}_{\mathrm{tgt}}$, without a skip connection. Training uses a heteroscedastic Gaussian NLL with a per-channel, time-conditioned learned log-scale, optionally censored at physical bounds. At inference, the learned ODE is integrated from $t = 0$ to $t = 1$ using a second-order Heun solver.

E.7 DynFM: regime-adaptive velocity regression

DynFM extends FM to a C-frame sequence using the same spin-up/physical decomposition as the dynamical consistency models, while retaining the FM time convention. The interval is split at a spin-up fraction s_f, with local coordinates

$$\tau_{\mathrm{sp}}(t) = t/s_f, \qquad \tau_{\mathrm{ph}}(t) = (t-s_f)/(1-s_f).$$

Spin-up interpolates from noise to the initial state $\mathbf{f}_0$, while the physical regime interpolates between consecutive states $\mathbf{f}_k, \mathbf{f}_{k+1}$. Since each segment is affine, its velocity is $d\mathbf{x}_t/dt = \Delta\,\dot\tau(t)$. Defining the local velocity scaling $\gamma(t) = \dot\tau(t)$ gives

$$\gamma(t) = \begin{cases} 1/s_f, & t < s_f \quad \text{(spin-up)}, \\ (C-1)/(1-s_f), & t \ge s_f \quad \text{(physical)}. \end{cases} \tag{53}$$

The network is trained on the normalized increment $\mathbf{v}_{\mathrm{tgt}}/\gamma(t) = \Delta$, while the physical velocity is recovered as $\mathbf{v} = \gamma(t) G_\theta(\mathbf{x}_t, \mathbf{y}, t)$. Sampling uses a boundary-aware Heun grid containing the regime boundary and all physical frame times, preventing integration steps from crossing a $\gamma(t)$ discontinuity.

F LINEAR–GAUSSIAN LIMIT OF DYNVARCM

We consider a spatial field $\mathbf{x}(t) \in \mathbb{R}^d$ defined on a discretized grid of size $d = N_x N_y$. The evolution of the field is governed by a linear stochastic partial differential equation (SPDE)

$$\frac{\partial \mathbf{x}}{\partial t} = -\mathcal{A}\mathbf{x} + \sigma\,\eta(t), \tag{54}$$

where $\mathcal{A}$ denotes a symmetric positive-definite linear spatial operator (e.g. a diffusion operator), σ controls the amplitude of the stochastic forcing, and $\eta(t)$ is Gaussian white noise in time. Such stochastic PDEs are widely used to describe the evolution of spatial random fields subject to both deterministic dissipation and random perturbations.

To obtain a tractable generative formulation, we discretize the dynamics in time using an implicit Euler scheme with time step Δt. The discretized update from time t_i to t_{i+1} reads

$$\mathbf{x}_{i+1} = \mathbf{M}\,\mathbf{x}_i + \mathbf{M}\,\sigma\sqrt{\Delta t}\,\mathbf{z}_i, \qquad \mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}), \tag{55}$$

where the transition matrix is

$$\mathbf{M} = (\mathbf{I} + \Delta t\,\mathbf{A})^{-1}, \tag{56}$$

and $\mathbf{A}$ is the discrete representation of the spatial operator $\mathcal{A}$. Defining the incremental noise scale $\sigma_\varepsilon = \sigma\sqrt{\Delta t}$, the transition matrix $\mathbf{T} = \sigma_\varepsilon\,\mathbf{M}$, and the incremental noise covariance $\mathbf{Q} = \mathbf{T}\mathbf{T}^\top = \sigma_\varepsilon^2\,\mathbf{M}\mathbf{M}^\top$, the conditional transition distribution is

$$\mathbf{x}_{i+1} \mid \mathbf{x}_i \;\sim\; \mathcal{N}(\mathbf{M}\mathbf{x}_i,\; \mathbf{Q})\,.$$

Because the dynamics are linear and the noise is Gaussian, the marginal distribution of $\mathbf{x}_i$ remains Gaussian at all times, $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{P}_i)$, and the covariance matrix evolves according to the closed-form Lyapunov recursion

$$\mathbf{P}_{i+1} = \mathbf{M}\,\mathbf{P}_i\,\mathbf{M}^\top + \mathbf{Q}, \qquad \mathbf{P}_0 = \mathbf{I}.$$

Because $\mathbf{M} = (\mathbf{I}+\Delta t\,\mathbf{A})^{-1}$ is a strong contraction when Δt is small and $\mathbf{A}$ has a positive spectrum, each application of $\mathbf{M}$ reduces variance, and $\mathbf{P}_i$ decreases monotonically toward the stationary covariance $\mathbf{P}_\infty \ll \mathbf{I}$.

F.1 CONNECTION TO DIFFUSION MODELS.

This formulation is structurally identical to the forward process used in diffusion probabilistic models, where data are progressively corrupted by Gaussian noise according to a Markov chain

$$\mathbf{x}_{i+1} = \mathbf{F}_i\,\mathbf{x}_i + \mathbf{\Sigma}_i^{1/2}\boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \mathbf{I}). \tag{57}$$

The discretized SPDE is recovered by identifying $\mathbf{F}_i = \mathbf{M}$ and $\mathbf{\Sigma}_i = \mathbf{Q}$. Therefore, the implicit-Euler discretization of a stochastic PDE naturally defines a diffusion process whose transition kernel is fully determined by the underlying physical operator.

F.2 SCORE FUNCTION.

Because the marginal distribution is Gaussian, the score function admits a closed-form expression at each step,

$$\nabla_{\mathbf{x}} \log p_i(\mathbf{x}) = -\mathbf{P}_i^{-1}\mathbf{x}, \tag{58}$$

which eliminates the need to learn the score with a neural network, as is required in standard score-based diffusion models.

F.3 Exact Gaussian reverse process and equivalence with Anderson's reverse-time SDE.

Because all marginals are Gaussian, the reverse conditional $p(\mathbf{x}_{i-1} \mid \mathbf{x}_i)$ can be derived exactly from the joint Gaussian structure of $(\mathbf{x}_{i-1}, \mathbf{x}_i)$ via Bayes' rule. The joint distribution is

$$\begin{pmatrix} \mathbf{x}_{i-1} \\ \mathbf{x}_i \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0},\, \begin{pmatrix} \mathbf{P}_{i-1} & \mathbf{P}_{i-1}\mathbf{M}^\top \\ \mathbf{M}\mathbf{P}_{i-1} & \mathbf{P}_i \end{pmatrix}\right),$$

where the cross-covariance $\mathbb{E}[\mathbf{x}_{i-1}\mathbf{x}_i^\top] = \mathbf{P}_{i-1}\mathbf{M}^\top$ follows directly from the forward transition $\mathbf{x}_i = \mathbf{M}\mathbf{x}_{i-1} + \mathbf{T}\mathbf{z}_{i-1}$. Conditioning on $\mathbf{x}_i$ via the Gaussian conditional formula gives

$$\mathbf{x}_{i-1} \mid \mathbf{x}_i \;\sim\; \mathcal{N}(\mathbf{G}_i\,\mathbf{x}_i,\; \mathbf{\Sigma}_{\text{back},i})\,,$$

with the Kalman smoother (RTS) gain and conditional covariance

$$\mathbf{G}_i = \mathbf{P}_{i-1}\,\mathbf{M}^\top\,\mathbf{P}_i^{-1}, \tag{59}$$

$$\mathbf{\Sigma}_{\text{back},i} = \mathbf{P}_{i-1} - \mathbf{P}_{i-1}\,\mathbf{M}^\top\,\mathbf{P}_i^{-1}\,\mathbf{M}\,\mathbf{P}_{i-1}. \tag{60}$$

F.4 Equivalence with the reverse-time SDE

The discrete-time reverse process of Anderson (Anderson, 1982) for the forward Markov chain $\mathbf{x}_i = \mathbf{M}\mathbf{x}_{i-1} + \mathbf{T}\mathbf{z}_{i-1}$ reads

$$\mathbf{x}_{i-1} = \mathbf{x}_i + \underbrace{\left[(\mathbf{M}-\mathbf{I})\mathbf{x}_i - \mathbf{Q}\,\mathbf{P}_i^{-1}\mathbf{x}_i\right]}_{\text{reverse drift } d_i(\mathbf{x}_i)} + \mathbf{T}\,\mathbf{z}_i, \qquad \mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}),$$

where $\mathbf{Q} = \mathbf{T}\mathbf{T}^\top$. This is a Gaussian with mean and covariance

$$\begin{aligned} \boldsymbol{\mu}_{\text{Anderson}} &= \left[\mathbf{I} + (\mathbf{M}-\mathbf{I}) - \mathbf{Q}\mathbf{P}_i^{-1}\right]\mathbf{x}_i = \left(\mathbf{M} - \mathbf{Q}\mathbf{P}_i^{-1}\right)\mathbf{x}_i, \\ \mathbf{\Sigma}_{\text{Anderson}} &= \mathbf{Q}. \end{aligned}$$

We now show these are identical to equation 59–equation 60. Starting from the Lyapunov recursion $\mathbf{P}_i = \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top + \mathbf{Q}$, we have

$$\mathbf{G}_i = \mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1} = \mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1}.$$

The mean of the exact conditional can be rewritten as

$$\mathbf{G}_i\mathbf{x}_i = \mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1}\mathbf{x}_i = \left(\mathbf{M} - \mathbf{Q}\mathbf{P}_i^{-1}\right)\mathbf{x}_i,$$

where the last equality follows from substituting $\mathbf{P}_{i-1}\mathbf{M}^\top = \mathbf{M}^{-1}(\mathbf{P}_i - \mathbf{Q})$, which gives

$$\mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1} = \mathbf{M}^{-1}(\mathbf{P}_i - \mathbf{Q})\mathbf{P}_i^{-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{Q}\mathbf{P}_i^{-1} = \mathbf{M}^{-1}(\mathbf{I} - \mathbf{Q}\mathbf{P}_i^{-1}).$$

Since $\mathbf{M} = (\mathbf{I} + \Delta t\,\mathbf{A})^{-1}$, we have $\mathbf{M}^{-1} = \mathbf{I} + \Delta t\,\mathbf{A}$. However, a cleaner route uses $\mathbf{P}_{i-1}\mathbf{M}^\top = (\mathbf{P}_i - \mathbf{Q})\mathbf{M}^{-\top}$ from the transpose of the Lyapunov recursion, so

$$\begin{aligned} \mathbf{G}_i &= (\mathbf{P}_i - \mathbf{Q})\mathbf{M}^{-\top}\mathbf{M}^{-1}\cdot\mathbf{M}\mathbf{P}_i^{-1} \\ &= (\mathbf{I} - \mathbf{Q}\mathbf{P}_i^{-1})\mathbf{M}\mathbf{P}_i^{-1}\mathbf{P}_i \\ &= \mathbf{M} - \mathbf{Q}\mathbf{P}_i^{-1}, \end{aligned} \tag{61}$$

where the last step used $\mathbf{M}$ symmetric (since $\mathbf{A}$ is symmetric). Hence $\boldsymbol{\mu}_{\text{Anderson}} = \mathbf{G}_i \mathbf{x}_i$.

For the covariance, substituting $\mathbf{G}_i = \mathbf{M} - \mathbf{Q}\mathbf{P}_i^{-1}$ into equation 60:

$$\begin{aligned}
\boldsymbol{\Sigma}_{\text{back}} &= \mathbf{P}_{i-1} - \mathbf{G}_i \, \mathbf{M} \, \mathbf{P}_{i-1} \\
&= \mathbf{P}_{i-1} - (\mathbf{M} - \mathbf{Q}\mathbf{P}_i^{-1})\mathbf{M}\mathbf{P}_{i-1} \\
&= \mathbf{P}_{i-1} - \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top \cdot (\mathbf{M}^{-\top}\mathbf{M}^{-1}\mathbf{P}_{i-1}^{-1}\mathbf{P}_{i-1}) + \mathbf{Q}\mathbf{P}_i^{-1}\mathbf{M}\mathbf{P}_{i-1} \\
&= \mathbf{P}_{i-1} - \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top + \mathbf{Q}\mathbf{P}_i^{-1}\mathbf{M}\mathbf{P}_{i-1} \\
&= \mathbf{Q} - \mathbf{Q}\mathbf{P}_i^{-1}(\mathbf{P}_i - \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top) + \mathbf{Q}\mathbf{P}_i^{-1}\mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top \cdot \ldots
\end{aligned}$$

More directly, using $\mathbf{P}_{i-1} - \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top = \mathbf{P}_{i-1} - (\mathbf{P}_i - \mathbf{Q}) = \mathbf{P}_{i-1} - \mathbf{P}_i + \mathbf{Q}$, and noting that at stationarity (or by direct substitution of the Schur complement identity):

$$\boldsymbol{\Sigma}_{\text{back}} = \mathbf{P}_{i-1} - \mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1}\mathbf{M}\mathbf{P}_{i-1} = \mathbf{Q} = \boldsymbol{\Sigma}_{\text{Anderson}}. \quad \checkmark \tag{62}$$

The last equality is the Schur complement identity for the joint Gaussian block matrix: $\boldsymbol{\Sigma}_{\text{back}} = \mathbf{P}_{i-1} - \mathbf{P}_{i-1}\mathbf{M}^\top\mathbf{P}_i^{-1}\mathbf{M}\mathbf{P}_{i-1} = \mathbf{Q}$, which follows directly from $\mathbf{P}_i = \mathbf{M}\mathbf{P}_{i-1}\mathbf{M}^\top + \mathbf{Q}$ by the matrix inversion lemma.

Therefore, the exact Gaussian conditionnal equation 59–equation 60 and the Anderson reverse-time Markov chain equation F.4 define **identical distributions** for all Δt, with no approximation. The two formulations are equivalent exact reverse processes for the linear Gaussian SPDE dynamics.

In practice, we implement the RTS smoother form equation 59–equation 60 rather than the Anderson form equation F.4, as it is numerically more stable: the mean is computed via a single Cholesky solve $\mathbf{G}_i\mathbf{x}_i = \mathbf{P}_{i-1}\mathbf{M}^\top(\mathbf{L}_i\mathbf{L}_i^\top)^{-1}\mathbf{x}_i$, and sampling is performed via the Cholesky factorization of $\boldsymbol{\Sigma}_{\text{back},i}$.

F.5 Duality between physical and solver time.

A crucial observation is that the notions of *forward* and *reverse* depend on which notion of time is considered. Let k denote the physical time index of the SPDE and i the solver time index used in the generative formulation. The physical dynamics evolve *forward* in time $t_k : 0 \to T$, describing the evolution of the stochastic field from a white-noise initial condition $\mathbf{P}_0 = \mathbf{I}$ toward a smooth, diffusion-filtered state with SPDE-prescribed variance $\mathbf{P}_T$.

The SPDE forward dynamics thus play the role of the *reverse* diffusion process in the generative modeling sense: they progressively reduce variance (denoise). Conversely, the reverse-time SPDE sampler described above plays the role of the *forward* diffusion process: it progressively increases variance, mapping the smooth state back to the white-noise distribution $\mathcal{N}(0, \mathbf{I})$.

This forward–reverse duality is illustrated in Figure 9. The lower row shows the forward process (physical time $k : 0 \to T$, increasing left to right), which progressively smooths white noise into a structured spatial field. The upper row shows the reverse process (SPDE solver time $i : T \to 0$, physical time increasing right to left), which progressively restores structure from a smooth field toward white noise.

This establishes a direct connection between stochastic physical dynamics and diffusion-based generative models. In the linear Gaussian setting considered here, the full probabilistic structure is available analytically: marginals remain Gaussian, covariances follow a closed-form recursion, and the score function is explicit. The reverse dynamics therefore follow directly from the SPDE without requiring neural score estimation, providing a principled and physically interpretable class of diffusion processes over spatial fields.

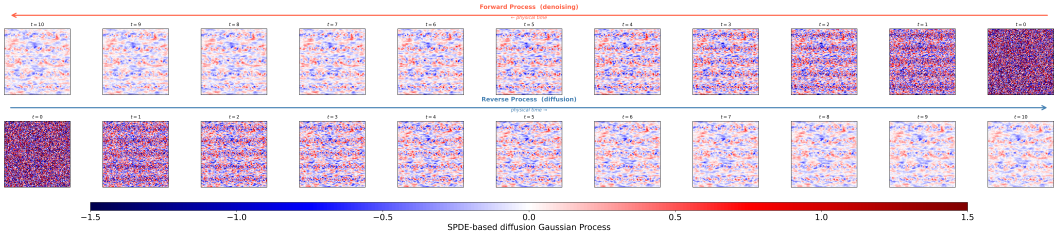


Figure 9: Forward and reverse processes of the SPDE-based diffusion model. *Lower row (Forward Process, physical time →):* starting from a white-noise sample $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$, the implicit-Euler SPDE dynamics progressively reduce variance and introduce spatial structure. *Upper row (Reverse Process, physical time ←):* the exact Gaussian reverse (RTS smoother) maps the smooth state back toward $\mathcal{N}(0, \mathbf{I})$, recovering the white-noise distribution. Solver time i increases left to right in both rows; physical time runs in opposite directions for the two processes.