CHRONOSYNC: RECOVERING LONG-HORIZON COUPLED DYNAMICS VIA CROSS-FIELD SYNCHRONIZATION OF DECOUPLED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon prediction of coupled multiphysics PDEs is challenging even when fully coupled trajectories are available for training. We study a harder setting in which the target joint dynamics are unavailable during training and only decoupled conditional trajectories for individual physical fields are observed. We introduce ChronoSync, an inference algorithm that recovers coupled dynamics by synchronizing independently trained conditional Elucidated Rolling Diffusion Models. ChronoSync sequentially composes field-wise probability flows within each denoising step using operator splitting, while a rolling temporal window propagates the synchronized state over long PDE time horizons. We characterize the local composition perturbation through Lie brackets and derive a recursive bound for its propagation across rollouts. Under a mixed regime which allows for varying degrees of local rollout stability in the perturbations, we show that the long-horizon splitting error can remain bounded. We evaluate ChronoSync on FitzHugh–Nagumo, viscous Burgers, Darcy transport, and Rayleigh–Bénard convection, spanning increasingly complex multiphysics interactions from bidirectional linear coupling to nonlinear transport, density-driven flow, and buoyancy–advection feedback under incompressibility. Across these case studies, ChronoSync recovers close to 90% of the conditional to joint oracle error gap on average and is generally the best decoupled data method, substantially approaching models trained directly on coupled trajectories. ChronoSync additionally recovers physically meaningful coupled observables, closely tracking solute transport in Darcy flow and convective heat transport in Rayleigh-Bénard convection. These results demonstrate that long-horizon coupled dynamics can be recovered from decoupled generative models by synchronizing their probability flows across diffusion time and propagating that synchronization coherently across PDE time.

1 INTRODUCTION

Developing surrogate models for multiphysics (i.e., multi-field) coupled partial differential equations (PDEs) over long time horizons is challenging due to the complex spatial and temporal interactions between the physics variables (Zhang et al., 2025; Gao et al., 2026). Localized dynamics in one field can influence not only its own future evolution, but also the evolution of other coupled fields at different spatial locations and later times. Consequently, surrogates need to resolve two kinds of dependencies simultaneously: cross-field and long-horizon temporal evolution (Lippe et al., 2023). While deterministic and generative surrogates have demonstrated strong performance for long-horizon prediction of single-physics systems (Brandstetter et al., 2022; Lippe et al., 2023; Rühling Cachay et al., 2026; Kohl et al., 2026), coupled multiphysics PDEs remain more challenging because nonlinear interactions can arise through multiple, system-dependent coupling mechanisms (Keyes et al., 2013).

Generally, learning coupled dynamics assumes access to training trajectories from the fully coupled system, which can be computationally expensive. Whereas, in many settings, the available information may instead be fragmented across decoupled or conditional simulations and observations

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.

(Abdul Majid et al., 2026). Recovering a coherent coupled solution from such partial information is interesting beyond computational considerations: it asks whether independently learned physics knowledge can be composed to reconstruct dynamics that were never observed jointly during training (Neary & Topcu, 2023). Moreover, even when coupled trajectories are available, directly learning the full joint dynamics need not be the only effective strategy. Decomposing the problem into specialized conditional models can reduce the complexity of the individual learning tasks, at the cost of introducing an additional approximation when these models are composed.

Composing separately trained conditional models introduces a further challenge. Even if the conditional models can be combined accurately within a local prediction window, their composition generally incurs a splitting error. For example, prior work has shown that composing deterministic surrogates trained on decoupled physics can deteriorate substantially (Zhang et al., 2025). Critically, in a rolling forecast, this error is not isolated. It perturbs the state supplied to subsequent windows and can therefore accumulate over PDE time. Thus, long-horizon composition requires controlling not only local synchronization error, but also its repeated propagation through the rollout.

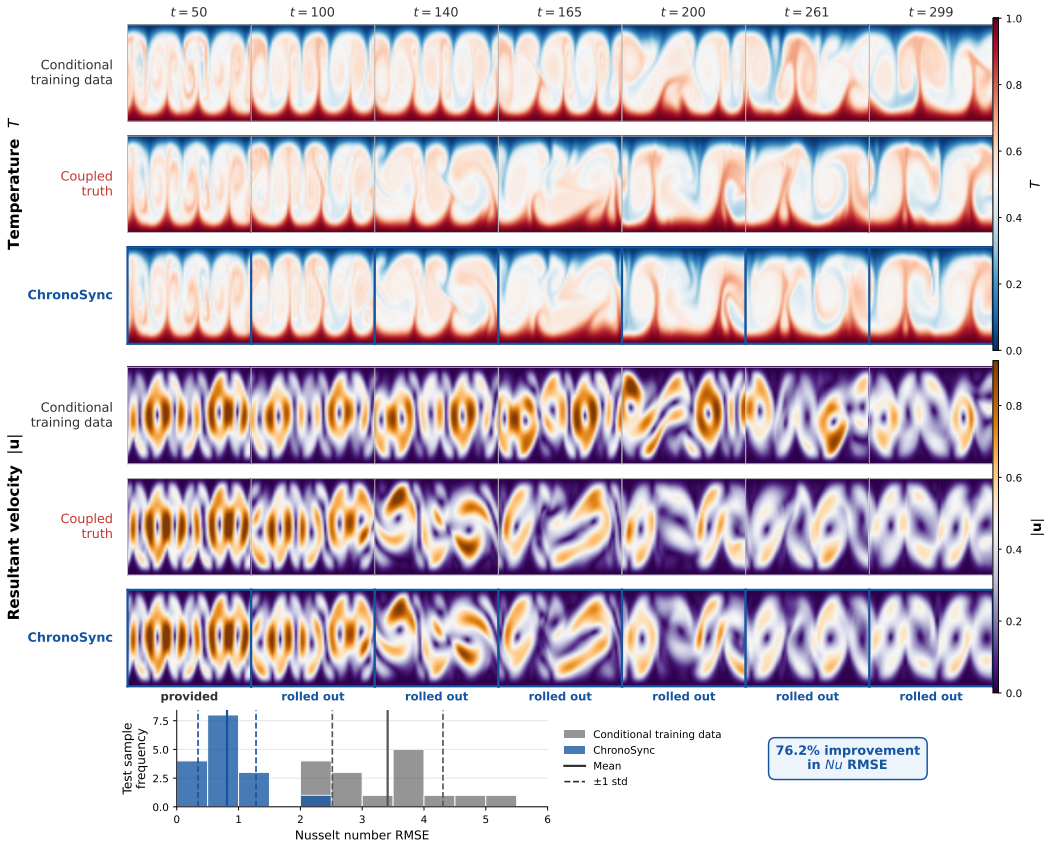


Figure 1: **ChronoSync's autoregressive $T \leftrightarrow (u_x, u_y)$ synchronization tracks coupled Rayleigh–Bénard convection beyond initially aligned conditional training data.** The states upto $t = 54$ are provided, while subsequent states are rolled out autoregressively. For temperature T and resultant velocity $|\mathbf{u}|$, the rows compare conditional training data, coupled reference solution, and ChronoSync composition. ChronoSync closely tracks the T plume dynamics and the $|\mathbf{u}|$ spatial structures over the rollout despite substantial drift between the conditional and coupled trajectories. The bottom histogram compares Nusselt number Nu (a nondimensional measure of convective heat transport) RMSE over 16-sample stress-test subset selected solely by poor conditional performance, without reference to ChronoSync. ChronoSync reduces mean Nu RMSE by 76.2% relative to the conditional training data and exhibits lower variability.

We propose *ChronoSync*, which recovers coupled long-horizon dynamics by synchronizing independently trained conditional Elucidated Rolling Diffusion Models (ERDMs) at inference. ERDMs

generate PDE evolution given a window length and recursively roll these windows forward in physical time (Rühling Cachay et al., 2026). Each conditional ERDM can be viewed as a probability flow ODE and defines a vector field over diffusion time for each PDE field. ChronoSync sequentially composes these probability flow operators within each denoising step and then advances the resulting synchronized state through the next PDE time window. As shown in Figure 1, this procedure can recover dynamics substantially closer to the coupled reference despite never training on coupled trajectories.

Operator splitting provides a natural route to analyzing ChronoSync. The local discrepancy introduced by composing the conditional probability flow ODEs is characterized by their Lie brackets. This local splitting defect enters each rollout as a perturbation. By recursively propagating these perturbations across rolling windows, we obtain a long-horizon error recurrence. Further, we show that under a mixed regime stability condition, the accumulated splitting perturbation error can remain bounded.

We evaluate ChronoSync on four increasingly challenging coupled PDE systems spanning reaction-diffusion (PDEBench Takamoto et al. (2022)), nonlinear transport, Darcy flow (MultiPhysicsBench Yang et al. (2026)), and Rayleigh-Bénard convection (Kontolati et al., 2024). Across these problems, ChronoSync substantially reduces the gap between decoupled conditional trajectories and coupled data reference models, while also recovering physically meaningful coupled observables. As such, the main contributions of our paper include: (1) ChronoSync introduces a coupled rolling inference procedure that synchronizes independently trained conditional probability flow fields within each diffusion step using operator splitting, enabling long-horizon coupled PDE generation from decoupled models; (2) We characterize the local composition error through Lie brackets and derive a recursive perturbation bound describing how splitting error propagates across rolling PDE time windows; and (3) Our empirical evaluations show that ChronoSync substantially approaches models trained directly on coupled PDE data despite its training regime being weaker than the coupled/joint data models.

2 Related Works

Long-horizon forecasting Long-horizon PDE forecasting has been studied extensively using both deterministic autoregressive surrogates and generative models. Deterministic approaches can accurately propagate learned dynamics. However, repeated autoregressive rollout can amplify local prediction errors and induce distribution shift over long rollouts (Lippe et al., 2023; Liao et al., 2026). Generative approaches have consequently explored diffusion-based and flow-based formulations for modeling temporal evolution and uncertainty, including DYffusion (Rühling Cachay et al., 2023), Dynamical Diffusion (Guo et al., 2025), autoregressive conditional diffusion models (Kohl et al., 2026), Recursive Flow Matching (Huang et al., 2026), and autoregressive models (Yang & Xue, 2026). In particular, ERDMs generate fixed temporal windows and recursively advance them through PDE time (Rühling Cachay et al., 2026). This potentially provides a stable mechanism for long-horizon generative prediction over PDE time. Despite these advances, maintaining accurate dynamics under repeated rollout remains challenging, as these surrogates condition future predictions on their own outputs (Brandstetter et al., 2022).

Compositional modeling A complementary direction decomposes the complex problem of learning coupled PDE systems into reusable components that are composed at inference time. Neural operator approaches such as Origo (Sun et al., 2026) and Learnable Composition (Chen et al., 2026) use operator or spatial decomposition to construct coupled physical models. More closely related to this study is M2PDE (Zhang et al., 2025) which learns conditional denoising diffusion probabilistic models (Ho et al., 2020) from decoupled physics and iteratively composes them to recover coupled solutions. More recently, GenCP (Gao et al., 2026) formulates the problem of conditional training to joint/coupled generation through the composition of flow matching models (Lipman et al., 2023). However, when such composition is repeatedly embedded within a long-horizon rolling forecast, local composition errors become perturbations of the state supplied to subsequent windows. This creates an additional error propagation mechanism that existing compositional frameworks do not explicitly address. In contrast, ChronoSync explicitly addresses this and synchronizes conditional models via local composition over both diffusion time and overlapping PDE time rollouts.

3 METHODS

3.1 PRELIMINARIES

Setup and notation Consider a time-dependent system of PDEs (see Appendix E for canonical examples) which has been discretized in space using a finite difference or finite element approach, leading to a semi-discrete representation of the PDE solution. Let $y(t) \in \mathcal{Y}$ denote the *coupled* PDE state, with field components $y(t) = \left(y^1(t), \ldots, y^K(t)\right)$ evaluated at time t on a structured spatial grid. We consider discrete times $t_n = n\,\Delta t$ and write $y_n^k := y^k(t_n)$ for field component k. $y_n^k | y_n^{\neg k}\ \forall\ k \in \{1 \ldots K\}$ denotes the field k at time t_n with all the other fields $y_n^{\neg k} := \{y_n^j\}_{j \neq k}$ frozen at t_n. These frozen fields are the conditional fields. $y_{n+1:n+W}^k | y_{n+1:n+W}^{\neg k}$ denotes a window of time frames of length W for field y^k given the conditional fields $y^{\neg k}$ for the same window. We write $\mathbf{y}_n^k := y_{n+1:n+W}^k$ and $\mathbf{y}_n^{\neg k} := y_{n+1:n+W}^{\neg k}$ for a window of W frames.

Problem formulation We will consider a window of W frames autoregressively and consider one field at a time by conditioning on the other fields: $p_{\theta^k}^W(\mathbf{y}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s}) := p_{\theta^k}(y_{n+1:n+W}^k | y_{n+1:n+W}^{\neg k}, \mathbf{s})$. Additionally, the conditional models $p_{\theta^k}^W(\mathbf{y}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s})\ \forall\ k \in \{1, \ldots, K\}$ need not be trained on the joint PDE data; they can be trained on decoupled PDE data obtained by solving for one PDE field at a time while fixing the other PDE fields. The goal is to approximate $p(y_{1:T} | \mathbf{s})$ by composing the conditional diffusion models $p_{\theta^k}^W(\mathbf{y}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s})$ autoregressively along the PDE time.

3.2 ELUCIDATED ROLLING DIFFUSION MODELS

Training Conditional ERDMs model $p_{\theta^k}^W(\mathbf{y}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s})$ by learning a denoiser network $\mathcal{D}_{\theta^k}^W(\bar{\mathbf{x}}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s}; \bar{\boldsymbol{\sigma}}^W(\tau))$, where τ represent the diffusion time, $\bar{\boldsymbol{\sigma}}^W(\tau) := (\bar{\sigma}_1(\tau), \ldots, \bar{\sigma}_W(\tau))$, and $\bar{\mathbf{x}}_n^k \sim \mathcal{N}(\mathbf{y}_n^k, \mathrm{diag}(\bar{\sigma}_1^2(\tau)\mathbf{I}_d, \ldots, \bar{\sigma}_W^2(\tau)\mathbf{I}_d))$ (Rühling Cachay et al., 2026). Here, d denotes the number of spatial degrees of freedom in a single frame of field k, and $\mathbf{I}_d$ is the corresponding identity matrix. Furthermore, $\bar{\sigma}_w(\tau) = (\sigma_{\max}^{1/\rho} + \tau_w(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho}))^\rho$, where $\tau_w = 1 - \frac{w-\tau}{W}$. The denoiser network comprises a raw neural network F_θ and preconditioning to account for frame-dependent noise levels (see Karras et al. (2022) for more details). The denoiser is trained to predict the clean window $\mathbf{y}_n^k$ from the noisy version $\bar{\mathbf{x}}_n^k$ given the conditioning other fields $\mathbf{y}_n^{\neg k}$ and $\mathbf{s}$. During training, $\tau \sim \mathbb{U}[0, 1)$ is randomized so the model sees the full diffusion time horizon.

Inference ERDM training teaches the model to predict the window $\mathbf{y}_n^k$ given the noisy window $\bar{\mathbf{x}}_n^k$ with the weighting factors skewing the model capacity towards frames with lower noise. Inference denoises a state under a decreasing noise schedule via the score-SDE formalism by Song et al. (2021). Extending this to a window W of frames with increasing noise levels is possible so long as the covariance $\boldsymbol{\Sigma}$ is block-diagonal (Karras et al., 2022). The rate of change of $\bar{\mathbf{x}}_n^k$ with the noise levels is $\frac{d\bar{\mathbf{x}}_n^k}{d\tau} = \frac{d\bar{\boldsymbol{\sigma}}^W}{d\tau} \odot \left[(\bar{\mathbf{x}}_n^k - \hat{\mathbf{y}}_n^k) \oslash \bar{\boldsymbol{\sigma}}^W\right]$, $\hat{\mathbf{y}}_n^k := \mathcal{D}_{\theta^k}^W(\bar{\mathbf{x}}_n^k | \mathbf{c}_n^k; \bar{\boldsymbol{\sigma}}^W)$, where $\mathbf{c}_n^k$ is the conditioning data while $\odot$ and $\oslash$ respectively denote component-wise product and division. This formulation provides a means to apply an Euler scheme to denoise the frame. A 2nd order Heun corrector is further applied to improve this frame's denoising accuracy. Appendix A provides the full details.

3.3 OPERATOR SPLITTING IN ELUCIDATED ROLLING DIFFUSION MODELS

Scope of the analysis We analyze the error from sequentially composing learned conditional probability-flow fields relative to the unsplit flow induced by their sum. We do not assume that learned conditionals are exactly compatible with a common joint distribution, nor that their summed flow equals the unavailable ideal coupled-data flow. Discrepancies from the ideal flow are treated separately; Appendix C.6 incorporates bounded model perturbations into the rollout recurrence, while the overall practical consequences are evaluated empirically.

Commutativity Consider a PDE with two coupled physics states $y^1(t)$ and $y^2(t)$. We assume the states to be scalar-valued variables, but the discussion can be easily extended multiple vector-valued states. Assuming the PDE time is discretized using N_t time points, we define the coupled state vector $\mathbf{x} = \left(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_t)}\right)$ where $\mathbf{x}^{(n)} = \left(\mathbf{x}^{1,(n)}, \mathbf{x}^{2,(n)}\right)$ is the noisy estimate of (y_n^1, y_n^2). The

probability flow ODE for inference can be generally written as

$$\dot{\mathbf{x}}(\tau) = \mathcal{V}(\mathbf{x}, \tau) = \frac{1}{2}\dot{C}(\tau)C(\tau)^{-1}\Big(\mathbf{x} - \mathcal{D}(\mathbf{x}; C(\tau))\Big) \tag{1}$$

where, $C(\tau)$ is the covariance depending on diffusion time τ, while $\mathcal{D}(.)$ is the denoiser (the conditioning information is suppressed for brevity). The flow field $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2)$ can be split as $\mathcal{V}(\mathbf{x}, \tau) = \mathcal{F}(\mathbf{x}, \tau) + \mathcal{G}(\mathbf{x}, \tau)$ where $\mathcal{F} = \big(\mathcal{V}_1, 0\big)$ and $\mathcal{G} = \big(0, \mathcal{V}_2\big)$ are the flow fields for variables $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively. We can introduce an additional level of splitting based on the PDE time frames as $\mathcal{F} = \sum_{n=1}^{N_t} \mathcal{F}^{(n)},\ \mathcal{G} = \sum_{n=1}^{N_t} \mathcal{G}^{(n)}$, where $\mathcal{F}^{(n)}$ and $\mathcal{G}^{(m)}$ have block supports only on $\mathbf{x}^{1,(n)}$ and $\mathbf{x}^{2,(m)}$, respectively. Note that the flow field in equation 1 is exact in that it operates on the coupled physics variables jointly. We define a flow map associated with the exact flow field in equation 1, which we remark is a non-autonomous ODE. Thus, the flow map is not diffusion time translation invariant, and needs to be explicitly represented both the start and end times (Blanes et al., 2024). We subdivide the diffusion time interval $[0, 1]$ into S sub-intervals with the ordering $\tau_S < \cdots < \tau_0$ and set $h_k := |\tau_{k+1} - \tau_k|$. Denoting $\mathbf{x}_k = \mathbf{x}(\tau_k)$, the one step flow map is $\mathbf{x}_{k+1} \equiv \Phi_{\mathcal{F}+\mathcal{G}}^{\tau_{k+1},\tau_k}(\mathbf{x}_k)$.

Now consider the probability flow ODE where the flow field is $\mathcal{F}(\mathbf{x}, \tau)$. This corresponds to training a denoiser for $\mathbf{x}^1$ conditioned on $\mathbf{x}^2$. Similary, the ODE with flow field $\mathcal{G}(\mathbf{x}, \tau)$ corresponds to training a denoiser for $\mathbf{x}^2$ conditioned on $\mathbf{x}^1$. This leads to a decoupled physics solve for the two variables. In this setting, the one step coupled flow map a Lie-Trotter-type splitting scheme is defined as $\mathbf{x}_{k+1}^{\mathrm{LT}} \equiv \Phi_{\mathcal{F}}^{\tau_{k+1},\tau_k} \circ \Phi_{\mathcal{G}}^{\tau_{k+1},\tau_k}(\mathbf{x}_k)$. If the one step flow maps commute, then $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^{\mathrm{LT}}$. In general, non-commutativity results in a splitting error of the form (see Appendix C.1.2 for details)

$$\left\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{LT}\right\| = \left\|\frac{h_k^2}{2}[\mathcal{F}, \mathcal{G}](\mathbf{x}_k, \tau_k) + \mathcal{O}(h_k^3)\right\| \tag{2}$$

under sufficient regularity of the flow fields. In equation 2 the non-commutativity is characterized by the Lie bracket $[\mathcal{F}, \mathcal{G}](\mathbf{x}_k, \tau_k)$, with the higher order error terms also controlled by it.

Splitting perturbation in a single window ERDMs maintain a rolling window of W frames which advances along the PDE time axis. During rollout r, we denote the coupled variable after k denoising steps within the Heun loop as $\mathbf{x}_{r,k} = \Big(\mathbf{x}_{r,k}^{(1)}, ..., \mathbf{x}_{r,k}^{(W)}\Big)$. At the end of the S denoising steps, ERDM commits the first frame after a rollout, while retaining the remaining partially denoised time frames for the next rollout after shifting by $n_{\mathrm{cln}} = 1$ frames. The state for the next rollout is initialized as $\mathbf{x}_{r+1,0} = \Big(\mathbf{x}_{r,S}^{(2)}, ..., \mathbf{x}_{r,S}^{(W)}, \mathbf{x}_{r+1,0}^{(W)}\Big)$, where $\mathbf{x}_{r+1,0}^{(W)}$ is a newly initialized noise frame. Introducing the shorthand notation $[\mathcal{F}, \mathcal{G}]_{r,k} := [\mathcal{F}, \mathcal{G}]\,(\mathbf{x}_{r,k}, \tau_k)\ ,\ [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]_{r,k} := [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]\,(\mathbf{x}_{r,k}, \tau_k)$, the accumulated Lie-Trotter perturbation generated over W frames at rollout r is expressed as

$$\eta_r = \sum_{k=1}^{S} \frac{h_k^2}{2} \sum_{n=1}^{W} [\mathcal{F}^{(n)}, \mathcal{G}^{(n)}]_{r,k} + \sum_{k=1}^{S} \frac{h_k^2}{2} \sum_{\substack{n,m=1 \\ n \neq m}}^{W} [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]_{r,k} + \mathcal{O}(h^2) = \mathrm{I} + \mathrm{II} + \mathcal{O}(\mathrm{h}^2) \tag{3}$$

where $h = \max_k h_k$. In equation 3, I captures non-commutativity between the two physics flows at the same PDE time, while II captures non-commutativity across different PDE times t_n and t_m.

Accumulated splitting errors with the rolling state After committing the first frame of the window, ERDM shifts the overlapping window forward by one frame. Hence, the perturbation propagated between rollouts must be defined over the entire W-frame rolling state. Let $\delta_r = (\delta_r^{(1)}, \ldots, \delta_r^{(W)})$ denote the perturbation of the rolling state entering rollout r due to splitting errors accumulated during previous rollouts, with δ_r having a form similar to equation 3. For brevity, we write $\mathbf{x}_r := \mathbf{x}_{r,0}$. At rollout r, let Ψ_r denote the differentiable reference (i.e., unsplit) denoising map over a completed Heun loop, with $\Psi_r(\mathbf{x}_r) = \mathbf{x}_{r,S}$. Applying the reference map to the perturbed window and performing a local linearization gives $\Psi_r(X_r + \delta_r) = \Psi_r(X_r) + J_r\delta_r + \mathcal{O}(\|\delta_r\|^2)$, where $J_r \equiv \nabla_{\mathbf{x}}\Psi_r(\mathbf{x})$.

During rollout r, ERDM performs Heun integration and introduces operator splitting perturbation η_r. After rollout r and before shifting, the perturbation is $\tilde{\delta}_r := J_r\delta_r + \eta_r$, where we have ignored higher order terms. The first term linearly approximates the inherited accumulated perturbation from

the previous rollout into r. The second term represents new perturbation from operator splitting. In the inherited perturbation term $J_r\delta_r$, all components of δ_r contribute. However, after shifting the window, the first component of the perturbation which represents the committed frame from the previous rollout does not propagate to the next rollout. To formalize this, we define a window (linear) shift operator $\mathcal{S}_W(z^{(1)}, z^{(2)}, \ldots, z^{(W)}) \equiv (z^{(2)}, \ldots, z^{(W)}, 0)$. Therefore, the perturbation entering the new rollout is $\delta_{r+1} = \mathcal{S}_W(\tilde{\delta}_r)$. We define the sensitivity coefficient $L_r \equiv \|\mathcal{S}_W J_r\|$ measuring the local sensitivity of the shifted rolling state to perturbations accumulated from the previous rollout. Then the magnitude of the new rollout can be estimated as

$$\|\delta_{r+1}\| \leq L_r\|\delta_r\| + \|\mathcal{S}_W\eta_r\|. \tag{4}$$

Considering equation 4 and under standard regularity assumptions stated precisely in Appendix C, we can show the following rollout error bound resulting from operator splitting in ERDMs.

Theorem 1 (Informal error bound)**.** *Given the one rollout error bound in equation 4, the perturbation error after R rollouts is bounded by the propagated initial full window splitting perturbation error plus the sum of all splitting perturbations weighted by the sensitivities of subsequent rollouts,*

$$\|\delta_{R+1}\| \leq \left(\prod_{q=1}^{R} L_q\right)\|\delta_1\| + \sum_{r=1}^{R}\left(\prod_{q=r+1}^{R} L_q\right)\|\mathcal{S}_W\eta_r\| \tag{5}$$

If the first window is exactly provided, then $\|\delta_1\| \approx 0$. Furthermore, the denoiser error in modeling the decoupled physics variables y^1 and y^2 introduce additional perturbations ζ_r over η_r during the rollout. A detailed derivation and discussion on Theorem 1 is provided in Appendix C.

Using equation 5, we can perform a regime analysis of the rollout splitting error growth based on whether the sensitivity coefficient is non-expansive ($L_r < 1 \;\forall\, r$), neutral ($L_r \approx 1 \;\forall\, r$), or expansive ($L_r > 1 \;\forall\, r$). For practical coupled PDEs, we consider the following mixed regime.

Corollary 1 (Mixed regime)**.** *If $\prod_{q=r+1}^{R} L_q \leq C_L\rho^{R-r}$ for some $0 < \rho < 1$ and a constant C_L, and the per-rollout splitting error is bounded $\|\mathcal{S}_W\eta_r\| \leq \epsilon$, then*

$$\lim_{R\to\infty} \sup \|\delta_R\| \leq C_L\epsilon/(1-\rho) \tag{6}$$

Under the regularity assumptions of Appendix C, bounded aggregate Lie brackets and higher-order remainders imply $\|\mathcal{S}_W\eta_r\| \leq \epsilon$, providing a uniform bound on the splitting perturbation injected by each rollout. Note that the mixed regime condition does not require every rollout to be locally non-expansive. Individual rollouts may have $L_r > 1$, provided that their cumulative sensitivity products decay geometrically. Thus, $\prod_{q=r+1}^{R} L_q \leq C_L\rho^{R-r}$ with $0 < \rho < 1$ is a sufficient condition analogous to uniform exponential stability in non-autonomous dynamical systems (Potzsche, 2010).

3.4 ChronoSync Algorithm

From Section 3.3, the splitting scheme advances the coupled state flow map via the composition $\mathbf{x}_{i+1}^{\mathrm{LT}} = \Phi_{\mathcal{F}}^{\tau_{i+1},\tau_i} \circ \Phi_{\mathcal{G}}^{\tau_{i+1},\tau_i}(\mathbf{x}_i)$ with i denoting the discrete diffusion time index. Restricting the noisy states to a window W of PDE time frames and maintaining a rolling window where the first frame is committed after each rollout, as in ERDMs, ChronoSync composes decoupled flow fields to recover the coupled state $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2)$. In the context of Theorem 1 and the discussion leading up to it, provided that the per-rollout perturbation error is bounded and that expansive rollouts ($L_r > 1$) are sufficiently offset by non-expansive ones ($L_r < 1$), the PDE time long-horizon composition error due to operator splitting stays bounded (Corollary 1).

Mapping this discussion to the conditional models introduced in Section 3.1, $p_{\theta^k}^W(\mathbf{y}_n^k \mid \mathbf{y}_n^{\neg k}, \mathbf{s})$ for $k \in \{1, \ldots, K\}$, we construct the coupled probability flow by sequentially evaluating the conditional field models within each Heun step. At diffusion step i, each conditional model denoises its noisy window $\bar{\mathbf{x}}_n^k$ using the current cross-field estimates $\tilde{\mathbf{y}}_n^{\neg k}$ as conditioning. After completing the predictor-corrector update for field k, its denoised estimate is immediately made available to the subsequent field models in the ordered sweep $k = 1, \ldots, K$. With $\kappa = 1$, this recovers the sequential Lie-Trotter composition, while $0 < \kappa < 1$ under-relaxes the cross-field update according to $\tilde{\mathbf{y}}_n^k \leftarrow \kappa\,\mathbf{y}_n^{k,\mathrm{new}} + (1-\kappa)\tilde{\mathbf{y}}_n^{k,\mathrm{old}}$.

Algorithm 1 summarizes the resulting ChronoSync Heun loop. Across PDE time, Algorithm 1 is applied for R rolling window rollouts. After completing rollout r, the window is shifted forward by one PDE time frame and its first newly generated frame is committed to the coupled trajectory. When entering rollout $r+1$, the cross-field conditioning window therefore contains $W-1$ available frames from the current coupled trajectory. ChronoSync pads the missing terminal frame independently for each field k by repeating its last valid frame. The target windows are shifted in the same manner, except that their newly exposed terminal frame is initialized at the corresponding ERDM noise level and subsequently denoised through the coupled Heun loop. Consequently, synchronization is performed repeatedly both along diffusion time within each rollout and along PDE time as the window advances.

ChronoSync optionally performs $n_{\text{boot}}^{\text{roll}}$ additional bootstrap refinements before committing the newly generated frame of each rollout. Each bootstrap therefore re-solves the same PDE time window using the most recently synchronized coupled estimate as its prescribed state. After $n_{\text{boot}}^{\text{roll}}$ such refinements, the resulting window is written back to the rolling trajectory and the algorithm proceeds to rollout $r+1$.

Algorithm 1 ChronoSync Heun loop (descriptive)

Require: noisy windows $\{\bar{\mathbf{x}}_n^k\}_{k=1}^K$, cross-field estimates $\{\tilde{\mathbf{y}}_n^k\}_{k=1}^K$, static conditioning $\mathbf{s}$, $n_{\text{cln}}=1$, steps S, relaxation $\kappa\in[0,1]$

$\Delta\tau \leftarrow 1/S$
for $i=0,\dots,S-1$ **do**
  $\tau_i \leftarrow i/S, \quad \tau_{i+1}\leftarrow (i+1)/S$
  $\boldsymbol{\sigma}_i \leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_i), \quad \boldsymbol{\sigma}_{i+1}\leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
  $\dot{\boldsymbol{\sigma}}_i \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_i)$
  $\tilde{\mathbf{y}}_n^{k,\text{old}} \leftarrow \tilde{\mathbf{y}}_n^k \quad \forall k$
  for $k=1,\dots,K$ **do**
    $\mathbf{c}_n^k \leftarrow (\tilde{\mathbf{y}}_n^{\neg k}, \mathbf{s})$
    $\hat{\mathbf{y}}^k \leftarrow \mathcal{D}_{\theta^k}^W\left(\bar{\mathbf{x}}_n^k \mid \mathbf{c}_n^k; \boldsymbol{\sigma}_i\right)$
    $\mathbf{d}^k \leftarrow \dot{\boldsymbol{\sigma}}_i \odot (\bar{\mathbf{x}}_n^k - \hat{\mathbf{y}}^k)/\boldsymbol{\sigma}_i$
    $\bar{\mathbf{z}}^k \leftarrow \bar{\mathbf{x}}_n^k + \Delta\tau\, \mathbf{d}^k$
    if $\tau_{i+1}<1$ **then**
      $\dot{\boldsymbol{\sigma}}_{i+1} \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
      $\hat{\mathbf{y}}'^k \leftarrow \mathcal{D}_{\theta^k}^W\left(\bar{\mathbf{z}}^k \mid \mathbf{c}_n^k; \boldsymbol{\sigma}_{i+1}\right)$
      $\mathbf{d}'^k \leftarrow \dot{\boldsymbol{\sigma}}_{i+1} \odot (\bar{\mathbf{z}}^k - \hat{\mathbf{y}}'^k)/\boldsymbol{\sigma}_{i+1}$
      $\bar{\mathbf{x}}_n^k \leftarrow \bar{\mathbf{x}}_n^k + \frac{\Delta\tau}{2}(\mathbf{d}^k + \mathbf{d}'^k)$
      $\mathbf{y}^{k,\text{new}} \leftarrow \hat{\mathbf{y}}'^k$
    else
      $\bar{\mathbf{x}}_n^k \leftarrow \bar{\mathbf{z}}^k$
      $\mathbf{y}^{k,\text{new}} \leftarrow \hat{\mathbf{y}}^k$
    end if
    $\tilde{\mathbf{y}}_n^k \leftarrow \kappa\, \mathbf{y}^{k,\text{new}} + (1-\kappa)\tilde{\mathbf{y}}_n^{k,\text{old}}$
  end for
  $[\bar{\mathbf{x}}_n^k]_{1:n_{\text{cln}}} \leftarrow [\mathbf{y}_n^k]_{1:n_{\text{cln}}} \quad \forall k$
end for
return $\{\bar{\mathbf{x}}_n^k\}_{k=1}^K, \{\tilde{\mathbf{y}}_n^k\}_{k=1}^K$

4 Experiments

4.1 Case Studies and Baselines

We evaluate ChronoSync on 4 coupled PDE systems with increasing complexity of the interactions between physical fields: FitzHugh-Nagumo (PDEBench Takamoto et al. (2022)), viscous Burgers, Darcy flow (MultiPhysicsBench Yang et al. (2026)), and Rayleigh-Bénard convection (Kontolati et al., 2024). The coupled variables of interest are $u \leftrightarrow v$ for FitzHugh-Nagumo, $v \rightarrow c$ for Burgers, $c \leftrightarrow q$ for Darcy and $T \leftrightarrow (u_x, u_y)$ for Rayleigh-Bénard. The corresponding spatio-temporal resolutions are $128X \times 101T$, $256X \times 512T$, $64X \times 64Y \times 300T$, and $64X \times 64Y \times 300T$. We evaluate ChronoSync against the original M2PDE (Zhang et al., 2025) and an autoregressive extension of it, M2PDE-AR, to handle long time horizons. We also evaluate against the GenCP (Gao et al., 2026) extension GenCP-AR. These form our decoupled baselines since these models are trained on decoupled physics (i.e., conditional) data. For generating the conditional training data for Darcy and Rayleigh-Bénard systems, we adopt a windowed pre-iteration strategy similar to that adopted in Zhang et al. (2025); Gao et al. (2026). For reference, we additionally train ERDM-Joint (Rühling Cachay et al., 2026) and FNO-AR-Joint (Brandstetter et al., 2022) directly on coupled trajectories as coupled data oracles. Across all methods, we use broadly comparable model complexity and representational power for the backbones. ChronoSync uses $\kappa=0.25$ and $n_{\text{boot}}^{\text{roll}}=1$.

4.2 Scalar Error Metrics

Table 1 presents the long-horizon relative (ℓ_2) errors. All compositional methods perform well on the relatively simple FitzHugh-Nagumo system, whereas their differences become pronounced as coupling complexity increases. Across Burgers, Darcy transport, and Rayleigh-Bénard convection,

ChronoSync reduces error by 53%-74% relative to the strongest decoupled compositional baseline for each field. M2PDE-AR diverges for Darcy transport, while applying the original M2PDE to the complete 300 frames trajectory exceeds the available GPU memory. Overall, ChronoSync is the strongest method trained exclusively on decoupled trajectories.

Table 1: **Long-horizon rollout accuracy measured by relative ℓ_2 error (lower is better).** ERDM-Joint and FNO-AR-Joint are trained directly on coupled trajectories with the same amount of data as the decoupled models and serve as oracles. Bold indicates the best method trained without coupled trajectories and italic indicates the second best. All decoupled models and the coupled data oracles use the same window length and the same amount of training data for a given case study.

		Increasing complexity $\longrightarrow$						
		FitzHugh-Nagumo		Burgers	Darcy transport		Rayleigh-Bénard	
Training data	Method	u	v	c	c	q	T	(u_x, u_y)
Decoupled	Conditional trajectories	0.745	0.922	0.504	0.165	0.358	0.219	0.977
	M2PDE	0.013	0.01	0.131	OOM†	OOM†	OOM†	OOM†
	M2PDE-AR*	**0.007**	*0.006*	0.1	–	–	0.189	0.758
	GenCP-AR*	0.027	0.055	*0.067*	*0.159*	*0.141*	*0.162*	*0.663*
	ChronoSync (Ours)	*0.009*	**0.005**	**0.025**	**0.075**	**0.036**	**0.074**	**0.217**
Coupled	ERDM-Joint	0.007	0.007	0.008	0.104	0.122	0.127	0.45
	FNO-AR-Joint	0.003	0.003	0.008	0.038	0.018	0.06	0.138

$*$: Autoregressive extensions of M2PDE and GenCP to handle long time horizons. †: OOM when processing the full 300 time frames jointly.

Considering the coupled data oracles, we define the conditional to oracle gap recovered as the percentage ratio of the difference between conditional and decoupled method errors to the difference between conditional and FNO-AR-Joint errors. For the different case studies, ChronoSync recovers conditional to oracle gaps of 99.1% and 99.7% for FitzHugh-Nagumo, 96.5% for Burgers, 70.8% and 94.7% for Darcy, and 91.9% and 90.5% for Rayleigh-Bénard convection. Thus, ChronoSync, on an average, recovers about 90% of the conditional to joint data oracle gap across all the case studies. Even relative to a hypothetical zero error oracle, ChronoSync recovers about 83% of the conditional to oracle gap on average. We provide ablation studies in Appendix J.

4.3 Time Rollout Errors

We evaluate relative ℓ_2 error as a function of PDE time for the more complex 2D systems: Darcy and Rayleigh-Bénard. Figure 2 presents the time rollout relative ℓ_2 error. The conditional training data remain substantially separated from the coupled reference throughout the PDE time steps. ChronoSync continuously synchronizes the decoupled field dynamics toward the coupled trajectory, maintaining substantially lower errors over long PDE time horizons. The effect is particularly pronounced for Darcy transport, where the q exhibits large errors throughout the rollout and ChronoSync maintains low error for both q and c. In Rayleigh-Bénard, the conditional $\mathbf{u}$ and T trajectories progressively diverge from the coupled reference, while ChronoSync substantially suppresses this growth and remains closer to the coupled data oracles.

4.4 Physical Consistency

For Darcy, we measure the vertical center of mass of the solute concentration $y_c = \int_\omega ycd\omega / \int_\omega cd\omega$, where ω is the spatial domain, against the coupled reference data. y_c tracks the bulk vertical migration of the solute and integrates the effect of the coupled pressure and concentration dynamics. Figure 3(a) shows that the conditional training data and GenCP-AR develop persistent deviation of the y_c relative to the coupled data. ChronoSync strongly suppresses these deviations after the initial transition and maintains a small deviation over the rest of the rollout, approaching the coupled data oracles. For Rayleigh-Bénard, we evaluate the Nusselt number which measures convective heat transport and report its RMSE compared to the coupled reference data. Figure 3(b) shows that

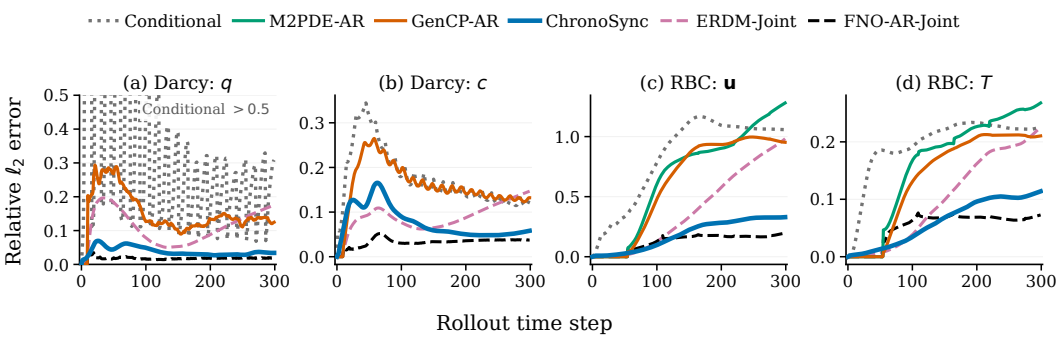


Figure 2: **Long-horizon rollout errors for Darcy and Rayleigh–Bénard systems** as a function of PDE time for (a) Darcy pressure anomaly q, (b) Darcy concentration c, (c) Rayleigh–Bénard velocity $\mathbf{u}$, and (d) Rayleigh–Bénard temperature T.

ChronoSync substantially reduces the Nusselt number RMSE compared to its conditional training data and also GenCP-AR and produces a distribution that is comparable to the coupled data oracles.

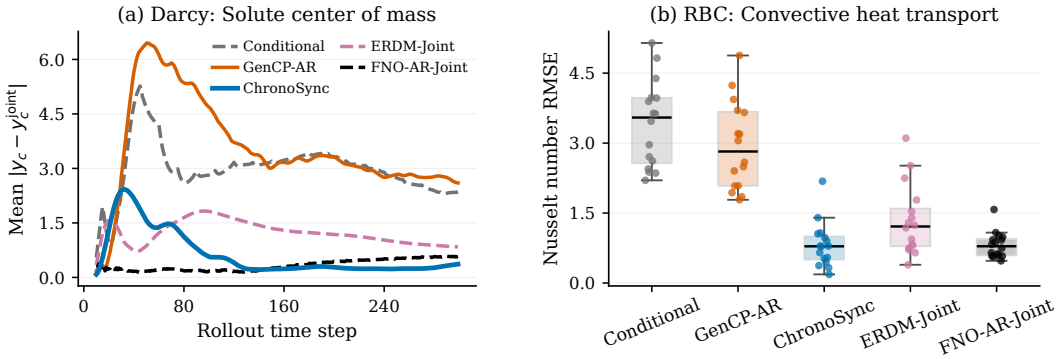


Figure 3: **Physical consistency of long-horizon predictions.** (a) Mean absolute error in the vertical solute center of mass trajectory $y_c = \int_\omega y c d\omega / \int_\omega c d\omega$ relative to the coupled Darcy reference solution. (b) Nusselt-number RMSE for Rayleigh–Bénard, measuring error in convective heat transport.

4.5 OOD Testing: Time Extrapolation

We also test whether the recovered coupled dynamics remain coherent outside the PDE time range used for training. For the Rayleigh-Bénard, we extend the inference from the 300 to 399 frames and compare ChronoSync with FNO-AR-Joint, the best coupled data oracle. FNO-AR-Joint obtains temperature and velocity relative ℓ_2 errors of 0.089 and 0.345, respectively, whereas, ChronoSync obtains 0.113 and 0.322. The median Nusselt number RMSE is 0.64 for FNO-AR-Joint and 0.72 for ChronoSync, showing that it remains consistent with the best coupled data oracle.

5 Conclusions and Limitations

ChronoSync is an inference time method for recovering long-horizon coupled PDE dynamics from independently trained conditional field diffusion models. Across four increasingly challenging systems, ChronoSync demonstrates strong performance and recovers about 90% of the conditional to joint oracle performance gap on average. A limitation is inference cost. Each field model is evaluated repeatedly throughout the Heun reverse process, with optional bootstrap refinements, and inference efficiency was not optimized in this work. One-step generative forecasting (Yang & Xue, 2026) and recursive composition of learned flow maps (Huang et al., 2026) offer promising routes to reducing this cost. Future work should also study systems with more coupled fields and settings where the available conditional data are very weakly informative or even mutually incompatible.

AI USE STATEMENT

The research questions, the ChronoSync formulation, the benchmarks setup (M2PDE, M2PDE-AR, GenCP-AR, ERDM-Joint, and FNO-AR-Joint), the case studies setup, the theoretical analysis, and the experimental design in this work were conceived and developed by the authors. Generative AI was used as an assistant as detailed below.

Generative AI was used for feedback on methodology and experiment design, and for implementing methods in research code, in both cases under the direction of the authors. It was also used for discussion of mathematical formulations and conceptual arguments, all of which were derived and verified by the authors. Generative AI was also used for feedback on manuscript organization and wording, and to generate candidate prose at selected points. Generative AI was not used for synthetic data generation, for writing proofs or supplying key proof ingredients, for proposing or refining hypotheses, for dataset cleaning or reformatting, or for interpreting results. Translation and qualitative or thematic analysis support are not applicable to this work.

The authors alone executed all numerical simulations and machine learning experiments, inspected the resulting data, and verified every reported quantitative result. The authors have reviewed all AI-assisted work, testing and rewriting it where necessary, and take full responsibility for the content, claims, code, and results presented in this work.

REPRODUCIBILITY STATEMENT

We provide anonymized source code with the submission, including implementations of ChronoSync and all baselines, and training and inference configurations. Appendix E specifies the PDEs, numerical solvers, and simulation parameters; Appendix G reports model architectures; and Appendix H provides training and inference settings.

REFERENCES

Harris Abdul Majid, Giannis Daras, Francesco Tudisco, and Steven McDonagh. Ambient physics: Training neural pde solvers with partial observations. *arXiv preprint arXiv:2602.13873*, 2026.

Sergio Blanes, Fernando Casas, and Ander Murua. Splitting methods for differential equations. *Acta Numerica*, 33:1–161, 2024.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural pde solvers. In *International Conference on Learning Representations*, 2022.

Keaton J. Burns, Geoffrey M. Vasil, Jeffrey S. Oishi, Daniel Lecoanet, and Benjamin P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2:023068, 2020.

Zituo Chen, Baiming Zhang, and Sili Deng. Learnable composition for neural operator. *arXiv preprint arXiv:2609.03069*, 2026.

Philip G. Drazin and William H. Reid. *Hydrodynamic Stability*. Cambridge University Press, 2nd edition, 2004.

Tianrun Gao, Haoren Zheng, Wenhao Deng, Haodong Feng, Tao Zhang, Ruiqi Feng, Qianyi Chen, and Tailin Wu. Gencp: Towards generative modeling paradigm of coupled physics. In *International Conference on Learning Representations*, 2026.

Xingzhuo Guo, Yu Zhang, Baixu Chen, Haoran Xu, Jianmin Wang, and MingSheng Long. Dynamical diffusion: Learning temporal dynamics with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Logan Harbour, Guillaume Giudicelli, Alexander D. Lindsay, Peter German, Joshua Hansel, Casey Icenhour, Mengnan Li, Jason M. Miller, Roy H. Stogner, Patrick Behne, Daniel Yankura, Zachary M. Prince, Corey DeChant, Daniel Schwen, Benjamin W. Spencer, Mauricio Tano, Namjae Choi, Yaqi Wang, Max Nezdyur, Yinbin Miao, Tianchen Hu, Shikhar Kumar, Christopher Matthews, Brandon Langley, Nuno Nobre, Alexander Blair, Chris MacMackin, Henrique Bergallo Rocha, Edward Palmer, Jesse Carter, Jörg Meier, Andrew E. Slaughter, David Andrš, Robert W. Carlsen, Fande Kong, Derek R. Gaston, and Cody J. Permann. 4.0 MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 31:102264, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Jiahe Huang, Sihan Xu, Sharvaree Vadgama, and Rose Yu. Recursive flow matching. *arXiv preprint arXiv:2605.26535*, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

David E. Keyes, Lois C. McInnes, Carol Woodward, William Gropp, Eric Myra, Michael Pernice, John Bell, and et al. Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27(1):4–83, 2013.

Gene A. Klaasen and William C. Troy. Stationary wave solutions of a system of reaction-diffusion equations derived from the fitzhugh–nagumo equation. *SIAM Journal on Applied Mathematics*, 44(1):96–110, 1984.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *Neural Networks*, 199:108641, 2026.

Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D. Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, 2024.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Junyi Liao, Johann Guilleminot, and Vahid Tarokh. Stable by construction: Variational latent markov operators for long-horizon pde prediction. *arXiv preprint arXiv:2609.16621*, 2026.

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

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67398–67433, 2023.

Cyrus Neary and Ufuk Topcu. Compositional learning of dynamical system models using port-hamiltonian neural networks. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 679–691, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Christian Potzsche. *Geometric Theory of Discrete Nonautonomous Dynamical Systems*. Lecture Notes in Mathematics. Springer, 2010.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators for robust and accurate learning of pdes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200, 2023.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45259–45287, 2023.

Salva Rühling Cachay, Miika Aittala, Karsten Kreis, Noah Brenowitz, Arash Vahdat, Morteza Mardani, and Rose Yu. Elucidated rolling diffusion models for probabilistic forecasting of complex dynamics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 1821–1860, 2026.

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

Li Sun, Hongbo Lv, Zhikai Jiang, Zhongtian Sun, Lanxu Yang, and Philip S. Yu. Origo: Interpretable multi-physics PDE foundation model through neural operator splitting. In *Forty-third International Conference on Machine Learning*, 2026.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 53681–53705, 2024.

Changfan Yang, Lichen Bai, Yinpeng Wang, Shufei Zhang, and Zeke Xie. Multiphysics bench: Benchmarking and investigating scientific machine learning for multiphysics pdes. *arXiv preprint arXiv:2505.17575*, 2026.

Tianyue Yang and Xiao Xue. Autoregressive one-step generative modeling for dynamical system forecasting. *arXiv preprint arXiv:2605.05540*, 2026.

Tao Zhang, Zhenhai Liu, Feipeng Qi, Yongjun Jiao, and Tailin Wu. M2pde: Compositional generative multiphysics and multi-component pde simulation. In *International Conference on Machine Learning*, volume 267, pp. 75638–75666, 2025.

APPENDIX CONTENTS

A Traditional ERDM Training and Inference **16**

B ChronoSync Algorithm **17**

C ChronoSync Error Analysis **19**

C.1 Lie-Trotter operator splitting scheme for nonlinear ODEs 19

C.2 Lie-Trotter operator splitting for probability flows in EDMs 22

C.3 Operator splitting in ERDMs 23

C.4 Recursive rollout error bound 25

C.5 Regime analysis 26

C.6 Discussion 28

D Benchmark Methods: M2PDE-AR (Decoupled Data), GenCP-AR (Decoupled Data), and FNO-AR-Joint (Coupled Data) **29**

D.1 M2PDE-AR: Block autoregressive extension of M2PDE 29

D.2 GenCP-AR: Block autoregressive extension of GenCP 30

D.3 FNO-AR-Joint oracle 32

E PDE Case Studies **33**

E.1 FitzHugh-Nagumo system 33

E.2 Viscous Burgers system 34

E.3 Darcy transport system 34

E.4 Rayleigh-Bénard convection system 35

F Data Generation **37**

G Backbone Architectures **38**

G.1 ChronoSync, M2PDE, M2PDE-AR, and ERDM-Joint 38

G.2 GenCP-AR 39

G.3 FNO-AR-Joint 39

H Training and Inference Settings **40**

H.1 Training settings 40

H.2 Inference settings 40

I Additional Results **40**

I.1 Spatio-temporal plots 40

I.2 Rollout errors over PDE time: FitzHugh-Nagumo and Burgers 41

I.3 Conditional-to-oracle gap recovered 42

I.4 OOD testing for time extrapolation: Rayleigh-Bénard convection 42

J Ablations for ChronoSync **43**

J.1 FitzHugh-Nagumo 44

J.2 Viscous Burgers 44

J.3 Darcy transport 44

J.4 Rayleigh–Bénard convection 45

A TRADITIONAL ERDM TRAINING AND INFERENCE

Conditional ERDMs model the conditional probability density $p^W_{\theta^k}(\mathbf{y}^k_n|\mathbf{y}^{\neg k}_n, \mathbf{s})$ by learning a denoiser network $\mathcal{D}^W_{\theta^k}\left(\bar{\mathbf{x}}^k_n|\mathbf{y}^{\neg k}_n, \mathbf{s}; \bar{\boldsymbol{\sigma}}^W(\tau)\right)$, where $\bar{\boldsymbol{\sigma}}^W(\tau) := (\bar{\sigma}_1(\tau), \ldots, \bar{\sigma}_W(\tau))$ and $\bar{\mathbf{x}}^k_n \sim \mathcal{N}\left(\mathbf{y}^k_n, \mathrm{diag}(\bar{\sigma}^2_1(\tau)\mathbf{I}_d, \ldots, \bar{\sigma}^2_W(\tau)\mathbf{I}_d)\right)$ (Rühling Cachay et al., 2026). Here, d denotes the number of spatial degrees of freedom in a single frame of field k, and $\mathbf{I}_d$ is the corresponding identity matrix. The sigma component $\bar{\sigma}_w(\tau) = (\sigma^{1/\rho}_{\text{max}} + \tau_w(\sigma^{1/\rho}_{\text{min}} - \sigma^{1/\rho}_{\text{max}}))^\rho$, where $\tau_w = 1 - \frac{w-\tau}{W}$. The denoiser network $\mathcal{D}^W_{\theta^k}\left(\bar{\mathbf{x}}^k_n|\mathbf{y}^{\neg k}_n, \mathbf{s}; \bar{\boldsymbol{\sigma}}^W(\tau)\right)$ is comprised of a raw neural network F_{θ^k} and preconditioning to account for frame-dependent noise levels (see Karras et al. (2022) for more details). The denoiser network is trained to predict the clean window $\mathbf{y}^k_n$ from the noisy version $\bar{\mathbf{x}}^k_n$ conditioned on the other fields $\mathbf{y}^{\neg k}_n$ and $\mathbf{s}$. During training, $\tau \sim \mathbb{U}[0, 1)$ is randomized so the model sees the full diffusion time horizon.

ERDM weights the loss function with a frame-dependent $\lambda(\bar{\sigma}_w)$ so that the network sees targets with a unit variance. Moreover, considering that ERDM always covers the full noise range, Rühling Cachay et al. (2026) propose another weighting factor $f(\bar{\sigma}_w; P_{\text{mean}}, P_{\text{std}})$ derived from the log-normal noise level weighting used in EDM so that the training up-weights the most informative frames; P_{mean} and P_{std} are the mean and standard deviation of this distribution. Let $\mathbf{c}^k_n := (\mathbf{y}^{\neg k}_n, \mathbf{s})$ denote the conditioning fields, and $\omega(\bar{\sigma}_w(\tau)) := \lambda(\bar{\sigma}_w(\tau))\, f(\bar{\sigma}_w(\tau); P_{\text{mean}}, P_{\text{std}})$ the combined per-frame loss weight. Thus, the training loss function is:

$$\min_{\theta^k} \mathbb{E}_{(\mathbf{y}^k_n, \mathbf{c}^k_n) \sim p_{\text{data}},\, \tau \sim \mathbb{U}[0,1),\, \bar{\mathbf{x}}^k_n | \mathbf{y}^k_n, \tau} \frac{1}{W} \sum_{w=1}^{W} \omega\big(\bar{\sigma}_w(\tau)\big) \left\| \left[\mathcal{D}^W_{\theta^k}(\bar{\mathbf{x}}^k_n | \mathbf{c}^k_n; \bar{\boldsymbol{\sigma}}^W(\tau))\right]_w - \left[\mathbf{y}^k_n\right]_w \right\|^2_2 \tag{7}$$

where $[\cdot]_w$ denotes the w-th frame slice.

The frame-wise noise schedule enables rolling inference because $\bar{\sigma}_{w+1}(1) = \bar{\sigma}_w(0)$. Consequently, after the first frame becomes approximately clean and is committed, the remaining $W - 1$ frames can be shifted forward while preserving their appropriate noise levels, and only the newly appended final frame must be initialized from noise. Algorithm 2 summarizes the conventional inference for a single trained conditional ERDM.

Algorithm 2 ERDM inference with Heun predictor-corrector scheme (deterministic Heun with $S_{\text{churn}} = 0$)

Require: initial noisy window $\bar{\mathbf{x}}_0^k$, trained denoiser $\mathcal{D}_{\theta_k}^W$, noise schedule $\bar{\boldsymbol{\sigma}}^W(\cdot)$, conditioning trajectory $\mathbf{y}^{\neg k}$, static conditioning $\mathbf{s}$, window W, $n_{\text{cln}} = 1$, denoising steps S, rollouts R

$\Delta\tau \leftarrow 1/S, \quad \widehat{\mathbf{y}}^k \leftarrow \emptyset$
for $r = 0, \dots, R-1$ **do**
  $n_r \leftarrow r, \quad \mathbf{c}_r^k \leftarrow (\mathbf{y}^{\neg k}_{n_r+1:n_r+W}, \mathbf{s})$
  for $i = 0, \dots, S-1$ **do**
    $\tau_i \leftarrow i/S, \quad \tau_{i+1} \leftarrow (i+1)/S$
    $\boldsymbol{\sigma}_i \leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_i), \quad \boldsymbol{\sigma}_{i+1} \leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
    $\dot{\boldsymbol{\sigma}}_i \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_i)$
    $\hat{\mathbf{y}} \leftarrow \mathcal{D}_{\theta_k}^W(\bar{\mathbf{x}}_r^k \mid \mathbf{c}_r^k; \boldsymbol{\sigma}_i)$
    $\mathbf{d} \leftarrow \dot{\boldsymbol{\sigma}}_i \odot (\bar{\mathbf{x}}_r^k - \hat{\mathbf{y}})/\boldsymbol{\sigma}_i$
    $\bar{\mathbf{z}} \leftarrow \bar{\mathbf{x}}_r^k + \Delta\tau\, \mathbf{d}$
    if $\tau_{i+1} < 1$ **then**
      $\hat{\mathbf{y}}' \leftarrow \mathcal{D}_{\theta_k}^W(\bar{\mathbf{z}} \mid \mathbf{c}_r^k; \boldsymbol{\sigma}_{i+1})$
      $\dot{\boldsymbol{\sigma}}_{i+1} \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
      $\mathbf{d}' \leftarrow \dot{\boldsymbol{\sigma}}_{i+1} \odot (\bar{\mathbf{z}} - \hat{\mathbf{y}}')/\boldsymbol{\sigma}_{i+1}$
      $\bar{\mathbf{x}}_r^k \leftarrow \bar{\mathbf{x}}_r^k + \frac{\Delta\tau}{2}(\mathbf{d} + \mathbf{d}')$
    else
      $\bar{\mathbf{x}}_r^k \leftarrow \bar{\mathbf{z}}$
    end if
  end for
  $\hat{\mathbf{y}}_{n_r+1}^k \leftarrow [\bar{\mathbf{x}}_r^k]_1$ ▷ commit first frame
  Append $\hat{\mathbf{y}}_{n_r+1}^k$ to $\widehat{\mathbf{y}}^k$
  if $r < R-1$ **then**
    $\boldsymbol{\xi}_r \sim \mathcal{N}(0, \bar{\sigma}_W^2(0)\mathbf{I}_d)$
    $\bar{\mathbf{x}}_{r+1}^k \leftarrow \big([\bar{\mathbf{x}}_r^k]_{2:W}, \boldsymbol{\xi}_r\big)$
  end if
end for
return $\widehat{\mathbf{y}}^k$

B CHRONOSYNC ALGORITHM

Algorithm 3 summarizes the resulting ChronoSync Heun loop. At diffusion step i, each conditional model denoises its noisy window $\bar{\mathbf{x}}_n^k$ using the current cross-field estimates $\tilde{\mathbf{y}}_n^{\neg k}$ as conditioning. In the FRESH synchronization mode, after completing the predictor–corrector update for field k, its denoised estimate is immediately made available to the subsequent field models in the ordered sweep $k = 1, \dots, K$. With $\kappa = 1$, this recovers the sequential Lie–Trotter composition, while $0 < \kappa < 1$ under-relaxes the cross-field update according to $\tilde{\mathbf{y}}_n^k \leftarrow \kappa\, \mathbf{y}_n^{k,\text{new}} + (1-\kappa)\tilde{\mathbf{y}}_n^{k,\text{old}}$. We additionally consider a STALE synchronization mode, in which all conditional models within diffusion step i use the cross-field estimates available at the beginning of that step. Their new estimates are synchronized only after the complete field sweep and therefore enter the conditioning at diffusion step $i+1$.

Across PDE time, Algorithm 3 is applied for R rolling-window rollouts. After completing rollout r, the window is shifted forward by one PDE time frame and its first newly generated frame is committed to the coupled trajectory. When entering rollout $r+1$, the cross-field conditioning window therefore contains $W-1$ available frames from the current coupled trajectory. ChronoSync pads this missing frame independently for each field k by repeating the $W-1$ frame. The target windows are shifted in the same manner, except that their newly exposed terminal frame is initialized at the corresponding ERDM noise level and subsequently denoised through the coupled Heun loop. Consequently, synchronization is performed repeatedly both along diffusion time within each rollout and along PDE time as the window advances. Algorithm 4 presents the rolling composition loop of ChronoSync.

Algorithm 3 CHRONOSYNCHEUN loop

Require: noisy windows $\{\bar{\mathbf{x}}_n^k\}_{k=1}^K$, cross-field estimates $\{\tilde{\mathbf{y}}_n^k\}_{k=1}^K$, static conditioning $\mathbf{s}$, $n_{\text{cln}} = 1$, steps S, relaxation $\kappa \in [0, 1]$, mode $m \in \{\text{STALE}, \text{FRESH}\}$

$\Delta\tau \leftarrow 1/S$
for $i = 0, \dots, S-1$ **do**
  $\tau_i \leftarrow i/S, \quad \tau_{i+1} \leftarrow (i+1)/S$
  $\boldsymbol{\sigma}_i \leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_i), \quad \boldsymbol{\sigma}_{i+1} \leftarrow \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
  $\dot{\boldsymbol{\sigma}}_i \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_i)$
  $\tilde{\mathbf{y}}_n^{k,\text{old}} \leftarrow \tilde{\mathbf{y}}_n^k \quad \forall k$
  for $k = 1, \dots, K$ **do**
    if $m = \text{FRESH}$ **then**
      $\mathbf{c}_n^k \leftarrow (\tilde{\mathbf{y}}_n^{\neg k}, \mathbf{s})$
    else
      $\mathbf{c}_n^k \leftarrow (\tilde{\mathbf{y}}_n^{\neg k,\text{old}}, \mathbf{s})$
    end if
    $\hat{\mathbf{y}}^k \leftarrow \mathcal{D}_{\theta^k}^W\left(\bar{\mathbf{x}}_n^k \mid \mathbf{c}_n^k; \boldsymbol{\sigma}_i\right)$
    $\mathbf{d}^k \leftarrow \dot{\boldsymbol{\sigma}}_i \odot (\bar{\mathbf{x}}_n^k - \hat{\mathbf{y}}^k)/\boldsymbol{\sigma}_i$
    $\bar{\mathbf{z}}^k \leftarrow \bar{\mathbf{x}}_n^k + \Delta\tau\, \mathbf{d}^k$
    if $\tau_{i+1} < 1$ **then**
      $\dot{\boldsymbol{\sigma}}_{i+1} \leftarrow \partial_\tau \bar{\boldsymbol{\sigma}}^W(\tau_{i+1})$
      $\hat{\mathbf{y}}'^k \leftarrow \mathcal{D}_{\theta^k}^W\left(\bar{\mathbf{z}}^k \mid \mathbf{c}_n^k; \boldsymbol{\sigma}_{i+1}\right)$
      $\mathbf{d}'^k \leftarrow \dot{\boldsymbol{\sigma}}_{i+1} \odot (\bar{\mathbf{z}}^k - \hat{\mathbf{y}}'^k)/\boldsymbol{\sigma}_{i+1}$
      $\bar{\mathbf{x}}_n^k \leftarrow \bar{\mathbf{x}}_n^k + \frac{\Delta\tau}{2}(\mathbf{d}^k + \mathbf{d}'^k)$
      $\mathbf{y}^{k,\text{new}} \leftarrow \hat{\mathbf{y}}'^k$
    else
      $\bar{\mathbf{x}}_n^k \leftarrow \bar{\mathbf{z}}^k$
      $\mathbf{y}^{k,\text{new}} \leftarrow \hat{\mathbf{y}}^k$
    end if
    if $m = \text{FRESH}$ **then**
      $\tilde{\mathbf{y}}_n^k \leftarrow \kappa\, \mathbf{y}^{k,\text{new}} + (1-\kappa)\tilde{\mathbf{y}}_n^{k,\text{old}}$
    end if
  end for
  if $m = \text{STALE}$ **then**
    $\tilde{\mathbf{y}}_n^k \leftarrow \kappa\, \mathbf{y}^{k,\text{new}} + (1-\kappa)\tilde{\mathbf{y}}_n^{k,\text{old}} \quad \forall k$
  end if
  $[\bar{\mathbf{x}}_n^k]_{1:n_{\text{cln}}} \leftarrow [\mathbf{y}_n^k]_{1:n_{\text{cln}}} \quad \forall k$
end for
return $\{\bar{\mathbf{x}}_n^k\}_{k=1}^K, \{\tilde{\mathbf{y}}_n^k\}_{k=1}^K$

ChronoSync optionally performs $n_{\text{boot}}^{\text{roll}}$ additional bootstrap refinements before committing the newly generated frame of each rollout. After the first coupled reverse process, the current denoised windows are treated as updated cross-field estimates, re-noised according to the initial rolling-window noise schedule while preserving the clean prefix, and passed again through the ChronoSync Heun loop. Each bootstrap therefore re-solves the same PDE time window using the most recently synchronized coupled estimate as its prescribed state. After $n_{\text{boot}}^{\text{roll}}$ such refinements, the resulting window is written back to the rolling trajectory and the algorithm proceeds to rollout $r + 1$.

Algorithm 4 ChronoSync rolling composition

Require: initialized rolling windows $\{\bar{\mathbf{x}}_0^k\}_{k=1}^K$, initialized coupled trajectory buffers $\{\mathbf{Y}^k\}_{k=1}^K$ with W valid frames, static conditioning $\mathbf{s}$, window length W, $n_{\text{cln}} = 1$, Heun steps S, rollouts R, rolling bootstraps $n_{\text{boot}}^{\text{roll}}$, relaxation $\kappa \in [0, 1]$, mode $m \in \{\text{STALE}, \text{FRESH}\}$

$N_{\text{valid}} \leftarrow W$
$\mathbf{x}_{\text{win}}^k \leftarrow \bar{\mathbf{x}}_0^k \quad \forall k$
initialize output trajectories $\{\hat{\mathbf{Y}}^k\}_{k=1}^K$ with the provided n_{cln} frames
for $r = 1, \ldots, R$ **do**
  // Construct cross-field windows from the current coupled trajectory
  for $k = 1, \ldots, K$ **do**
    $[\tilde{\mathbf{y}}_r^k]_{1:W-1} \leftarrow$ latest $W - 1$ valid frames of $\mathbf{Y}^k$
    $[\tilde{\mathbf{y}}_r^k]_W \leftarrow [\tilde{\mathbf{y}}_r^k]_{W-1}$ ▷ pad missing terminal frame
  end for
  // Shift the target windows by one PDE time frame
  for $k = 1, \ldots, K$ **do**
    $[\bar{\mathbf{x}}_r^k]_{1:W-1} \leftarrow [\mathbf{x}_{\text{win}}^k]_{2:W}$
    $[\bar{\mathbf{x}}_r^k]_W \leftarrow [\bar{\boldsymbol{\sigma}}^W(0)]_W \odot \boldsymbol{\epsilon}_r^k, \quad \boldsymbol{\epsilon}_r^k \sim \mathcal{N}(0, I)$
  end for
  // Coupled reverse process
  $\left(\{\mathbf{x}_{\text{win}}^k\}_{k=1}^K, \{\tilde{\mathbf{y}}_r^k\}_{k=1}^K\right) \leftarrow$ ChronoSyncHeun $\left(\{\bar{\mathbf{x}}_r^k\}_{k=1}^K, \{\tilde{\mathbf{y}}_r^k\}_{k=1}^K, \mathbf{s}, n_{\text{cln}}, S, \kappa, m\right)$ from Algorithm 3
  // Optional rolling bootstrap refinement
  for $b = 1, \ldots, n_{\text{boot}}^{\text{roll}}$ **do**
    for $k = 1, \ldots, K$ **do**
      $\bar{\mathbf{x}}_r^{k,(b)} \leftarrow \tilde{\mathbf{y}}_r^k + \bar{\boldsymbol{\sigma}}^W(0) \odot \boldsymbol{\epsilon}_r^{k,(b)}, \quad \boldsymbol{\epsilon}_r^{k,(b)} \sim \mathcal{N}(0, I)$
      $[\bar{\mathbf{x}}_r^{k,(b)}]_{1:n_{\text{cln}}} \leftarrow [\mathbf{x}_{\text{win}}^k]_{1:n_{\text{cln}}}$
    end for
    $\left(\{\mathbf{x}_{\text{win}}^k\}_{k=1}^K, \{\tilde{\mathbf{y}}_r^k\}_{k=1}^K\right) \leftarrow$ ChronoSyncHeun $\left(\{\bar{\mathbf{x}}_r^{k,(b)}\}_{k=1}^K, \{\tilde{\mathbf{y}}_r^k\}_{k=1}^K, \mathbf{s}, n_{\text{cln}}, S, \kappa, m\right)$ from Algorithm 3
  end for
  // Commit one PDE time frame and update the rolling buffers
  for $k = 1, \ldots, K$ **do**
    $\mathbf{y}_r^{k,\text{new}} \leftarrow [\mathbf{x}_{\text{win}}^k]_{n_{\text{cln}}}$ ▷ single committed PDE time frame
    append $\mathbf{y}_r^{k,\text{new}}$ to $\hat{\mathbf{Y}}^k$
    $[\tilde{\mathbf{y}}_r^k]_{1:n_{\text{cln}}} \leftarrow [\mathbf{x}_{\text{win}}^k]_{1:n_{\text{cln}}}$
    $[\mathbf{Y}^k]_{N_{\text{valid}}-W+2:N_{\text{valid}}+1} \leftarrow [\tilde{\mathbf{y}}_r^k]_{1:W}$
  end for
  $N_{\text{valid}} \leftarrow N_{\text{valid}} + 1$
end for
return $\{\hat{\mathbf{Y}}^k\}_{k=1}^K$

C ChronoSync Error Analysis

C.1 Lie-Trotter operator splitting scheme for nonlinear ODEs

C.1.1 Autonomous systems

Let $\Omega \in \mathbb{R}^d$ and $\mathcal{F} : \Omega \to \mathbb{R}^d$, $\mathcal{G} : \Omega \to \mathbb{R}^d$ be smooth nonlinear vector fields. We consider the following autonomous system of nonlinear ODEs

$$\partial_t X = \mathcal{F}(X) + \mathcal{G}(X), \tag{8}$$

on some time interval $t \in [0, T]$, with $X(0) = X_0$. The solution can be defined in terms of the flow map $\Phi_{\mathcal{F}+\mathcal{G}}^t$ with $\Phi_{\mathcal{F}+\mathcal{G}}^t(X_0) = X(t; X_0)$. We remark that the flow map satisfies $\Phi_{\mathcal{F}+\mathcal{G}}^{t+s} = \Phi_{\mathcal{F}+\mathcal{G}}^t \circ \Phi_{\mathcal{F}+\mathcal{G}}^s$.

Motivated by the fact that the flow $\Phi^t_{\mathcal{F}+\mathcal{G}}$ may be challenging to compute directly, we split the vector field and consider the two subproblems

$$\partial_t V = \mathcal{F}(V) \quad \text{and} \quad \partial_t W = \mathcal{G}(W), \tag{9}$$

which may be substantially easier to solve for. We denote the flow maps associated with the subproblems equation 9 by $\Phi^t_{\mathcal{F}}$ and $\Phi^t_{\mathcal{G}}$.

To approximate the solution at final time T, we subdivide $[0, T]$ into smaller N intervals of length h with $t_n = nh$ for $1 \leq n \leq N$. Denoting X_n as the approximation for $X(t_n)$, we can approximate the solution of equation 8 at the next time point using a one-step scheme

$$X_{n+1} = \Phi^h_{\mathcal{F}+\mathcal{G}}(X_n). \tag{10}$$

As an alternative, we can use the split form described by the subproblems equation 9 and evaluate

$$X^{LT}_{n+1} = \Phi^h_{\mathcal{F}} \circ \Phi^h_{\mathcal{G}}(X_n), \tag{11}$$

which is known as the Lie-Trotter scheme (Blanes et al., 2024; Gao et al., 2026). This scheme results in a splitting error which is locally second-order

$$\left\|\Phi^h_{\mathcal{F}+\mathcal{G}}(X) - \Phi^h_{\mathcal{F}} \circ \Phi^h_{\mathcal{G}}(X)\right\| = \left\|\frac{h^2}{2}[\mathcal{F}, \mathcal{G}](X) + \mathcal{O}(h^3)\right\|, \tag{12}$$

where $[\mathcal{F}, \mathcal{G}](X) = \nabla_X \mathcal{G}(X)\mathcal{F}(X) - \nabla_X \mathcal{F}(X)\mathcal{G}(X)$ is the Lie bracket capturing the non commutativity of the flow maps. In particular, if the flow maps commute, i.e.,

$$\Phi^t_{\mathcal{F}} \circ \Phi^s_{\mathcal{G}} = \Phi^s_{\mathcal{G}} \circ \Phi^t_{\mathcal{F}}, \tag{13}$$

then, under sufficient smoothness assumptions, the Lie bracket vanishes ($[\mathcal{F}, \mathcal{G}](X) = 0$), which in turn guarantees that $\Phi^t_{\mathcal{F}+\mathcal{G}} = \Phi^t_{\mathcal{F}} \circ \Phi^t_{\mathcal{G}}$.

C.1.2 Non-autonomous systems

Let us now consider the system

$$\partial_t X = \mathcal{F}(X, t) + \mathcal{G}(X, t), \tag{14}$$

where the nonlinear vector fields are now defined as $\mathcal{F} : \Omega \times [0, T] \to \mathbb{R}^d$, $\mathcal{G} : \Omega \times [0, T] \to \mathbb{R}^d$ be smooth nonlinear vector fields. The solution can be defined in terms of the evolution map $\Phi^{t,s}_{\mathcal{F}+\mathcal{G}}$ where $\Phi^{t,s}_{\mathcal{F}+\mathcal{G}}(\tilde{X}) = X(t; s, \tilde{X})$, which is the solution at time t given the state $\tilde{X}$ at time s. For an autonomous system, the solution operator defines a one-parameter family of flow maps $\{\Phi^t_{\mathcal{F}+\mathcal{G}}\}_t$. For a non-autonomous system, the explicit dependence on time requires a two-parameter family of evolution maps $\{\Phi^{t,s}_{\mathcal{F}+\mathcal{G}}\}_{t,s}$, where s and t denote the initial and final times, respectively. We remark that the evolution maps satisfy $\Phi^{s,s}_{\mathcal{F}+\mathcal{G}} = I$ and $\Phi^{t,s}_{\mathcal{F}+\mathcal{G}} = \Phi^{t,r}_{\mathcal{F}+\mathcal{G}} \circ \Phi^{r,s}_{\mathcal{F}+\mathcal{G}}$. We also define the subproblems after splitting the operator

$$\partial_t V = \mathcal{F}(V, t) \quad \text{and} \quad \partial_t W = \mathcal{G}(W, t), \tag{15}$$

with their evolution maps denoted by $\Phi^{t,s}_{\mathcal{F}}$ and $\Phi^{t,s}_{\mathcal{G}}$.

We subdivide $[0, T]$ as done in the autonomous systems case. The one-step scheme is given by

$$X_{n+1} = \Phi^{t_{n+1},t_n}_{\mathcal{F}+\mathcal{G}}(X_n), \tag{16}$$

while a Lie-Trotter type split scheme similar to equation 11 is given by

$$X^{LT}_{n+1} = \Phi^{t_{n+1},t_n}_{\mathcal{F}} \circ \Phi^{t_{n+1},t_n}_{\mathcal{G}}(X_n). \tag{17}$$

It can be shown that equation 17 leads to a second-order perturbation of equation 16. To see this, we first consider

$$\partial_t W = \mathcal{G}(W, t) \quad \text{with} \quad W(t_n) = X_n. \tag{18}$$

The first stage of the splitting scheme gives

$$W(t_{n+1}) = \Phi^{t_{n+1},t_n}_{\mathcal{G}}(X_n). \tag{19}$$

Assuming sufficient regularity, a Taylor expansion of $W(t_n + h)$ about t_n leads to the expression

$$\begin{aligned} W(t_n + h) =& X_n + h\mathcal{G}(X_n, t_n) + \frac{h^2}{2}\partial_t\mathcal{G}(X_n, t_n) \\ &+ \frac{h^2}{2}\nabla_X\mathcal{G}(X_n, t_n)\mathcal{G}(X_n, t_n) + \mathcal{O}(h^3). \end{aligned} \tag{20}$$

where we have also utilized equation 18. For compactness, let us write $W_{n+1} = W(t_{n+1})$.

Next, consider

$$\partial_t V = \mathcal{F}(V, t) \quad \text{with} \quad V(t_n) = W_{n+1}. \tag{21}$$

Then, the second stage of the splitting scheme giving us

$$X^{LT}_{n+1} = V(t_{n+1}) = \Phi^{t_{n+1},t_n}_{\mathcal{F}}(W_{n+1}). \tag{22}$$

Using Taylor expansion of $V(t_n + h)$ about t_n and equation 21, we get

$$\begin{aligned} X^{LT}_{n+1} = V(t_n + h) = W_{n+1} + h\mathcal{F}(W_{n+1}, t_n) + \frac{h^2}{2}\partial_t\mathcal{F}(W_{n+1}, t_n) \\ + \frac{h^2}{2}\nabla_X\mathcal{F}(W_{n+1}, t_n)\mathcal{F}(W_{n+1}, t_n) + \mathcal{O}(h^3). \end{aligned} \tag{23}$$

Now note that we have from equation 21 that

$$W_{n+1} = X_n + h\mathcal{G}(X_n, t_n) + \mathcal{O}(h^2). \tag{24}$$

Thus, expanding $\mathcal{F}(W_{n+1}, t_n)$ about X_n based on equation 24 leads to

$$\begin{aligned} \mathcal{F}(W_{n+1}, t_n) &= \mathcal{F}(X_n, t_n) + \nabla_X\mathcal{F}(W_{n+1}, t_n)(W_{n+1} - X_n) + \mathcal{O}\left(|W_{n+1} - X_n|^2\right) \\ &= \mathcal{F}(X_n, t_n) + h\nabla_X\mathcal{F}(W_{n+1}, t_n)\mathcal{G}(X_n, t_n) + \mathcal{O}(h^2). \end{aligned} \tag{25}$$

Similar expansions give us

$$\partial_t\mathcal{F}(W_{n+1}, t_n) = \partial_t\mathcal{F}(X_n, t_n) + \mathcal{O}(h), \tag{26}$$

and

$$\nabla_X\mathcal{F}(W_{n+1}, t_n)\mathcal{F}(W_{n+1}, t_n) = \nabla_X\mathcal{F}(X_n, t_n)\mathcal{F}(X_n, t_n) + \mathcal{O}(h). \tag{27}$$

Plugging equation 24 - equation 27 into equation 23 gives us

$$\begin{aligned} X^{LT}_{n+1} =& X_n + h(\mathcal{F}(X_n, t_n) + \mathcal{G}(X_n, t_n)) + \frac{h^2}{2}\left(\partial_t\mathcal{F}(X_n, t_n) + \partial_t\mathcal{G}(X_n, t_n)\right) \\ &+ \frac{h^2}{2}\left(\nabla_X\mathcal{F}(X_n, t_n)\mathcal{F}(X_n, t_n) + \nabla_X\mathcal{G}(X_n, t_n)\mathcal{G}(X_n, t_n)\right) \\ &+ h^2\nabla_X\mathcal{F}(X_n, t_n)\mathcal{G}(X_n, t_n) + \mathcal{O}(h^3). \end{aligned} \tag{28}$$

Next, for the one-step scheme we consider

$$\partial_t X = \mathcal{F}(X, t) + \mathcal{G}(X, t) \quad \text{with} \quad X(t_n) = X_n, \tag{29}$$

and the approximation at t_{n+1} given by

$$X_{n+1} = \Phi^{t_{n+1},t_n}_{\mathcal{F}+\mathcal{G}}(X_n). \tag{30}$$

Performing a Taylor expansion about t_n leads to

$$\begin{aligned} X_{n+1} =& X_n + h(\mathcal{F}(X_n, t_n) + \mathcal{G}(X_n, t_n)) + \frac{h^2}{2}\left(\partial_t\mathcal{F}(X_n, t_n) + \partial_t\mathcal{G}(X_n, t_n)\right) \\ &+ \frac{h^2}{2}\left(\nabla_X\mathcal{F}(X_n, t_n)\mathcal{F}(X_n, t_n) + \nabla_X\mathcal{G}(X_n, t_n)\mathcal{G}(X_n, t_n)\right) \\ &+ \frac{h^2}{2}\left(\nabla_X\mathcal{F}(X_n, t_n)\mathcal{G}(X_n, t_n) + \nabla_X\mathcal{G}(X_n, t_n)\mathcal{F}(X_n, t_n)\right) + \mathcal{O}(h^3). \end{aligned} \tag{31}$$

Thus, the error between the one-step approximation and the Lie-Trotter approximation is given by the (normed) difference between equation 31 and equation 28

$$\left\|X_{n+1} - X_{n+1}^{LT}\right\| = \left\| \frac{h^2}{2} \underbrace{\left(\nabla_X \mathcal{G}(X_n, t_n)\mathcal{F}(X_n, t_n) - \nabla_X \mathcal{F}(X_n, t_n)\mathcal{G}(X_n, t_n)\right)}_{\text{Lie Bracket } [\mathcal{F},\mathcal{G}](X_n,t_n)} + \mathcal{O}(h^3) \right\|, \quad (32)$$

which is the same as the second-order error estimate equation 12 for the autonomous system.

Other splitting schemes have also been proposed by first extending the system to larger autonomous system (Blanes et al., 2024). They also lead to second-order local approximations similar to equation 28 but may bring in a dependence on the temporal derivatives of the vector fields in the leading error term.

C.2 Lie-Trotter operator splitting for probability flows in EDMs

C.2.1 Probability flow operators

Let f and g be two coupled physics variables associated with a time-dependent PDE system. We restrict our discussions to two spatial dimensions and assuming that f and g are scalar-valued variables, but the formulation below can be easily extended to higher dimensional settings. We define a structured spatiotemporal grid

$$\mathcal{Q} = \{(x_i, y_j, t_n) \ : \ 1 \le i \le N_x, \ 1 \le j \le N_y, \ \ 1 \le n \le N_t\}.$$

Let the discrete coupled state vector be given by $X = (X_f, X_g)$ where $X_f, X_g \in \mathbb{R}^{N_x \times N_y \times N_t}$ are the noisy estimates of f, g on $\mathcal{Q}$, respectively.

For an additive Gaussian perturbation with covariance $C(\tau)$, which depends on the probalility flow time τ, the ODE for the state evolution can be written as

$$\frac{dX}{d\tau} = \mathcal{V}(X, \tau) = \frac{1}{2}\dot{C}(\tau)C(\tau)^{-1}\Big(X - D\left(X; C(\tau)\right)\Big), \quad (33)$$

where $D(.)$ is the denoiser and $\mathcal{V} = (\mathcal{V}_f, \mathcal{V}_g)$ is the coupled probability flow field over the coupled state X. For a standard EDM probability flow, the same scalar noise is applied to every component of X, i.e., $C(t) = \sigma(\tau)^2 I$, with the ODE reducing to

$$\frac{dX}{d\tau} = \frac{\dot{\sigma}(\tau)}{\sigma(\tau)}\Big(X - D(X; \sigma(\tau))\Big). \quad (34)$$

In the specific EDM formulation by Karras et al. (2022), the authors suggest choosing the noise level as $\sigma(\tau) = \tau$, which further simplifies the ODE to

$$\frac{dX}{d\sigma} = \frac{X - D(X; \sigma)}{\sigma}. \quad (35)$$

Working with the more general form equation 33, we split the vector field as $\mathcal{V}(X, \tau) = \mathcal{F}(X, \tau) + \mathcal{G}(X, \tau)$ where

$$\mathcal{F}(X, \tau) = (\mathcal{V}_f(X, \tau), 0), \quad \mathcal{G}(X, \tau) = (0, \mathcal{V}_g(X, \tau)). \quad (36)$$

We can now approximate the solution using the Lie-Trotter scheme developed in Section C.1.2 for non-autonomous systems. However, instead of working with a uniform discretization, we work with a non-uniform τ discretization associated with S steps of a Heun integrator (Karras et al., 2022; Rühling Cachay et al., 2026), with the ordering $\tau_S < \tau_{S-1} < \cdots < \tau_1 < \tau_0$. Note that this typically leads to a non-uniform noise schedule, even for the Karras et al. (2022) model. By setting $h_k = |\tau_k - \tau_{k-1}|$, we can adapt equation 32 to obtain the local error due to the splitting scheme at step k

$$\mathcal{E}_k = \left\| \frac{h_k^2}{2}[\mathcal{F}, \mathcal{G}](X, \tau_k) + \mathcal{O}(h_k^3) \right\|. \quad (37)$$

Summing the error over all S steps, the accumulated local Lie-Trotter error becomes

$$\mathcal{E} = \sum_{k=1}^{S} \left\| \frac{h_k^2}{2}[\mathcal{F}, \mathcal{G}](X, \tau_k) + \mathcal{O}(h_k^3) \right\|, \quad (38)$$

which is first-order in h.

C.2.2 EXPLICIT PDE TIME REPRESENTATION

For further analysis, we reformulate the probability flow problem so that the PDE time is treated explicitly. Let the state X be expressed as $X = \left(X^{(1)}, \cdots, X^{(N_t)}\right)$ where $X^{(n)} = \left(X_f^{(n)}, X_g^{(n)}\right)$, and $X_f^{(n)}, X_g^{(n)} \in \mathbb{R}^{N_x \times N_y}$ are the noisy estimates of f, g, respectively, on the $N_x \times N_y$ spatial grid at the PDE time t_n. The EDM model in equation 33 can be defined for this X, with the flow field split as

$$\mathcal{V}(X,\tau) = \mathcal{F}(X,\tau) + \mathcal{G}(X,\tau) = \sum_{n=1}^{N_t} \mathcal{F}^{(n)}(X,\tau) + \sum_{m=1}^{N_t} \mathcal{G}^{(m)}(X,\tau),$$

where $\mathcal{F}^{(n)}$ and $\mathcal{G}^{(m)}$ have block support only on $X_f^{(n)}$ and $X_g^{(m)}$, respectively. In other words, $\mathcal{F}^{(n)}$ is zero outside the $X_f^{(n)}$ block of X, while $\mathcal{G}^{(m)}$ is zero outside the $X_g^{(n)}$ block.

The bilinearity of the Lie bracket gives[1]

$$\begin{aligned}[\mathcal{F},\mathcal{G}] &= \Big[\sum_{n=1}^{N_t} \mathcal{F}^{(n)}, \sum_{m=1}^{N_t} \mathcal{G}^{(m)}\Big](X,\tau) \\ &= \sum_{n=1}^{N_t} \Big[\mathcal{F}^{(n)}, \mathcal{G}^{(n)}\Big](X,\tau) + \sum_{\substack{n,m=1 \\ n\neq m}}^{N_t} \Big[\mathcal{F}^{(n)}, \mathcal{G}^{(m)}\Big](X,\tau).\end{aligned} \tag{39}$$

Using equation 39 in equation 38 gives us the bound

$$\mathcal{E} \leq \underbrace{\sum_{k=1}^{\mathcal{S}} \frac{h_k^2}{2} \sum_{n=1}^{N_t} \left\| [\mathcal{F}^{(n)}, \mathcal{G}^{(n)}](X, \tau_k) \right\|}_{\text{Ia}} + \underbrace{\sum_{k=1}^{\mathcal{S}} \frac{h_k^2}{2} \sum_{\substack{n,m=1 \\ n\neq m}}^{N_t} \left\| [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}](X, \tau_k) \right\|}_{\text{Ib}} + \underbrace{\mathcal{O}(h^2)}_{\text{II}}. \tag{40}$$

where $h = \max_k h_k$.

We note that the splitting error can be separated into two components. One arising from the non-commutativity of the operators which is first order, i.e., Ia + Ib. The other arising from the higher order terms, i.e., II. Additionally, the non-commutativity error can be split into same PDE time (or diagonal) and cross PDE time (or non-diagonal) components. The diagonal terms represent the non-commutativeness of the two physics probability flows at the same physical time. The off-diagonal terms represent the non-commutativeness of updating the probability flow of one physics field at PDE time t_n and the change in the probability flow of the other physics field at a different PDE time t_m.

C.3 OPERATOR SPLITTING IN ERDMS

C.3.1 A SINGLE COMMITTED PDE TIME FRAME

ERDMs, unlike EDMs, operate over a window of W frames where $W < N_t$. Let r denote the rollout index and k denote the denoising-step index. We write

$$X_{r,k} = \left(X_{r,k}^{(1)}, ..., X_{r,k}^{(W)}\right), \quad X_{r,k}^{(n)} = \left([X_f]_{r,k}^{(n)}, [X_g]_{r,k}^{(n)}\right), \tag{41}$$

where $[X_f]_{r,k}^{(n)}$ and $[X_g]_{r,k}^{(n)}$ are the noisy estimates of f and g, respectively, on the $N_x \times N_y$ spatial grid at the PDE time t_{r+n}, and after k denoising steps within rollout r. Furthermore, the noise levels are different for each frame of the window

$$\boldsymbol{\sigma}(\tau) = \left(\sigma^{(1)}(\tau), \sigma^{(2)}(\tau), \cdots, \sigma^{(W)}(\tau)\right),$$

[1] Consider the operators $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{H}$. Bilinearity of the Lie bracket implies $[\mathcal{F}+\mathcal{G},\mathcal{H}] = [\mathcal{F},\mathcal{H}] + [\mathcal{G},\mathcal{H}]$ and $[\mathcal{F},\mathcal{G}+\mathcal{H}] = [\mathcal{F},\mathcal{G}] + [\mathcal{F},\mathcal{H}]$.

with

$$C(\tau) = \text{diag}\left((\sigma^{(1)}(\tau))^2 I, (\sigma^{(2)}(\tau))^2 I, \cdots, (\sigma^{(W)}(\tau))^2 I\right),$$

where each block corresponds to one frame $X_{r,k}^{(n)}$. In (Rühling Cachay et al., 2026), the authors choose the noise levels such that $\sigma^{(1)}(\tau) < \sigma^{(2)}(\tau) < \cdots < \sigma^{(W)}(\tau)$ to build in higher uncertainty in the prediction of future states.

During rollout r, the Heun loop jointly denoises all W frames over S denoising steps. Thus, each window advances from $X_{r,0}$ to $X_{r,S}$. We set $n_{\text{cln}} = 1$ throughout this paper and under this setting ERDM commits/accepts the first time frame after a rollout (i.e., one full Heun loop). The remaining time frames, which are partially denoised, will be retained for the next rollout after shifting by n_{cln} frames. With this setup in place, the windowed state for the next rollout is initialized as

$$X_{r+1,0} = \left(X_{r,S}^{(2)}, ..., X_{r,S}^{(W)}, X_{r+1,0}^{(W)}\right) \tag{42}$$

where $X_{r+1,0}^{(W)}$ is a newly initialized noise frame, while

$$X_{r+1,0}^{(n)} = X_{r,S}^{(n+1)} \quad \forall\, n = 1, ..., W-1. \tag{43}$$

Since we are operating only on W frames, we split the probability vector field as

$$\mathcal{V}(X_{r,k}, \tau_k) = \mathcal{F}(X_{r,k}, \tau_k) + \mathcal{G}(X_{r,k}, \tau_k),$$

where

$$\mathcal{F}(X_{r,k}, \tau_k) = \sum_{n=1}^{W} \mathcal{F}^{(n)}(X_{r,k}, \tau_k), \quad \mathcal{G}(X_{r,k}, \tau_k) = \sum_{m=1}^{W} \mathcal{G}^{(m)}(X_{r,k}, \tau_k).$$

We introduce the shorthand notation for the Lie brackets

$$[\mathcal{F}, \mathcal{G}]_{r,k} := [\mathcal{F}, \mathcal{G}]\,(X_{r,k}, \tau_k)\,, \quad [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]_{r,k} := [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]\,(X_{r,k}, \tau_k)\,. \tag{44}$$

We denote the accumulated Lie-Trotter perturbation generated over the entire W-frame window at rollout r as η_r, which is given by (adapting the discussion in Section C.2.2)

$$\begin{aligned}\eta_r &= \sum_{k=1}^{S} \left(\frac{h_k^2}{2}[\mathcal{F}, \mathcal{G}]_{r,k} + \mathcal{O}(h_k^3)\right) \\ &= \sum_{k=1}^{S} \frac{h_k^2}{2} \sum_{n=1}^{W} [\mathcal{F}^{(n)}, \mathcal{G}^{(n)}]_{r,k} + \sum_{k=1}^{S} \frac{h_k^2}{2} \sum_{\substack{n,m=1 \\ n \neq m}}^{W} [\mathcal{F}^{(n)}, \mathcal{G}^{(m)}]_{r,k} + \mathcal{O}(h^2).\end{aligned} \tag{45}$$

C.3.2 Multiple committed PDE time frames

The perturbation committed after rollout r is not the perturbation inherited by the subsequent rollout. After committing the first frame of the window, ERDM shifts the overlapping window forward by one frame. Hence, the perturbation propagated between rollouts must be defined over the entire W-frame rolling state. For brevity, we write $X_r := X_{r,0}$ for the window entering rollout r. Let $\delta_r = \left(\delta_r^{(1)}, \ldots, \delta_r^{(W)}\right)$ denote the perturbation of the rolling state entering rollout r due to splitting errors accumulated during previous rollouts, with δ_r having a form similar to equation 45.

At rollout r, let Ψ_r denote the differentiable reference (unsplit) denoising map over a completed Heun loop, which advances the entire window from X_r to $\Psi_r(X_r) = X_{r,S}$. This map represents one (nonlinear) ERDM denoising pass over all the frames in this window. To make the analysis tractable, we make a local linearization via a first order Taylor approximation and define the Jacobian $J_r \equiv \nabla_X \Psi_r(X_r)$, where J_r is a $W \times W$ block matrix. Each block is over the entire spatial field. The Jacobian block $J_{r(i,j)}$ measures how perturbations in frame j influence frame i. Applying the reference map to the perturbed window gives

$$\Psi_r(X_r + \delta_r) = \Psi_r(X_r) + J_r \delta_r + \mathcal{O}(\|\delta_r\|^2). \tag{46}$$

During rollout r, ERDM performs Heun integration and introduces operator splitting perturbation η_r. After rollout r and before shifting, the perturbation is (ignoring higher order terms)

$$\tilde{\delta}_r = J_r \delta_r + \eta_r. \tag{47}$$

The first term linearly approximates the inherited accumulated perturbation from the previous rollout into r. The second term represents new perturbation from operator splitting. In the inherited perturbation term $J_r\delta_r$, all components of δ_r contribute. However, after shifting the window in ERDMs, the first component of the perturbation which represents the committed frame from the previous rollout does not propagate to the next rollout. To formalize this, we define a window shift operator

$$\mathcal{S}_W \left(z_r^{(1)}, z_r^{(2)}, \ldots, z_r^{(W)} \right) \equiv \left(z_r^{(2)}, \ldots, z_r^{(W)}, 0 \right), \tag{48}$$

where $z_r^{(j)}$ represents the perturbation at frame position j within rollout r. Therefore, the perturbation entering the new rollout is

$$\delta_{r+1} = \mathcal{S}_W \left(\tilde{\delta}_r \right). \tag{49}$$

We define the sensitivity coefficient $L_r \equiv \|\mathcal{S}_W J_r\|$ which measures the local sensitivity of the shifted rolling state to perturbations inherited/accumulated from the previous rollout. Then, taking the norm of equation 49 and applying a triangle inequality gives

$$\|\delta_{r+1}\| \leq L_r \|\delta_r\| + \|\mathcal{S}_W \eta_r\|. \tag{50}$$

C.4 Recursive rollout error bound

We begin by summarizing the necessary assumptions that would allow us to use the mathematical machinery developed in the previous sections.

Assumption 1 (Smooth vector fields)**.** *The probability vector fields $\mathcal{F}$ and $\mathcal{G}$ associated with the operator splitting have sufficient regularity, allowing us to obtain the ERDM perturbation bound η_r in equation 45.*

Assumption 2 (Perfect denoiser)**.** *We assume the denoiser D used in the probability flow model is learned perfectly. Note that the consequences of relaxing this assumption are discussed in Section C.6.*

Assumption 3 (Regularity of reference map)**.** *We assume the reference map Ψ_r for each rollout index r is locally regular (at least twice differentiable) to justify the Taylor series expansion 46. Furthermore, noting that the shift operator $\mathcal{S}_W$ is a bounded linear operator, we assume that the sensitivity coefficient $L_r := \|\mathcal{S}_W J_r\|$ is finite for every rollout index r.*

Assumption 4 (Finite perturbation)**.** *We assume that the local splitting perturbation η_r over the entire window W, and the accumulated perturbation δ_r from the previous rollout, are finite for every rollout index r.*

We now state the various results quantifying the rollout error bounds.

Theorem 2 (Recursive rollout error bound)**.** *Given the one rollout error bound*

$$\|\delta_{r+1}\| \leq L_r \|\delta_r\| + \|\mathcal{S}_W \eta_r\|$$

the perturbation error after R rollouts is bounded by the propagated initial full window perturbation error plus the sum of all splitting perturbations weighted by the sensitivities of subsequent rollouts.

$$\boxed{\|\delta_{R+1}\| \leq \left(\prod_{q=1}^{R} L_q \right) \|\delta_1\| + \sum_{r=1}^{R} \left(\prod_{q=r+1}^{R} L_q \right) \|\mathcal{S}_W \eta_r\|} \tag{51}$$

Proof. After the first rollout we get

$$\|\delta_2\| \leq L_1 \|\delta_1\| + \|\mathcal{S}_W \eta_1\|, \tag{52}$$

and after a subsequent rollout we get

$$\|\delta_3\| \leq L_2\|\delta_2\| + \|\mathcal{S}_W\eta_2\| \tag{53}$$

Substituting equation 52 into equation 53 yields

$$\|\delta_3\| \leq L_1L_2\|\delta_1\| + L_2\|\mathcal{S}_W\eta_1\| + \|\mathcal{S}_W\eta_2\|.$$

Repeating this R times, the recursive nature of the operations gives the required result under the convention that an empty product equals 1. □

If the initial window is exactly provided, then, $\|\delta_1\| \approx 0$ and the recursive rollout error bound in equation 51 can be simplified to

$$\|\delta_{R+1}\| \lesssim \sum_{r=1}^{R}\left(\prod_{q=r+1}^{R} L_q\right)\|\mathcal{S}_W\eta_r\|. \tag{54}$$

Expanding η_r in equation 54 results in

$$\begin{aligned}\|\delta_{R+1}\| \lesssim \sum_{r=1}^{R}\left(\prod_{q=r+1}^{R} L_q\right)\Big\|\mathcal{S}_W \sum_{k=1}^{\mathcal{S}}\Big[\frac{h_k^2}{2}\sum_{j=1}^{W}[\mathcal{F}^{(j)},\mathcal{G}^{(j)}]_{r,k}\Big] \\ + \frac{h_k^2}{2}\sum_{\substack{j,m=1\\ j\neq m}}^{W}[\mathcal{F}^{(j)},\mathcal{G}^{(m)}]_{r,k} + \mathcal{O}(h_k^3)\Big\|.\end{aligned} \tag{55}$$

The commutator terms quantify the splitting perturbations introduced by the non-commutativity of $\mathcal{F}$ and $\mathcal{G}$ and the sensitivity factors L_q control how these perturbations propagate through subsequent rollouts. Sensitivity factors below one contract perturbations, while factors above one allow perturbations to grow. Thus, small commutators may not guarantee a small error if the products of subsequent sensitivity factors are large.

C.5 Regime analysis

Lemma 1 (Bounded per-rollout splitting perturbation)**.** *Assume that the conditional probability flow fields are sufficiently smooth in a neighborhood of the rolling trajectory and that their aggregate Lie-bracket magnitude satisfies*

$$\mathcal{B}_{r,k} := \sum_{j=1}^{W}\|[F^{(j)},G^{(j)}]_{r,k}\| + \sum_{\substack{j,m=1\\ j\neq m}}^{W}\|[F^{(j)},G^{(m)}]_{r,k}\| \leq B \tag{56}$$

uniformly over rollouts r and denoising steps k. Assume additionally that the local higher-order terms are bounded by $C_3h_k^3$ and that $\mathcal{S}_W$ is bounded. Then

$$\|\mathcal{S}_W\eta_r\| \leq \|\mathcal{S}_W\|\left(\frac{B}{2}h + C_3h^2\right) =: \epsilon \tag{57}$$

where $h = \max_k h_k$.

Proof. Taking norms of the expansion of η_r and applying the triangle inequality gives

$$\|S_W\eta_r\| \leq \|S_W\|\sum_{k=1}^{S}\left(\frac{h_k^2}{2}\mathcal{B}_{r,k} + C_3h_k^3\right). \tag{58}$$

Using $\mathcal{B}_{r,k} \leq B$ and $\sum_k h_k = 1$ gives

$$\sum_k h_k^2 \leq h, \qquad \sum_k h_k^3 \leq h^2, \tag{59}$$

which results in equation 57. □

Corollary 2 (Non-expansive)**.** *If in equation 51 $L_r < 1 \; \forall \; r$, then the previous splitting errors are not amplified by rolling and*

$$\|\delta_{R+1}\| \leq \|\delta_1\| + \sum_{r=1}^{R} \|\mathcal{S}_W \eta_r\|. \tag{60}$$

Proof. If $L_r < 1 \; \forall \; r$, we can then write

$$\prod_{q=r+1}^{R} L_q \leq 1$$

Then, equation 60 directly follows. □

The above does not say the error stays constant because new local errors are still injected each rollout. However, there is no multiplicative explosion.

Corollary 3 (Non-expansive ($L_r \leq L < 1$) and bounded splitting error)**.** *If in equation 51 $L_r \leq L < 1 \; \forall \; r$ and also the splitting error is bounded $\|\mathcal{S}_W \eta_r\| \leq \epsilon \; \forall \; r$ following Lemma 1, we can then show that*

$$\lim_{R \to \infty} \sup \|\delta_R\| \leq \frac{\epsilon}{1 - L} \tag{61}$$

Proof. If $L_r \leq L < 1 \; \forall \; r$ and also $\|\mathcal{S}_W \eta_r\| \leq \epsilon \; \forall \; r$ then from equation 51

$$\|\delta_{R+1}\| \leq L^R \|\delta_1\| + \epsilon \sum_{j=0}^{R-1} L^j$$

The geometric series reduction of the summation gives ($L \neq 1$)

$$\|\delta_{R+1}\| \leq L^R \|\delta_1\| + \epsilon \frac{1 - L^R}{1 - L}$$

Taking the limit $R \to \infty$ results in equation 61. □

The above says if the rolling map is uniformly non-expansive and each rollout introduces a bounded amount of splitting error, then operator splitting perturbations do not grow without bound with rollout horizon.

Corollary 4 (Neutral and bounded splitting error)**.** *If in equation 51 $L_r \approx 1 \; \forall \; r$ and also the splitting error is bounded $\|\mathcal{S}_W \eta_r\| \leq \epsilon \; \forall \; r$ following Lemma 1, we can then show that*

$$\|\delta_{R+1}\| \lesssim \|\delta_1\| + R\epsilon \tag{62}$$

Proof. The proof directly follows from utilizing equation 51. □

The above says worst case growth is linear rather than exponential.

Corollary 5 (Expansive and bounded splitting error)**.** *If in equation 51 $L_r \approx L > 1 \; \forall \; r$ and also the splitting error is bounded $\|\mathcal{S}_W \eta_r\| \leq \epsilon \; \forall \; r$ following Lemma 1, the error then grows as $\mathcal{O}(L^R)$.*

Proof. The proof directly follows from utilizing equation 51 giving

$$\|\delta_{R+1}\| \lesssim L^R \|\delta_1\| + L^R \epsilon$$

which is a $\mathcal{O}(L^R)$ error growth. □

The above is the absolute worst case, which could occur in dynamically sensitive systems like chaotic systems.

Corollary 6 (Mixed regime)**.** *For a constant C_L, if*

$$\prod_{q=r+1}^{R} L_q \leq C_L \rho^{R-r}, \quad 0 < \rho < 1$$

and the splitting error is bounded $\|\mathcal{S}_W \eta_r\| \leq \epsilon$ following Lemma 1, then

$$\lim_{R\to\infty} \sup \|\delta_R\| \leq \frac{C_L \epsilon}{1-\rho} \tag{63}$$

Proof. Substituting $C_L \rho^{R-r}$ in equation 51 gives

$$\|\delta_{R+1}\| \leq C_L \rho^{R-1} \|\delta_1\| + C_L \epsilon \sum_{r=1}^{R} \rho^{R-r}$$

A geometric series reduction gives

$$\|\delta_{R+1}\| \leq C_L \rho^{R-1} \|\delta_1\| + C_L \epsilon \frac{1-\rho^R}{1-\rho}$$

Taking the limit $R \to \infty$ gives the required equation 63. □

Not all L_r need to be uniformly non-expansive. Some can be expansive. Provided that they are sufficiently offset by contractive rollouts, then perturbations decay cumulatively along the rolling trajectory.

Remark 1 (Exponential stability)**.** *The product*

$$\prod_{q=r+1}^{R} L_q$$

in Corollary 6 measures the amplification introduced due to perturbation at rollout r through subsequent rollouts. The mixed regime condition is a scalar sufficient condition analogous to uniform exponential stability of a non-autonomous discrete-time dynamical systems (Potzsche, 2010). Individual rollouts may be locally expansive ($L_r > 1$). The product bound over long horizons requires only that their cumulative sensitivity products decay geometrically.

C.6 Discussion

Denoiser error In equation 51, the denoiser $D(.)$ was assumed to be perfect. However, in practice, the denoiser is approximate

$$\hat{\mathcal{V}}(X, \sigma) = \frac{X - \hat{D}(X; \sigma)}{\sigma} \tag{64}$$

This introduces the probability flow error $\|\hat{\mathcal{V}}(X, \sigma) - \mathcal{V}(X, \sigma)\|$ away from $\sigma = 0$. We introduce the model perturbation due to denoiser error dependent on rollout r as ζ_r. Inserting this alongside the operator splitting perturbation η_r in equation 49 and then taking the norm and applying the Triangle inequality as in equation 50 gives

$$\|\delta_{R+1}\| \leq \left(\prod_{q=1}^{R} L_q\right) \|\delta_1\| + \sum_{r=1}^{R} \left(\prod_{q=r+1}^{R} L_q\right) \Big[\|\mathcal{S}_W \eta_r\| + \|\mathcal{S}_W \zeta_r\|\Big] \tag{65}$$

If the denoiser error remains bounded $\|\mathcal{S}_W \zeta_r\| \leq \xi$, then using Corollary 6, we can similarly obtain the bound

$$\lim_{R\to\infty} \sup \|\delta_R\| \leq \frac{C_L(\epsilon + \xi)}{1-\rho} \tag{66}$$

The affect of the denoiser error is to introduce an additional perturbation over the perturbation introduced due to operator splitting at each rollout.

D BENCHMARK METHODS: M2PDE-AR (DECOUPLED DATA), GENCP-AR (DECOUPLED DATA), AND FNO-AR-JOINT (COUPLED DATA)

D.1 M2PDE-AR: BLOCK AUTOREGRESSIVE EXTENSION OF M2PDE

D.1.1 PROBLEM SETUP

M2PDE (Zhang et al., 2025) models the solution associated with each decoupled field as a conditional diffusion model (DDPM Ho et al. (2020)) and composes these models during reverse diffusion to obtain the coupled fields. In its original formulation, the decoupled field may contain the complete temporal trajectory. This becomes computationally prohibitive for the long-horizon spatio-temporal problems considered in this work. We therefore extend the M2PDE to long time horizon problems with M2PDE-AR. M2PDE-AR is a block autoregressive adaptation that preserves the M2PDE field composition procedure while restricting each diffusion model to a temporal block of W frames.

Let $\mathbf{y}_n^k := y_{n+1:n+W}^k$ and $\mathbf{y}_n^{\neg k} := y_{n+1:n+W}^{\neg k}$ denote temporal length W target and conditioning windows, respectively. The conditional models used by M2PDE-AR are:

$$p_{\theta_k}^{W,\mathrm{AR}}(\mathbf{y}_n^k | \mathbf{y}_{n-W}^k, \mathbf{y}_n^{\neg k}, \mathbf{s}), \tag{67}$$

where $\mathbf{y}_{n-W}^k$ is the immediately preceding target block and $\mathbf{s}$ contains static conditioning information like ICs/BCs and source or forcing functions. Compared with $p_{\theta_k}^W(\mathbf{y}_n^k | \mathbf{y}_n^{\neg k}, \mathbf{s})$ used by ChronoSync, M2PDE-AR additionally conditions on the preceding block of the generated field. The complete future block is generated jointly.

D.1.2 TRAINING

Training constructs the adjacent block pair $\{\mathbf{y}_{n-W}^k, \mathbf{y}_n^k\}$ and uses the prescribed other fields over the future block. Only the target future block $\mathbf{y}_n^k$ is noised/denoised. For diffusion index l,

$$\mathbf{x}_{n,l}^k = \sqrt{\bar{\alpha}_l}\, \mathbf{y}_n^k + \sqrt{1-\bar{\alpha}_l}\, \epsilon, \quad \text{with } \epsilon \sim \mathcal{N}(0,1), \tag{68}$$

and the diffusion model is trained using the standard DDPM training objective (Ho et al., 2020)

$$\mathcal{L}_k = \mathbb{E}\Big[\| \epsilon - \epsilon_{\theta_k}(\mathbf{x}_{n,l}^k | \mathbf{y}_{n-W}^k, \mathbf{y}_n^{\neg k}, \mathbf{s}) \|_2^2 \Big]. \tag{69}$$

The preceding target block and prescribed fields are clean conditioning variables. A stride δ could be used during training to control the temporal spacing between training iteration samples. Each training sample still always maps a block length W along the temporal axis to its adjacent future block. So, $0 < \delta < W$ produces overlapping training samples without changing the physical prediction horizon of a block.

D.1.3 LONG-HORIZON M2PDE-AR COMPOSITION

Let $n_b = bW$ the block index with $b = 0, \ldots, B-1$ and B is the number of blocks. The first block $\{\mathbf{y}_0^k\}_{k=1}^K$ is provided exactly. For subsequent blocks $b \geq 1$, M2PDE-AR samples $\{\mathbf{y}_{n_b}^k\}_{k=1}^K$ using the original M2PDE compositional algorithm by Zhang et al. (2025). During this process, $\mathbf{y}_{n_b-1}^k$ is fixed as autoregressive history for field k. The unknown cross-fields $\mathbf{y}_{n_b}^{\neg k}$ are not explicitly prescribed but are progressively estimated by the other diffusion models during the same denoising process.

Following M2PDE, each outer refinement r maintains estimates of the clean future block fields. At reverse diffusion step l, the conditioning estimate for the other fields is blended as

$$\tilde{\mathbf{y}}_{n_b,l}^{\neg k,(r)} = \lambda_l\, \hat{\mathbf{y}}_{n_b}^{\neg k,(r)} + (1-\lambda_l)\, \hat{\mathbf{y}}_{n_b}^{\neg k,(r-1)}, \tag{70}$$

where λ_l increases as denoising proceeds. Then, the k^{th} conditional model performs one reverse update using

$$\epsilon_{\theta_k}(\mathbf{x}_{n_b,l}^{k,(r)} | \hat{\mathbf{y}}_{n_b-1}^k, \tilde{\mathbf{y}}_{n_b,l}^{\neg k,(r)}, \mathbf{s}). \tag{71}$$

As in M2PDE, the physical fields are updated sequentially within each diffusion step and later fields use the newest available estimate of fields updated earlier in the same step. The original

paper describes this sequential refinement and the outer-loop estimate blending as the core M2PDE inference procedure. After completion of the reverse process, the generated block becomes the history for the next temporal transition $\hat{\mathbf{y}}^k_{n_b} \rightarrow \mathbf{y}^k_{n_b}$ and this is used to generate the future block $\hat{\mathbf{y}}^k_{n_b+1}$ until the requested time horizon is reached.

D.1.4 M2PDE-AR COMPOSITION ALGORITHM

The algorithm is presented in Algorithm 5.

Algorithm 5 M2PDE-AR long-horizon block composition

Require: Conditional models $\{p^{W,\mathrm{AR}}_{\theta_k}\}_{k=1}^K$, provided first block $\{\mathbf{y}^k_0\}_{k=1}^K$, static conditioning $\mathbf{s}$, block length W, total number of blocks B, outer iterations R, reverse diffusion steps M

$\widehat{\mathbf{y}}^k_0 \leftarrow \mathbf{y}^k_0$ for all k
for $b = 1, \ldots, B-1$ **do**
  Initialize noisy future-block states $\{\mathbf{x}^k_M\}_{k=1}^K$
  for $r = 1, \ldots, R$ **do**
    Store previous outer-loop field estimates
    for reverse diffusion step $\ell = M, \ldots, 1$ **do**
      Compute blending weight λ_l
      for $k = 1, \ldots, K$ **do**
        Form $\widetilde{\mathbf{y}}^{\neg k}_{n_b,\ell}$ from current and previous outer-loop estimates using equation 70
        Update $\mathbf{x}^k_l$ using $p^{W,\mathrm{AR}}_{\theta_k}(\cdot \mid \widehat{\mathbf{y}}^k_{n_{b-1}}, \widetilde{\mathbf{y}}^{\neg k}_{n_b,\ell}, \mathbf{s})$
        Update the clean estimate $\widehat{\mathbf{y}}^k_{n_b} = \frac{x^k_l - \sqrt{1-\bar{\alpha}_l}\epsilon_{\theta_k}}{\sqrt{\bar{\alpha}_l}}$
      end for
    end for
  end for
  Append $\{\widehat{\mathbf{y}}^k_{n_b}\}_{k=1}^K$ to the rollout
end for
return $\{\widehat{\mathbf{y}}^k_{0:T}\}_{k=1}^K$

D.1.5 RELATION TO THE ORIGINAL M2PDE

M2PDE-AR does not alter the conditional energy composition principle of M2PDE (Zhang et al., 2025). The adaptation is exclusively temporal in that: (1) each conditional diffusion model generates a W frames block rather than the complete trajectory; (2) the previous target block is introduced as autoregressive conditioning; and (3) the M2PDE compositional reverse process is repeatedly applied to successive future blocks. These points permit M2PDE to operate on substantially longer trajectories that are considered in this work while retaining its original within block multiphysics/multi-field composition rule.

D.2 GENCP-AR: BLOCK AUTOREGRESSIVE EXTENSION OF GENCP

D.2.1 PROBLEM SETUP

GenCP formulates coupled multiphysics generation through flow matching of the conditional/decoupled models (Lipman et al., 2023; Gao et al., 2026). Separately trained conditional velocity fields representing the conditional/decoupled physics problems are composed during inference using operator splitting in the joint product space of the physical fields (Gao et al., 2026). In the GenCP original formulation, the PDE time axis is treated as a monolithic block during the training of the conditional models and the inference time composition. Applying this original formulation directly to long-horizon coupled physics problems becomes computationally prohibitive and also could potentially jeopardize the compositional accuracy. We therefore introduce GenCP-AR, a block autoregressive adaptation of GenCP in which each conditional flow model generates a temporal block of W frames and the generated blocks are recursively composed and propagated along the PDE time.

D.2.2 TRAINING

We index temporal blocks of W frames by b, so that $\mathbf{y}_b^k := y_{bW+1:(b+1)W}^k$ denotes block b of field k and $\mathbf{y}_{b-1}^k$ denotes its immediately preceding target block. For the conditional dataset associated with field k, the future W frames are represented in the joint product space as $\mathbf{Y}_b^{(k)} := (\mathbf{y}_b^k, \bar{\mathbf{y}}_b^{\neg k})$. Here $\mathbf{y}_b^k$ is the field evolved by the corresponding decoupled solver and $\bar{\mathbf{y}}_b^{\neg k}$ denotes the complementary field trajectory prescribed during that solve. GenCP constructs a flow in this product space starting from an independent Gaussian distribution. Specifically, we draw $\mathbf{Z}_b^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with the same dimensions as $\mathbf{Y}_b^{(k)}$. For the generative flow time $\tau \in [0, 1]$, we define

$$\mathbf{X}_{b,\tau}^{(k)} = (1 - \tau)\mathbf{Z}_b^{(k)} + \tau \mathbf{Y}_b^{(k)}. \tag{72}$$

The corresponding flow velocity is

$$\mathbf{v}_b^{(k)} = \partial_\tau \mathbf{X}_{b,\tau}^{(k)} = \mathbf{Y}_b^{(k)} - \mathbf{Z}_b^{(k)}, \tag{73}$$

and the component of $\mathbf{v}_b^{(k)}$ associated with field k is

$$\mathbf{v}_b^k := \mathbf{y}_b^k - \mathbf{z}_b^k. \tag{74}$$

The conditional GenCP model for field k receives the complete product space $\mathbf{X}_{b,\tau}^{(k)}$ and also the preceding target block $\mathbf{y}_{b-1}^k$ and static conditioning $\mathbf{s}$. The model predicts the velocity component associated with field k

$$\widehat{\mathbf{v}}_{\theta_k}\left(\mathbf{X}_{b,\tau}^{(k)}, \tau | \mathbf{y}_{b-1}^k, \mathbf{s}\right) \approx \mathbf{v}_b^k. \tag{75}$$

Hence, each model learns how one physical field evolves along the generative flow while observing the complete joint product space. The preceding target block is used as a clean autoregressive history.

Although the joint product space is interpolated from the Gaussian reference $\mathbf{Z}_b^{(k)}$ to decoupled state $\mathbf{Y}_b^{(k)}$ as in equation 72, the k^{th} conditional model is supervised only on the velocity component associated with field k. Thus, the training objective is

$$\mathcal{L}_k = \mathbb{E}\left[\left\|\mathbf{v}_b^k - \widehat{\mathbf{v}}_{\theta_k}\left(\mathbf{X}_{b,\tau}^{(k)}, \tau | \mathbf{y}_{b-1}^k, \mathbf{s}\right)\right\|_2^2\right]. \tag{76}$$

Therefore, the complementary fields participate in the common product space but are not prediction targets of the k^{th} model. Moreover, the preceding target block is a clean conditioning variable.

D.2.3 LONG-HORIZON GENCP-AR COMPOSITION

Let $b = 0, \ldots, B-1$ denote the temporal block index, where block b spans frames $bW + 1$ to $(b+1)W$. The first coupled block $\{\mathbf{y}_0^k\}_{k=1}^K$ is provided exactly. For each subsequent block $b \geq 1$, GenCP-AR initializes the joint product space from an independent Gaussian $\mathbf{X}_{b,0} = \mathbf{Z}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The previously generated block $\widehat{\mathbf{y}}_{b-1}^k$ is used as autoregressive history for model k. We define the lifted joint product space vector field as

$$\widehat{\mathbf{V}}_{\theta_k}(\mathbf{X}_{b,\tau}, \tau | \widehat{\mathbf{y}}_{b-1}^k, \mathbf{s}) := \left(0, \ldots, \widehat{\mathbf{v}}_{\theta_k}(\mathbf{X}_{b,\tau}, \tau | \widehat{\mathbf{y}}_{b-1}^k, \mathbf{s}), \ldots, 0\right), \tag{77}$$

so that only the coordinate associated with field k is advanced by this sub-flow.

For a flow step $\Delta\tau = 1/M$, Lie-Trotter composition updates the field-wise conditional vector fields sequentially. For example, considering a two-field case, let $\mathbf{X}_{b,m}$ denote the joint product space at $\tau_m = m\Delta\tau$. The first field is updated as

$$\mathbf{X}_{b,m+1/2} = \mathbf{X}_{b,m} + \Delta\tau\, \widehat{\mathbf{V}}_{\theta_1}(\mathbf{X}_{b,m}, \tau_m | \widehat{\mathbf{y}}_{b-1}^1, \mathbf{s}) \tag{78}$$

followed by

$$\mathbf{X}_{b,m+1} = \mathbf{X}_{b,m+1/2} + \Delta\tau\, \widehat{\mathbf{V}}_{\theta_2}(\mathbf{X}_{b,m+1/2}, \tau_m | \widehat{\mathbf{y}}_{b-1}^2, \mathbf{s}). \tag{79}$$

Note that the second conditional vector field acts on the state updated by the first field during the same flow step. This sequential update is performed for $m = 0, \ldots, M-1$, thereby advancing the joint product space from $\tau = 0$ to $\tau = 1$.

After the final flow step, the resulting state defines the next coupled temporal block

$$\widehat{\mathbf{Y}}_b := \mathbf{X}_{b,1} = (\widehat{\mathbf{y}}_b^1, \dots, \widehat{\mathbf{y}}_b^K). \tag{80}$$

This generated block is appended to the PDE time rollout, and each component $\widehat{\mathbf{y}}_b^k$ becomes the autoregressive history used by the model k to generate the next block.

D.2.4 GenCP-AR composition algorithm

The algorithm is presented in Algorithm 6.

Algorithm 6 GenCP-AR long-horizon block composition

Require: Conditional vector fields $\{\widehat{\mathbf{v}}_{\theta_k}\}_{k=1}^K$, provided first block $\{\mathbf{y}_0^k\}_{k=1}^K$, static conditioning $\mathbf{s}$, block length W, number of flow steps M, total number of blocks B

$\widehat{\mathbf{y}}_0^k \leftarrow \mathbf{y}_0^k$ for all k
$\Delta\tau \leftarrow 1/M$
for $b = 1, \dots, B-1$ **do**
  Sample a new joint product space block $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $m = 0, \dots, M-1$ **do**
    $\tau_m \leftarrow m\Delta\tau$
    for $k = 1, \dots, K$ **do**
      Evaluate $\mathbf{v}^k \leftarrow \widehat{\mathbf{v}}_{\theta_k}\left(\mathbf{x}, \tau_m \mid \widehat{\mathbf{y}}_{b-1}^k, \mathbf{s}\right)$
      Update only field k: $\mathbf{x}^k \leftarrow \mathbf{x}^k + \Delta\tau\, \mathbf{v}^k$
    end for
  end for
  $\widehat{\mathbf{Y}}_b \leftarrow \mathbf{x}$
  Append $\widehat{\mathbf{Y}}_b = (\widehat{\mathbf{y}}_b^1, \dots, \widehat{\mathbf{y}}_b^K)$ to the rollout
end for
return $\{\widehat{\mathbf{y}}_b^k\}_{k=1,\, b=0}^{K,\, B-1}$

Note: In equation 77, since $\widehat{\mathbf{V}}_{\theta_k}$ is non-zero only in coordinate k, the update in equation 78 reduces to advancing $\mathbf{x}^k$ only. This algorithm the component-wise form.

D.2.5 Relation to the original GenCP

GenCP-AR preserves the conditional flow matching and operator splitting principles of GenCP (Gao et al., 2026). At the algorithmic level, the adaptation is temporal in that: (1) each conditional flow model operates on a block of PDE time frames rather than generating the complete long-horizon trajectory; (2) the immediately preceding target block is introduced as clean autoregressive conditioning; and (3) the original GenCP joint product space flow and field-wise operator splitting are restarted for each successive future block.

D.3 FNO-AR-Joint oracle

FNO-AR-Joint is a deterministic autoregressive Fourier neural operator (FNO Li et al. (2021)) baseline trained directly on coupled trajectories. It follows the temporal bundling and pushforward training principles of Brandstetter et al. (2022). Temporal bundling predicts a block of future frames in a single model evaluation. Pushforward training introduces autoregressively generated histories as conditioning to the model during its training. These techniques reduce distribution shift and error accumulation in autoregressive predictions.

Let $\mathbf{y}_t = (y_t^1, \dots, y_t^K)$ the coupled state at time frame t and F_h is the number of history frames. Given the most recent history $\mathbf{y}_{n-F_h:n}$ and static conditioning $\mathbf{s}$, FNO-AR-Joint predicts the future frames

$$\hat{\mathbf{y}}_{n:n+F_h} = \mathcal{F}_\phi(\mathbf{y}_{n-F_h:n}, \mathbf{s}). \tag{81}$$

A single FNO-AR-Joint model predicts all the coupled fields jointly and is trained directly on coupled trajectories. In our implementation, the static conditioning fields $\mathbf{s}$ are repeated across the history window and supplied alongside the autoregressive state.

Training minimizes the loss

$$\mathcal{L} = \|\mathcal{F}_\phi(\mathbf{y}_{n-F_h:n}, \mathbf{s}) - \mathbf{y}_{n:n+F_h}\|_2^2. \tag{82}$$

Following the pushforward strategy of Brandstetter et al. (2022), additional training samples are constructed in which the first predicted block is detached and inserted into the subsequent autoregressive history before predicting the next block. This treats autoregressive instability as a distribution shift between ground truth histories observed during training and model generated histories encountered during rollout.

At inference, the first F_h coupled frames are provided exactly. The model predicts the next F_h frames jointly, appends them to the trajectory, retains the most recent F_h frames as the next history, and repeats.

$$\mathbf{y}_{n-F_h:n} \to \hat{\mathbf{y}}_{n:n+F_h} \to \dots \tag{83}$$

This deterministic block rollout is performed until the requested horizon is reached and the final block is truncated if needed.

E PDE Case Studies

This appendix specifies the four coupled PDE systems used in the Experiments section, in order of increasing coupling complexity. For each system, we state the coupled equations, the conditional subproblems used to generate the decoupled training data, and the numerical solver.

E.1 FitzHugh-Nagumo system

We consider the one-dimensional FitzHugh-Nagumo reaction-diffusion system (Klaasen & Troy (1984))

$$\begin{aligned} \partial_t u &= D_u \partial_{xx} u + u - u^3 - k - \alpha v \\ \partial_t v &= D_v \partial_{xx} v + \beta u - v \end{aligned} \tag{84}$$

on the spatial domain $x \in [-1, 1]$ and time domain $t \in (0, 2]$. The two fields are bidirectionally and linearly coupled: v enters $\partial_t u$ through the term αv, and u enters $\partial_t v$ through the term βu, with α and β controlling the coupling strength between u and v. This parameterization follows the FitzHugh-Nagumo setup in PDEBench (Takamoto et al. (2022)), but restricted to one spatial dimension.

The coupled reference trajectories are obtained by integrating both equations simultaneously from initial states (u_0, v_0). To construct the decoupled conditional datasets used to train M2PDE, M2PDE-AR, GenCP-AR, and ChronoSync, we additionally solve each equation while prescribing the trajectory of the other field. Specifically, the $u|v$ trajectories are generated by integrating

$$\partial_t u = D_u \partial_{xx} u + u - u^3 - k - \alpha v_{\text{pres}}(x, t) \tag{85}$$

where v_{pres} is fixed to the supplied conditioning trajectory. Similarly, the $v|u$ trajectories are generated from

$$\partial_t v = D_v \partial_{xx} v + \beta u_{\text{pres}}(x, t) - v \tag{86}$$

where u_{pres} is fixed to the supplied conditioning trajectory. Each conditional trajectory evolves a single field in response to a prescribed physical field, whereas the coupled reference evolves the mutually interacting (u, v) system. The initial state (u_0, v_0) is generated randomly using a Gaussian random field (GRF) in one spatial dimension, and the prescribed fields $u_{\text{pres}}(x, t)$ and $v_{\text{pres}}(x, t)$ are sampled from a GRF over space and time.

The Laplacian is discretized using a second-order central difference with (periodic) circular padding: $\partial_{xx} f_i \approx (f_{i+1} - 2f_i + f_{i-1})/\Delta x^2$. Time integration uses the classical fourth-order Runge-Kutta (RK4) method. When the output interval exceeds the stable explicit integration time step, each output interval is subdivided according to a CFL stability estimate. All reported trajectories contain 101 output frames over a physical duration of $T = 2.0$. The solver parameters are summarized in Table 2.

Table 2: FitzHugh-Nagumo system solver parameters

Parameter	Symbol	Value
Spatial domain	Ω	$[-1, 1]$
Spatial resolution	N_x	128
Boundary condition	–	Periodic
Output frames	N_t	101
Simulation duration	T	2.0
Diffusivity of u	D_u	1×10^{-3}
Diffusivity of v	D_v	5×10^{-3}
Reaction threshold	k	5×10^{-3}
Coupling parameter	α	1.0
Coupling parameter	β	1.0
Time integrator	–	RK4
Spatial discretization	–	Second-order finite difference
CFL safety factor	–	0.4

E.2 Viscous Burgers system

We consider a one-dimensional coupled viscous Burgers system consisting of a velocity-like field $v(x, t)$ and a transport scalar $c(x, t)$

$$\begin{aligned} \partial_t v &= -\partial_x(v^2) + \nu \partial_{xx} v + \beta(c - c_{\text{ref}}) \\ \partial_t c &= -\partial_x(vc) + D\partial_{xx} c - \lambda c \end{aligned} \tag{87}$$

on the spatial domain $x \in [0, 1]$ and time domain $t \in (0, 0.0512]$. The velocity drives the scalar nonlinearly through the advective flux $\partial_x(vc)$, and the scalar feeds back into the velocity equation through the forcing term $\beta(c - c_{\text{ref}})$. The $v \to c$ transport is the dominant coupling direction, and we therefore report errors on c in the main text. We set $\lambda = 0$ and $c_{\text{ref}} = 0$ throughout.

The coupled reference trajectories are obtained by integrating both equations simultaneously from the initial state (v_0, c_0). To construct the decoupled conditional datasets used to train M2PDE, M2PDE-AR, GenCP-AR, and ChronoSync, the $v|c$ trajectories are generated by integrating

$$\partial_t v = -\partial_x(v^2) + \nu \partial_{xx} v + \beta(c_{\text{pres}}(x, t) - c_{\text{ref}}) \tag{88}$$

where c_{pres} is fixed to the supplied conditioning trajectory. Similarly, the $c|v$ trajectories are generated from

$$\partial_t c = -\partial_x(v_{\text{pres}}(x, t)c) + D\partial_{xx} c - \lambda c \tag{89}$$

where v_{pres} is fixed to the supplied conditioning trajectory. As for FitzHugh-Nagumo, each conditional trajectory evolves a single field in response to a prescribed partner field, whereas the coupled reference evolves the mutually interacting (v, c) system. The initial state (v_0, c_0) is sampled from a GRF in one spatial dimension, and the prescribed fields $v_{\text{pres}}(x, t)$ and $c_{\text{pres}}(x, t)$ are sampled from a GRF over space and time.

The nonlinear advective terms are discretized using a conservative first-order upwind numerical flux, and the Laplacian using a second-order central difference; periodic boundary conditions are imposed in both cases. In the coupled solve, each time step first advances c using the current (c_n, v_n) state and then advances v using the updated c_{n+1}, i.e., $c_{n+1} = c_n + \Delta t\, \mathcal{F}_c(c_n, v_n)$ followed by $v_{n+1} = v_n + \Delta t\, \mathcal{F}_v(v_n, c_{n+1})$, where $\mathcal{F}_c$ and $\mathcal{F}_v$ denote the discretized right-hand sides of equation 87. The solver parameters are summarized in Table 3.

E.3 Darcy transport system

We consider a two-dimensional Darcy transport system on the rectangular domain $\Omega = [0, 2L] \times [0, L]$, with $L = 150$, based on MultiPhysicsBench (Yang et al., 2026). Let $c(\mathbf{x}, t)$ denote the solute concentration, $p(\mathbf{x}, t)$ the pressure, and $\rho(c) = \rho_0 + \beta c$ the concentration-dependent fluid density.

Table 3: Viscous Burgers system solver parameters

Parameter	Symbol	Value
Spatial domain	Ω	$[0, 1]$
Spatial resolution	N_x	256
Boundary condition	–	Periodic
Output frames	N_t	512
Time step	Δt	1×10^{-4}
Simulation duration	T	5.12×10^{-2}
Viscosity	ν	1×10^{-2}
Diffusivity	D	1×10^{-2}
Decay	λ	0
Coupling strength	β	1
Reference concentration	c_{ref}	0

The Darcy transport system is defined by

$$\begin{aligned} &\mathbf{u} = -\frac{\kappa}{\mu}\left[\nabla p - \rho(c)\mathbf{g}\right] \\ &\nabla \cdot [\rho(c)\mathbf{u}] = 0 \\ &\partial_t(\theta_s c) + \mathbf{u} \cdot \nabla c - \nabla \cdot (\theta_s \tau D_L \nabla c) = S_c(\mathbf{x}) \end{aligned} \tag{90}$$

where κ is the permeability, μ the dynamic viscosity, $\mathbf{g}$ the gravitational acceleration vector, θ_s the porosity, τD_L the effective dispersion coefficient, and $S_c(\mathbf{x})$ the concentration source. We define the pressure anomaly relative to the reference hydrostatic pressure as $q := p + \rho_0 g y$. The concentration c controls the density-driven buoyancy forcing, q represents the pressure response required by the mass-continuity constraint, and the velocity $\mathbf{u}$ is recovered deterministically from (q, c) through Darcy's law. Hence, q and c constitute a sufficient pair of coupled fields, and M2PDE-AR, GenCP-AR, and ChronoSync compose the pressure anomaly and concentration fields ($q \leftrightarrow c$).

The coupled trajectories are obtained by solving equation 90 directly. They are used to train only the coupled-data oracles, ERDM-Joint and FNO-AR-Joint, and to evaluate all methods.

Following Zhang et al. (2025) and Gao et al. (2026), the decoupled conditional datasets used to train M2PDE-AR, GenCP-AR, and ChronoSync are constructed via windowed pre-iteration. We start from a prescribed concentration field $c_{\mathrm{pres}}(\mathbf{x}, t)$ sampled from a GRF and partition the simulation horizon into pre-iteration windows of 16 frames (distinct from the model window length W). Within the first window, we solve the first two equations of equation 90 for $(q, \mathbf{u})$ given $c_{\mathrm{pres}}(\mathbf{x}, t)$, yielding $q|c$ in that window. Using the obtained $q|c$, we then solve the third equation of equation 90 for c, yielding $c|q$. This procedure is repeated window by window until the end of the simulation. When moving to a new window, the prescribed concentration is linearly blended with the concentration from the previous window to ensure temporal continuity. This windowing limits long-horizon drift, keeping the conditional trajectories closer to the support of physically admissible solutions while remaining decoupled from the coupled dynamics.

The concentration source is a Gaussian

$$S_c(x, y) = \frac{A}{T} \exp\left[-\frac{(x - x_0)^2}{2s_x^2} - \frac{(y - y_0)^2}{2s_y^2}\right] \tag{91}$$

with amplitude A, center (x_0, y_0), widths (s_x, s_y) and simulation duration T. We impose $c = 0$ on $y = 0$ and $c = c_s$ on $y = L$ for $x \in [L, 2L]$. The governing equations are solved using the Multiphysics Object Oriented Simulation Environment (MOOSE)(Harbour et al., 2025) with first-order Lagrange finite elements for q and c on a structured 64×64 quadrilateral mesh. Temporal integration uses the second-order backward differentiation (BDF2). The solver parameters are summarized in Table 4.

E.4 Rayleigh-Bénard convection system

We consider two-dimensional Rayleigh-Bénard convection under the Boussinesq approximation on the rectangular domain $\Omega = [0, L_x] \times [0, L_y]$. Let $\mathbf{u} = (u_x, u_y)$ denote the velocity field, p the

Table 4: Darcy transport system solver parameters

Parameter	Symbol	Value
Spatial domain	Ω	$[0, 300] \times [0, 150]$
Spatial resolution	(N_x, N_y)	$(64, 64)$
Output frames	N_t	300
Simulation duration	T	6.31152×10^8 s (20 years)
Output time interval	Δt	2.110876×10^6 s
Reference density	ρ_0	1000
Density-concentration coefficient	β	200
Porosity	θ_s	0.1
Effective dispersion	τD_L	3.56×10^{-6}
Dynamic viscosity	μ	1×10^{-3}
Permeability	κ	4.845×10^{-13}
Gravitational acceleration	g	9.81
Boundary concentration	c_s	1.0
Source amplitude	A	0.10
Gaussian center	(x_0, y_0)	$(150, 75)$
Gaussian widths	(s_x, s_y)	$(22.5, 45)$

pressure, and T the temperature. In the non-dimensional form, the governing equations are (Drazin & Reid, 2004; Kontolati et al., 2024)

$$\begin{aligned} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} \nabla^2 \mathbf{u} + (T - T_{\mathrm{lin}})\mathbf{e}_y \\ \nabla \cdot \mathbf{u} &= 0 \\ \partial_t T + \mathbf{u} \cdot \nabla T &= \frac{1}{\sqrt{\mathrm{RaPr}}} \nabla^2 T \end{aligned} \tag{92}$$

where Ra and Pr denote the Rayleigh and Prandtl numbers, respectively, $\mathbf{e}_y$ is the vertical unit vector, and

$$T_{\mathrm{lin}}(y) = T_{\mathrm{hot}} + (T_{\mathrm{cold}} - T_{\mathrm{hot}}) \frac{y}{L_y} \tag{93}$$

is the conductive temperature profile.

We impose periodic boundary conditions in the horizontal direction and no-slip velocity boundary conditions at $y = 0$ and $y = L_y$. The temperature is fixed to $T_{\mathrm{hot}} = 1$ at the lower wall and $T_{\mathrm{cold}} = 0$ at the upper wall. The velocity and temperature fields are bidirectionally coupled: temperature fluctuations generate buoyancy in the momentum equation, and the resulting velocity field advects temperature.

The coupled reference trajectories are obtained by solving all three equations in equation 92 simultaneously. The coupled trajectories are generated separately and are not used to construct the conditional training data. To construct the decoupled conditional datasets, we solve two physical subproblems. For $\mathbf{u}|T$, a temperature trajectory $T_{\mathrm{pres}}(\mathbf{x}, t)$ is prescribed, and the forced incompressible momentum and continuity equations are solved for $\mathbf{u}$. Conversely, for $T|\mathbf{u}$, a velocity trajectory $\mathbf{u}_{\mathrm{pres}}(\mathbf{x}, t)$ is prescribed and the advection-diffusion equation is solved for T.

As in the Darcy transport case, the conditional trajectories are constructed using windowed pre-iteration, with pre-iteration windows of 16 frames and a one-frame overlap between consecutive windows. For each realization, $T_{\mathrm{pres}}(\mathbf{x}, t)$ is sampled from a GRF that satisfies the boundary conditions. Within the first window, we solve the forced incompressible momentum and continuity equations for $\mathbf{u}$ given $T_{\mathrm{pres}}(\mathbf{x}, t)$, yielding $\mathbf{u}|T$ in that window. Using the obtained $\mathbf{u}$, we then solve the advection-diffusion equation in equation 92 for T, yielding $T|\mathbf{u}$. This procedure is repeated for each window until the end of the simulation time. When moving to a new window, the prescribed temperature is linearly blended with the temperature from the previous window to ensure temporal continuity.

The governing equations are solved using Dedalus 3 (Burns et al., 2020). The horizontal direction is represented using a real Fourier basis and the vertical direction using a Chebyshev basis, with a $3/2$

de-aliasing factor. The no-slip and fixed-temperature wall boundary conditions are imposed using the Dedalus tau formulation. We use the 64×64 spatial discretization for our experiments in this work. The solver parameters are presented in Table 5.

Table 5: Rayleigh-Bénard convection solver parameters

Parameter	Symbol	Value
Spatial domain	Ω	$[0, 4] \times [0, 1]$
Spatial resolution	(N_x, N_z)	$(64, 64)$
Output frames	N_t	300
Simulation duration	T	21.0
Solver time step	Δt_{sol}	1×10^{-3}
Rayleigh number	Ra	1×10^{6}
Prandtl number	Pr	1.0
Kinematic viscosity	ν	1×10^{-3}
Thermal diffusivity	κ_T	1×10^{-3}
Bottom temperature	T_{hot}	1.0
Top temperature	T_{cold}	0.0
Horizontal boundary	–	Periodic
Vertical velocity boundary	–	No-slip
Vertical temperature boundary	–	Dirichlet
Spatial discretization	–	Fourier-Chebyshev spectral
De-aliasing factor	–	$3/2$

F Data Generation

Table 6 summarizes the number of conditional and coupled trajectories used for training and testing in each case study. For every case study, the decoupled-data methods and the coupled-data oracles are trained on the same number of trajectories, and all methods are evaluated on the same held-out coupled test trajectories. For the two-dimensional Darcy transport and Rayleigh-Bénard convection systems, Table 7 additionally reports the wall-clock cost of generating one coupled trajectory and each of the two corresponding conditional trajectories.

Table 6: Number of trajectories used for training and testing. The conditional training trajectories are generated independently of the coupled dynamics (Appendix E whereas the coupled trajectories are used only to train the coupled-data oracles and to evaluate all methods.

	Training trajectories		Test
Case study	Conditional	Coupled	Coupled
FitzHugh-Nagumo	10,000	10,000	50
Viscous Burgers	10,000	10,000	50
Darcy transport	4,800 (1 pre-iteration)	4,800	24
Rayleigh-Bénard	4,329 (1 pre-iteration)	4,329	16

Table 7: Wall-clock time (in seconds) to generate one trajectory for the two-dimensional case studies. The two conditional columns correspond to the two conditional subproblems of each system.

Case study	Coupled	Conditional 1	Conditional 2
Darcy transport	130.070	24.977 ($c|q$)	21.013 ($q|c$)
Rayleigh-Bénard	102.206	56.732 ($T|\mathbf{u}$)	76.297 ($\mathbf{u}|T$)

G BACKBONE ARCHITECTURES

Across all benchmark methods, we use high-capacity backbones chosen to provide broadly comparable model complexity and representational power while preserving the architectural structure of each method. Since the underlying model families differ, we do not enforce exact parameter matching; instead, we align the overall width, depth, multiscale capacity, and spatio-temporal modeling capability where possible. Throughout, enc_i and dec_i denote the i-th encoder and decoder stages from high to low resolution, and mid denotes the bottleneck.

G.1 CHRONOSYNC, M2PDE, M2PDE-AR, AND ERDM-JOINT

ChronoSync, M2PDE, M2PDE-AR, and ERDM-Joint share the same spatio-temporal U-Net backbone family, with the input and output channels adapted to each case study. The backbone consists of three encoder stages, a bottleneck, and three mirrored decoder stages with skip connections. Residual blocks use GroupNorm, SiLU activations, 3×3 convolutions, and FiLM modulation(Perez et al., 2018), which conditions the feature maps on the diffusion-time embedding. Spatial resolution is reduced only along the physical spatial dimensions, while the PDE time dimension is preserved throughout the encoder-decoder hierarchy. Thus, downsampling operates along X for the one-dimensional spatial systems and along (X, Y) for the two-dimensional spatial systems.

For the one-dimensional FitzHugh-Nagumo and Burgers systems, we use a base width of 128 channels with stage multipliers $(1, 2, 4, 8)$. The encoder contains $(4, 4, 4)$ residual blocks across its three stages, the bottleneck contains two residual blocks, and the decoder mirrors the encoder with $(4, 4, 4)$ blocks. We use a dropout of 0.05. Dense spatial attention (Vaswani et al., 2017) is used at all attention stages, with 4 heads of dimension 64 and a maximum token budget of $32,768$.

For the two-dimensional Darcy and Rayleigh-Bénard convection systems, we use a base width of 64 channels and the same stage multipliers $(1, 2, 4, 8)$. The encoder contains $(3, 3, 6)$ residual blocks, the bottleneck contains six residual blocks, and the decoder contains $(6, 3, 3)$ residual blocks from low to high resolution. We use a dropout rate of 0.05 throughout the network. At the high-resolution stages $\{\mathrm{enc}_1, \mathrm{enc}_2, \mathrm{dec}_2, \mathrm{dec}_1\}$, spatial interactions are modeled using Transolver-style slice attention Wu et al. (2024) with 128 learned slices of dimension 64. Dense spatial self-attention is used at $\{\mathrm{enc}_3, \mathrm{mid}, \mathrm{dec}_3\}$, again with 4 heads of dimension 64 and a maximum token budget of $32,768$.

Temporal self-attention is applied along the PDE time dimension at fixed spatial locations at $\{\mathrm{enc}_3, \mathrm{mid}, \mathrm{dec}_3\}$ with a maximum token budget of $8,388,608$. ChronoSync and ERDM-Joint use causal temporal attention, so that a feature at a given PDE time step attends only to the same or preceding time steps. M2PDE and M2PDE-AR use bidirectional temporal attention, which attends across the complete temporal block. The U-Net configurations are summarized in Table 8.

Table 8: U-Net backbone configurations used for ChronoSync, M2PDE, M2PDE-AR, and ERDM-Joint on all the case studies.

Configuration	**FHN/Burgers**	**Darcy/RBC**
Base channels	128	64
Stage multipliers	$(1, 2, 4, 8)$	$(1, 2, 4, 8)$
Feature widths	$(128, 256, 512, 1024)$	$(64, 128, 256, 512)$
Encoder blocks	$(4, 4, 4)$	$(3, 3, 6)$
Bottleneck blocks	2	6
Decoder blocks	$(4, 4, 4)$	$(6, 3, 3)$
Dropout	0.05	0.05
Attention heads	4	4
Attention head dim.	64	64
Transolver stages	None	enc_1, enc_2, dec_2, dec_1
Number of slices	–	128
Slice dimension	–	64
Dense spatial attention	All stages	enc_3, mid, dec_3
Dense spatial token limit	$32,768$	$32,768$
Temporal attention stages	enc_3, mid, dec_3	enc_3, mid, dec_3
Temporal token limit	$8,388,608$	$8,388,608$

G.2 GENCP-AR

Following the original GenCP ((Raonic et al., 2023)), GenCP-AR uses a Convolutional Neural Operator (CNO) backbone for each conditional flow model. The backbone follows an operator U-Net structure with a lifting layer, 4 encoder and decoder levels, skip connections between matching resolutions, residual blocks at each resolution, a deeper residual bottleneck, and a final projection layer. The flow time τ is encoded using a sinusoidal embedding of dimension 128 and injected throughout the CNO using FiLM modulation. We use one residual block per encoder resolution and 6 residual blocks at the bottleneck, and the input and output channels are adapted to each case study.

For the one-dimensional FitzHugh-Nagumo and Burgers systems, we use 4 CNO levels with channel multiplier 32, giving feature widths $(16, 32, 64, 128, 256)$. Dense spatial attention is used at $\{\mathrm{enc}_1, \mathrm{enc}_2, \mathrm{enc}_3, \mathrm{mid}, \mathrm{dec}_3, \mathrm{dec}_2, \mathrm{dec}_1\}$, and dense temporal attention at $\{\mathrm{enc}_3, \mathrm{mid}, \mathrm{dec}_3\}$. Both attention mechanisms use 4 heads with head dimension 32, with spatial and temporal token limits $32,768$ and $8,388,608$ respectively. Temporal attention is causal along the PDE time dimension, and no Transolver-style spatial attention is used for these systems.

For the two-dimensional Darcy and Rayleigh-Bénard convection systems, we use the same 4-level CNO hierarchy with channel multiplier 64, giving feature widths $(32, 64, 128, 256, 512)$. At the high-resolution stages $\{\mathrm{enc}_1, \mathrm{enc}_2, \mathrm{dec}_2, \mathrm{dec}_1\}$, spatial attention is replaced by Transolver-style slice attention with 64 learned slices of dimension 32; dense spatial and temporal attention are applied at $\{\mathrm{enc}_3, \mathrm{mid}, \mathrm{dec}_3\}$. All attention modules use 4 heads of dimension 32, with the same token limits as above. The CNO configurations are summarized in Table 9.

Table 9: CNO backbone configurations used by GenCP-AR.

Configuration	**FHN/Burgers**	**Darcy/RBC**
CNO levels	4	4
Channel multiplier	32	64
Feature widths	$(16, 32, 64, 128, 256)$	$(32, 64, 128, 256, 512)$
Residual blocks per level	1	1
Bottleneck residual blocks	6	6
Transolver stages	None	enc_1, enc_2, dec_2, dec_1
Attention heads	4	4
Attention head dim.	32	32
Number of slices	–	64
Slice dimension	–	32
Dense spatial attention	All stages	enc_3, mid, dec_3
Dense spatial token limit	$32,768$	$32,768$
Temporal attention stages	enc_3, mid, dec_3	enc_3, mid, dec_3
Temporal token limit	$8,388,608$	$8,388,608$

G.3 FNO-AR-JOINT

FNO-AR-Joint is the deterministic autoregressive Fourier neural operator described in Appendix D.3, trained directly on coupled trajectories with temporal bundling and pushforward training (Brandstetter et al., 2022). At each autoregressive step, the model receives the preceding F_{h} PDE time frames of all the physical fields and jointly predicts the next F_{h} frames, which are appended to the rollout and used as history for the next autoregressive step. For each case study, F_{h} equals the window length W used by the other methods. The FNO backbone differs between the one- and two-dimensional systems, as described below.

For FitzHugh-Nagumo and Burgers, we use a one-dimensional FNO over the spatial dimension. The autoregressive history of all fields is flattened into the channel dimension and concatenated with the normalized spatial coordinate. The lifted representation is processed by 4 Fourier layers and projected to jointly predict all physical fields over the next F_{h} frames.

For Darcy and Rayleigh-Bénard convection, we use a two-dimensional FNO augmented with a temporal encoder. The autoregressive field histories and case-specific conditioning variables are first encoded along PDE time by a lightweight temporal Conv1D network. The resulting temporal features are flattened into channels, concatenated with normalized spatial coordinates, and processed by

4 two-dimensional Fourier blocks, each combining global spectral mixing with local and pointwise spatial operations. The final projection jointly predicts all physical fields over the next F_{h} frames. The FNO architecture configurations are summarized in Table 10.

Table 10: FNO-AR-Joint backbone configurations.

Configuration	**FHN/Burgers**	**Darcy/RBC**
Spatial operator	FNO1D	FNO2D
Latent FNO width	128	128
Fourier layers	4	4
Retained modes	16	16×16
Coordinate channels	x	(x, y)
Explicit temporal encoder	No	Yes
Temporal encoder width	–	32
Temporal encoder layers	–	2
Temporal kernel size	–	3
Temporal MLP ratio	–	2
FNO local convolution	–	3×3
FNO channel MLP ratio	–	4
FNO dropout	–	0.05
Output projection expansion	$4\times$	$4\times$

H Training and Inference Settings

H.1 Training settings

Table 11 reports the training settings for ChronoSync and all baselines. For a given case study, all methods use the same temporal window length W and the same number of training trajectories (Table 6). ChronoSync and ERDM-Joint share the ERDM training procedure of Section 3.2. ChronoSync trains one conditional ERDM per field on the decoupled data, whereas ERDM-Joint trains a single ERDM on the coupled data.

H.2 Inference settings

Inference settings used for the reported results are summarized in Table 12. Unless otherwise stated, rolling methods use the same temporal window at training and inference. The first W frames are provided as initial context, after which predictions are generated autoregressively over the remaining PDE time horizon.

I Additional Results

This appendix complements the Experiments section with qualitative comparisons for all four case studies, rollout errors over PDE time the for one-dimensional systems, the field-wise conditional-to-oracle gap recovered by each method, and additional details on the out-of-distribution time-extrapolation test for Rayleigh-Bénard convection. Throughout, M2PDE-AR, GenCP-AR, and ChronoSync are trained only on the decoupled conditional trajectories, whereas ERDM-Joint and FNO-AR-Joint are coupled-data oracles trained directly on coupled trajectories.

I.1 Spatio-temporal plots

Figures 4 and 5 compare representative spatio-temporal trajectories of two test samples for the FitzHugh-Nagumo and Burgers systems, respectively, across the decoupled data methods and coupled-data oracles, with the coupled reference shown for comparison.

Figures 7, 8, 9, and 10 show spatial snapshots of the pressure anomaly q and concentration c at selected PDE times for two representative Darcy test samples. Figures 11, 12, 13, and 14 show spatial snapshots of the temperature T and the resultant velocity magnitude $|\mathbf{u}|$ at selected PDE times for two representative Rayleigh-Bénard convection test samples. Each figure compares the

Table 11: Training settings for all methods. When a setting differs across case studies, values are reported in the order FitzHugh-Nagumo / Burgers / Darcy / Rayleigh-Bénard; a single value indicates that the same setting is used for all case studies. The effective batch size includes data-parallel workers and gradient accumulation.

Training setting	**ChronoSync/ERDM-Joint**	**M2PDE/M2PDE-AR**	**GenCP-AR**	**FNO-AR-Joint**
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	1×10^{-4}	1×10^{-4}	2×10^{-4}	1×10^{-3}
Weight decay	1×10^{-4}	1×10^{-4}	Default GenCP	1×10^{-4}
Effective batch size	32	32	32/32/16/16	32
Precision	bf16-mixed	bf16-mixed	bf16-mixed	bf16-mixed
LR schedule	Cosine	Cosine	Cosine	Cosine
Temporal window/history	16/50/10/55	16/50/10/55 (only M2PDE-AR)	16/50/10/55	16/50/10/55
Prediction block size	16/50/10/55	16/50/10/55 (only M2PDE-AR)	16/50/10/55	16/50/10/55
Diffusion/flow steps	Continuous τ^*	1000 DDPM	10 flow	–
Pushforward probability	–	–	–	0.5

*For ChronoSync and ERDM-Joint, diffusion time is sampled as $\tau \sim \mathbb{U}[0, 1)$ during training (Section 3.2), and noise-schedule settings follow Rühling Cachay et al. (2026): $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = -10$, $\sigma_{\text{data}} = 1$, $P_{\text{mean}} = 0.5$, and $P_{\text{std}} = 1.2$

Table 12: Inference settings used for ChronoSync and the benchmark methods. When values vary across case studies, they are reported in the order FitzHugh-Nagumo / Burgers / Darcy / Rayleigh-Bénard.

Inference setting	**ChronoSync/ERDM-Joint**	**M2PDE/M2PDE-AR**	**GenCP-AR**	**FNO-AR-Joint**
Sampler	Heun	DDPM	Flow solver	Deterministic
Sampling steps	32	1000	10	–
Window length W	16/50/10/55	16/50/10/55 for M2PDE-AR	16/50/10/55	16/50/10/55
Provided initial context	W	W for M2PDE-AR	W	W
κ	0.25	–	–	–
Bootstrap rollouts $n_{\text{boot}}^{\text{roll}}$	1	–	–	–

coupled reference, conditional trajectories, decoupled-data methods (including ChronoSync) and coupled-data oracles.

I.2 Rollout errors over PDE time: FitzHugh-Nagumo and Burgers

Figure 6 reports the relative ℓ_2 error as a function of PDE time for FitzHugh-Nagumo and Burgers, complementing the Darcy and Rayleigh-Bénard rollout errors in Figure 2

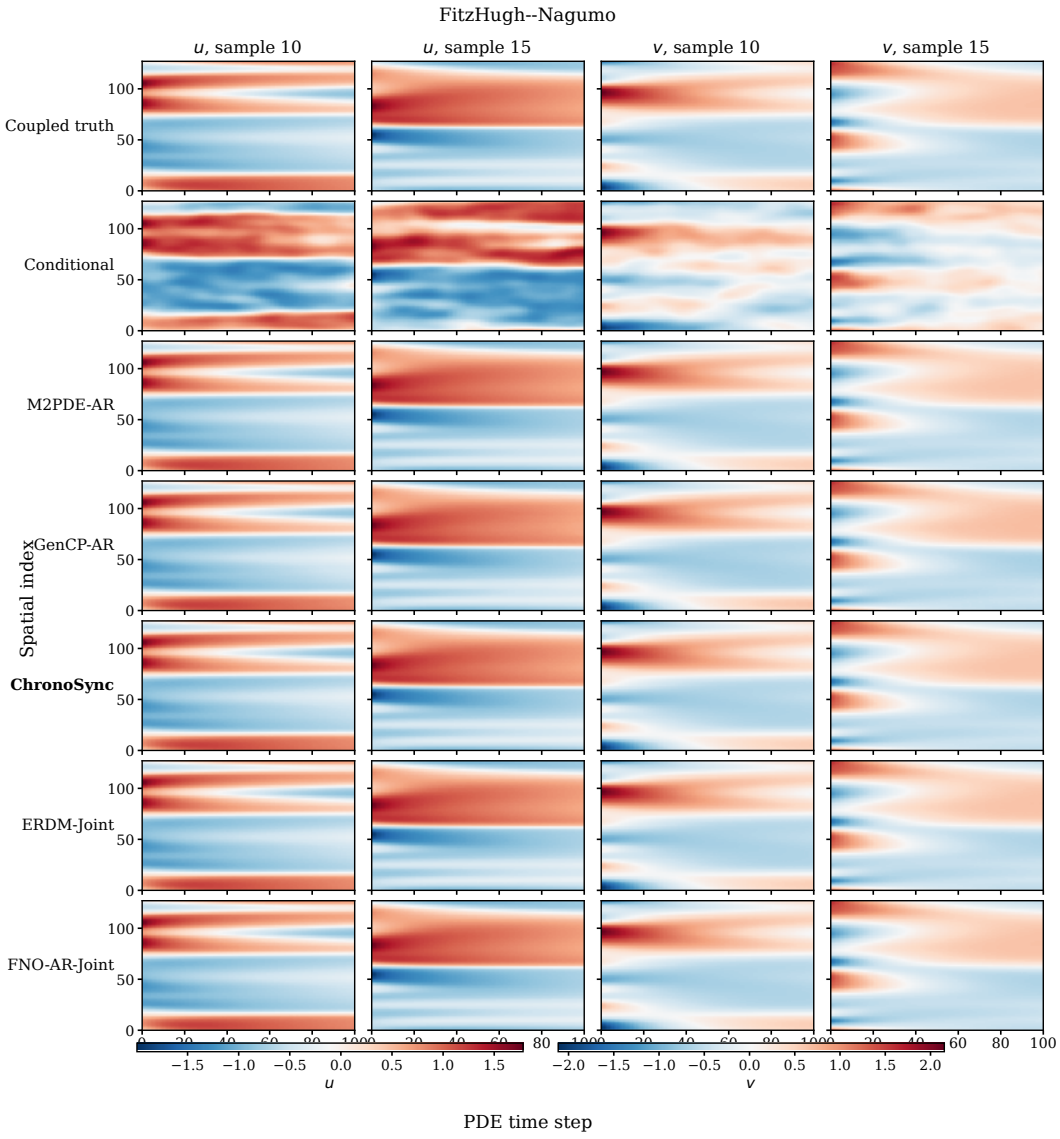


Figure 4: Spatio-temporal comparison for the FitzHugh-Nagumo system on two representative test samples. Rows show the coupled reference, conditional trajectories, M2PDE-AR, GenCP-AR, ChronoSync, ERDM-Joint, and FNO-AR-Joint. Columns show the two state variables u and v.

I.3 Conditional-to-oracle gap recovered

Table 13 reports, for each field, the percentage of the error gap between the conditional trajectories and FNO-AR-Joint that each method recovers, computed from the relative ℓ_2 errors in Table 1. Averaged over all seven fields, ChronoSync recovers over 90% of this gap.

I.4 OOD testing for time extrapolation: Rayleigh-Bénard convection

We provide additional details on the time- extrapolation summarized in Section 4.5, which examines whether the recovered coupled dynamics remain coherent beyond the PDE time range seen during training. For Rayleigh-Bénard convection, we extend inference from the 300-frame training horizon to $t = 399$ and compare ChronoSync with FNO-AR-Joint, the strongest coupled-data oracle. As shown in Figure 15(a), ChronoSync preserves the temperature plume and velocity structures beyond the training horizon. Over $t \in [300, 399]$, FNO-AR-Joint obtains temperature and velocity relative

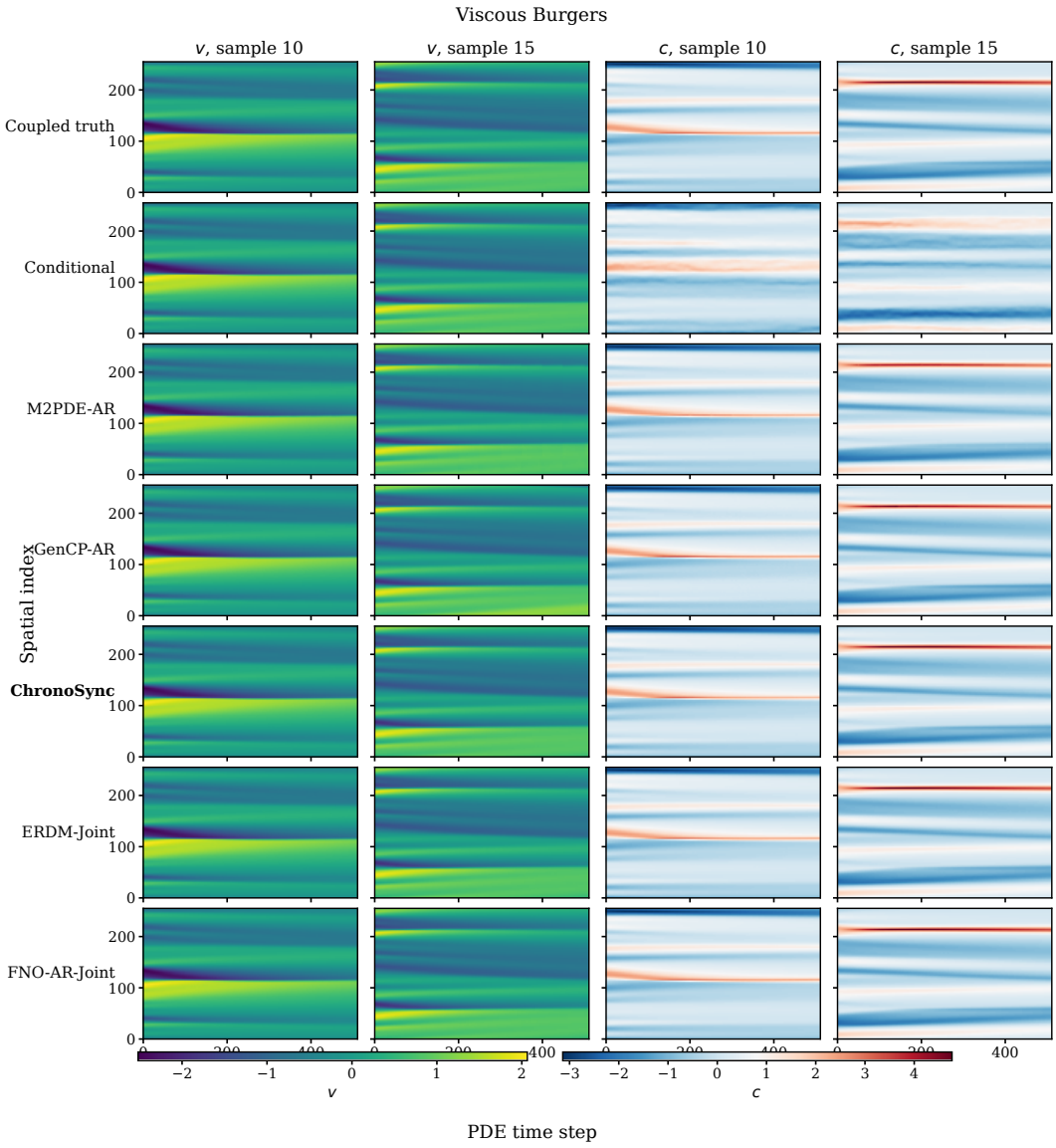


Figure 5: Spatio-temporal comparison for the viscous Burgers system on two representative test samples. Rows show the coupled reference, conditional trajectories, M2PDE-AR, GenCP-AR, ChronoSync, ERDM-Joint, and FNO-AR-Joint. Columns show the velocity field v and concentration field c.

ℓ_2 errors of 0.089 and 0.345, respectively, whereas ChronoSync obtains 0.113 and 0.322. As shown in Figure 15(b), the corresponding median Nusselt number RMSE with respect to the coupled reference is 0.64 for FNO-AR-Joint and 0.72 for ChronoSync. ChronoSync therefore remains close to the model trained directly on coupled data during temporal extrapolation, despite never observing coupled Rayleigh–Bénard trajectories during training.

J Ablations for ChronoSync

We examine the sensitivity of ChronoSync to the relaxation parameter κ, the rolling-window length W, and the number of bootstrap rollouts $n_{\text{boot}}^{\text{roll}}$.

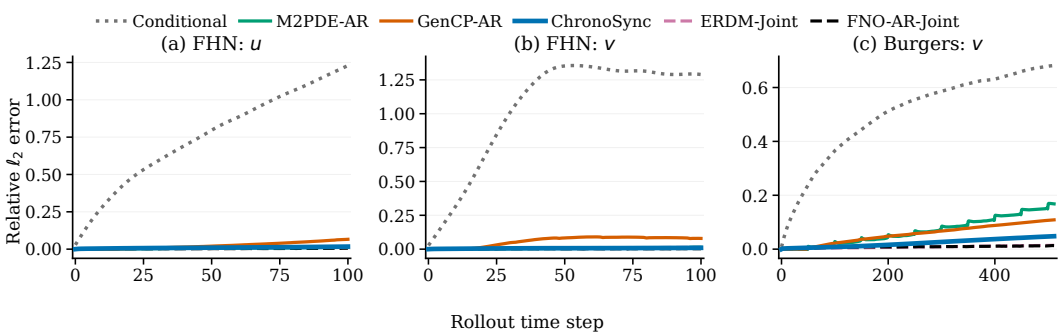


Figure 6: **Long-horizon rollout errors for FitzHugh-Nagumo and Burgers systems.** Relative ℓ_2 error against the coupled reference as a function of PDE time for (a) FitzHugh-Nagumo u, (b) FitzHugh-Nagumo v, and (c) Burgers c. ChronoSync, M2PDE-AR, and GenCP-AR are trained only on decoupled conditional trajectories, whose errors are also shown, whereas ERDM-Joint and FNO-AR-Joint are trained directly on coupled trajectories.

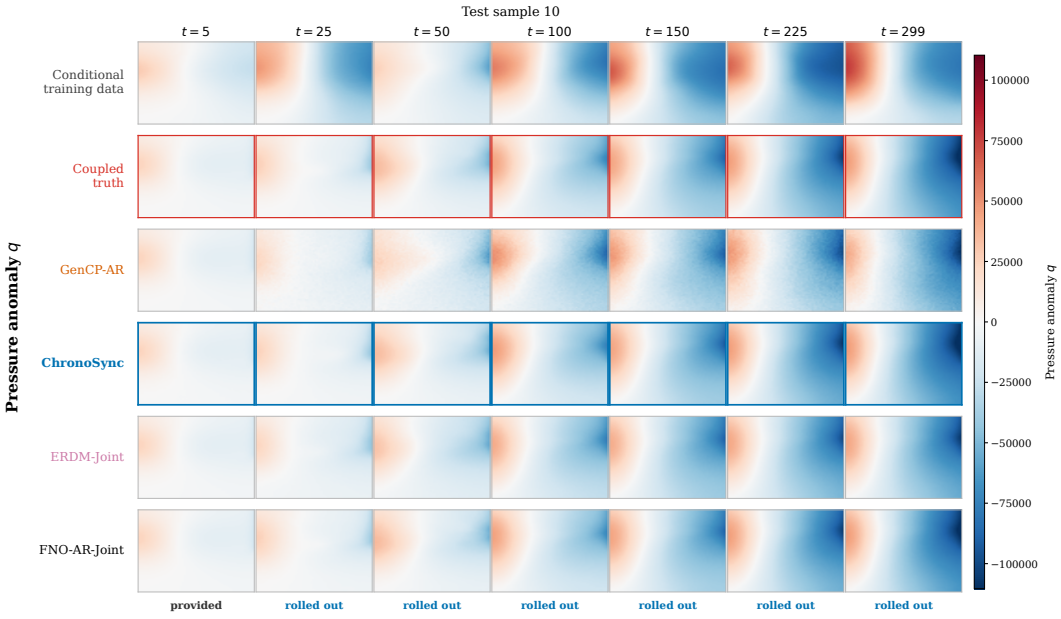


Figure 7: Darcy transport system, test sample 10: pressure anomaly q at selected PDE times.

J.1 FitzHugh-Nagumo

Figures 16, 17, and 18 show the resulting relative ℓ_2 errors for both state variables.

J.2 Viscous Burgers

Figures 19, 20, and 21 show the resulting relative ℓ_2 errors for both state variables.

J.3 Darcy transport

Darcy transport exhibits stronger sensitivity to the temporal window length W. Figure 22 shows the relative ℓ_2 errors for q and c across $W \in \{10, 25, 48\}$. ChronoSync performs best at the shorter window $W = 10$, whereas the coupled-data oracles generally benefit from larger windows. This highlights that the preferred temporal locality is method dependent.

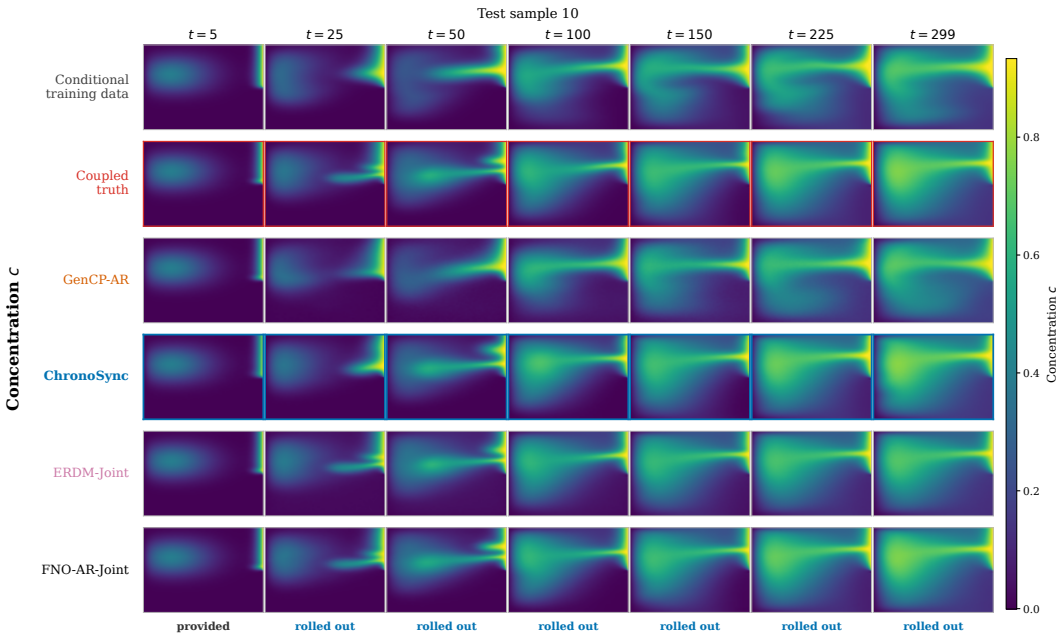


Figure 8: Darcy transport system, test sample 10: concentration c at selected PDE times.

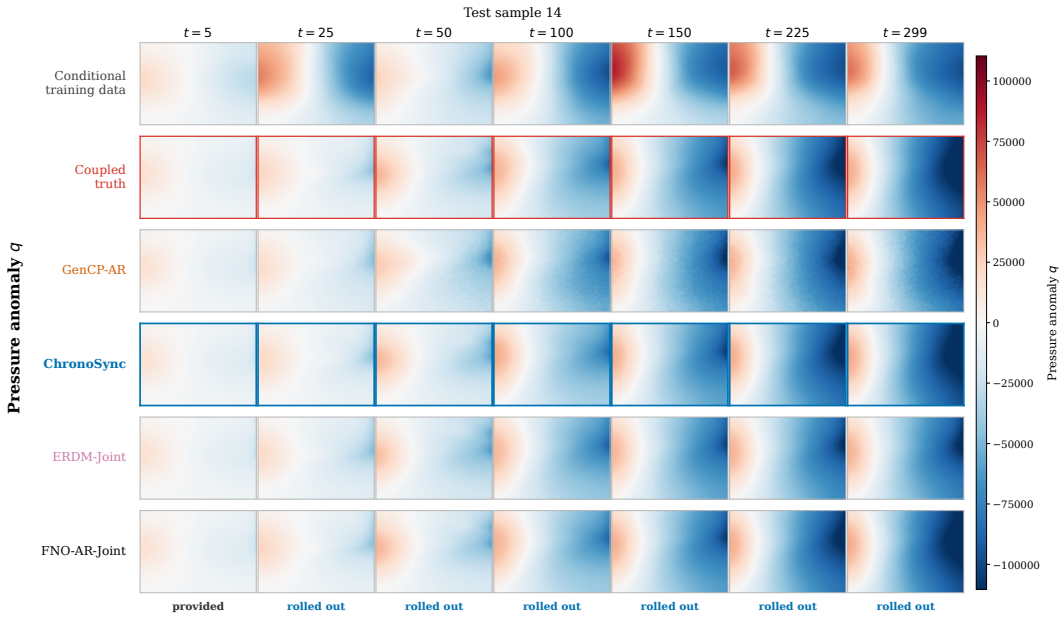


Figure 9: Darcy transport system, test sample 14: pressure anomaly q at selected PDE times.

J.4 Rayleigh–Bénard convection

We examine the sensitivity of ChronoSync to the rolling window length W for the Rayleigh-Bénard convection system. Figure 23 reports the relative ℓ_2 errors for velocity and temperature over $W \in \{10, 25, 55, 75\}$.

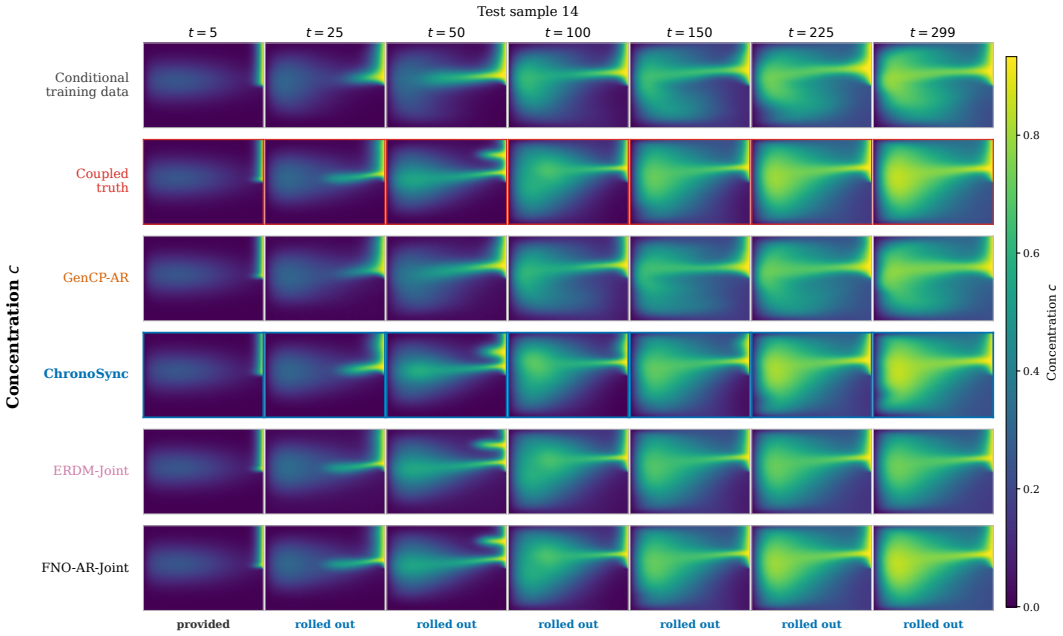


Figure 10: Darcy transport system, test sample 14: concentration c at selected PDE times.

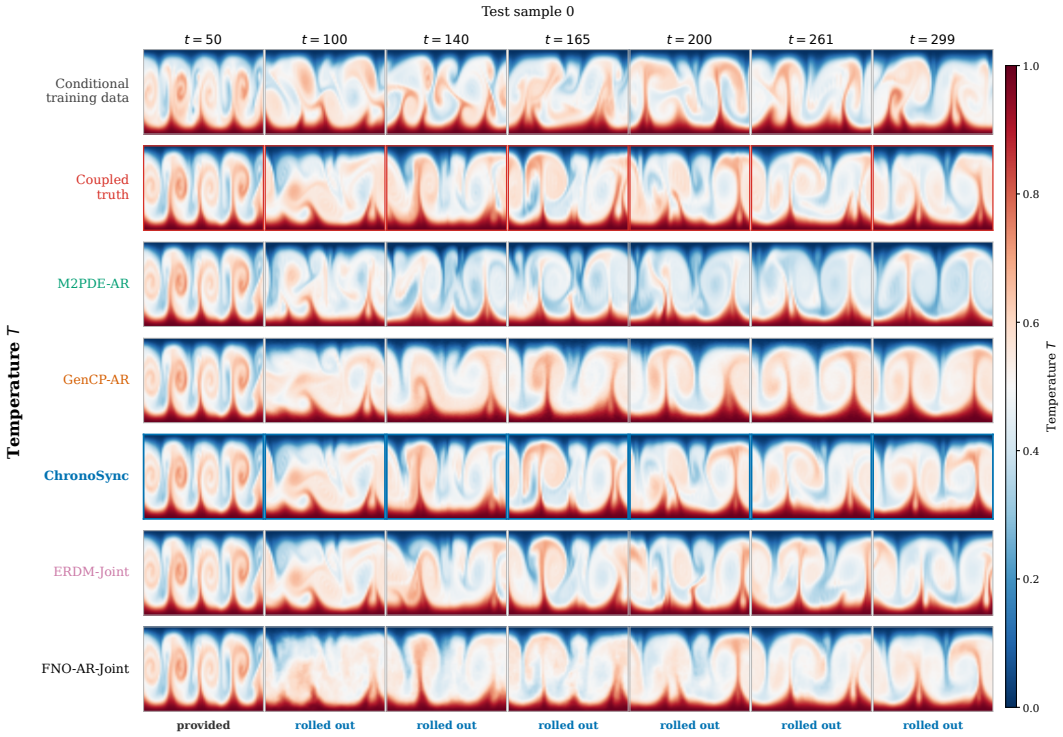


Figure 11: Rayleigh-Bénard convection, test sample 0: temperature T at selected PDE times.

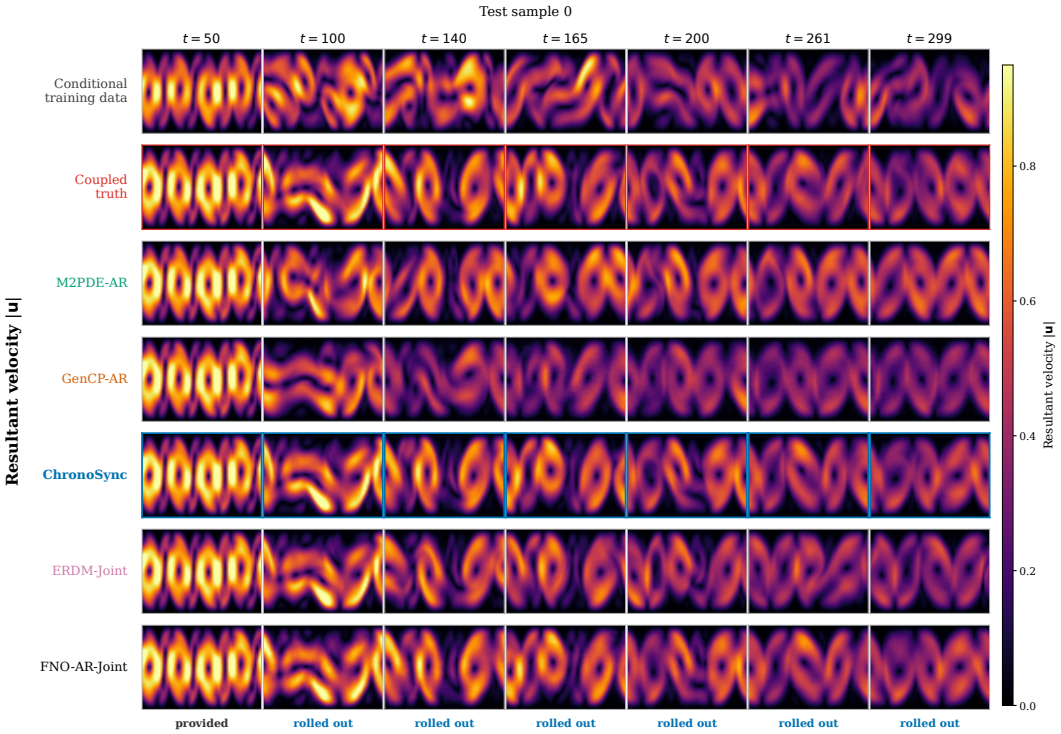


Figure 12: Rayleigh-Bénard convection, test sample 0: resultant velocity magnitude $|\mathbf{u}|$ at selected PDE times.

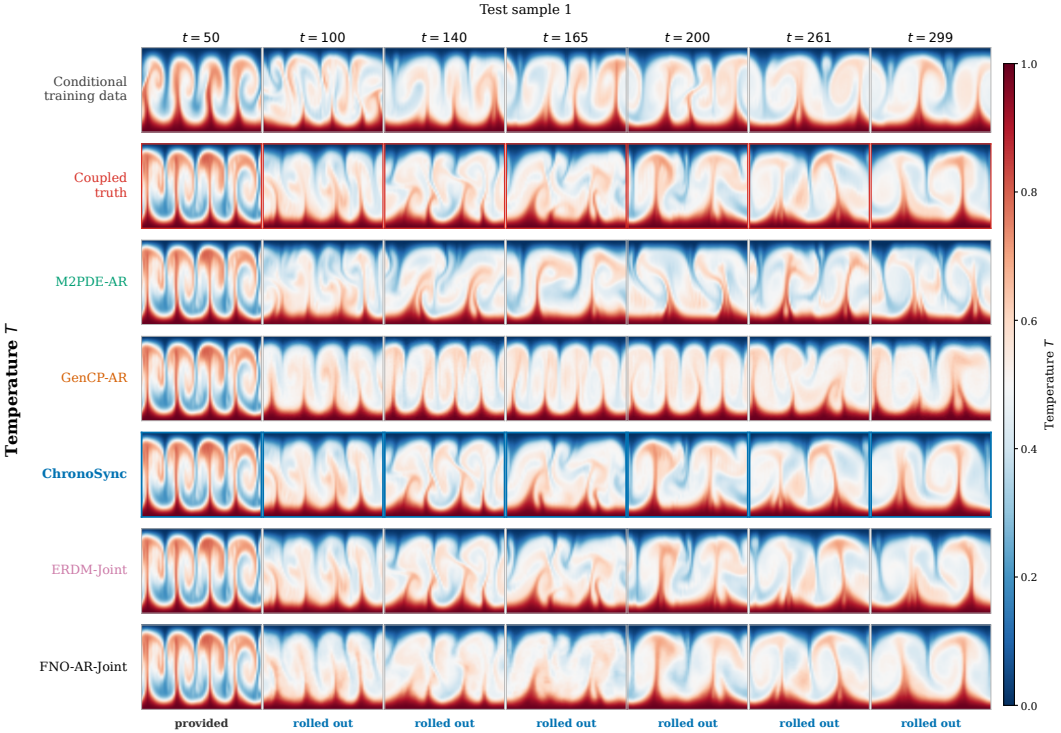


Figure 13: Rayleigh-Bénard convection, test sample 1: temperature T at selected PDE times.

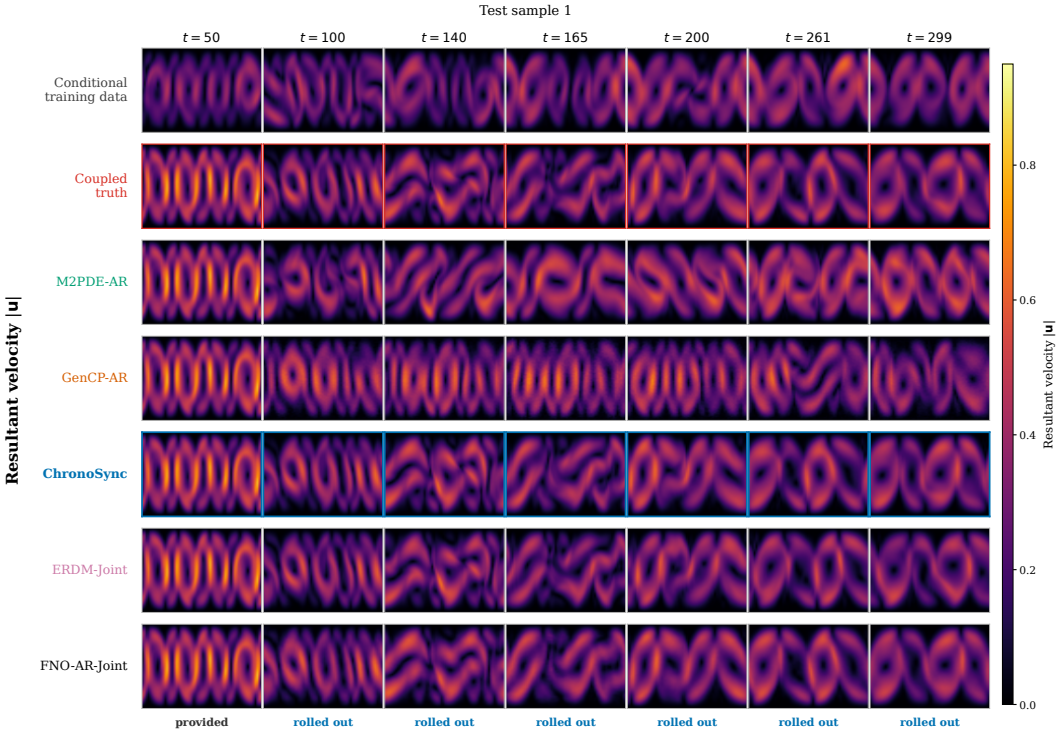


Figure 14: Rayleigh-Bénard convection, test sample 1: resultant velocity magnitude $|\mathbf{u}|$ at selected PDE times.

Table 13: Percentage of the conditional-to-coupled error gap recovered for each physical field (higher is better). Conditional trajectories define 0%, while FNO-AR-Joint defines 100%. Values are not clipped; negative values indicate performance worse than the conditional trajectories, and values above 100% indicate performance exceeding FNO-AR-Joint. Bold denotes the best method trained without coupled trajectories, and italic indicates the second best.

		Increasing complexity $\longrightarrow$						
		FitzHugh-Nagumo		Burgers	Darcy transport		Rayleigh-Bénard	
Training data	Method	u	v	c	c	q	T	(u_x, u_y)
Decoupled	Conditional trajectories	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	M2PDE	98.6	99.2	75.2	OOM†	OOM†	OOM†	OOM†
	M2PDE-AR*	**99.4**	*99.6*	81.4	–	–	18.8	26.1
	GenCP-AR*	96.7	94.3	*88.1*	*4.7*	*63.8*	*35.8*	*37.4*
	ChronoSync (Ours)	*99.1*	**99.7**	**96.5**	**70.8**	**94.7**	**91.9**	**90.5**
Coupled	ERDM-Joint	99.4	99.5	100.0	48.0	69.4	57.8	62.8
	FNO-AR-Joint	100.0	100.0	100.0	100.0	100.0	100.0	100.0

For method m, field f, the recovered gap is $G_{m,f} = 100\frac{e_{\text{cond},f} - e_{m,f}}{e_{\text{cond},f} - e_{\text{FNO-AR-Joint},f}}$, where e denotes the relative ℓ_2 error.
$*$: Autoregressive extensions of M2PDE and GenCP to handle long time horizons.
$\dagger$: Out of GPU memory when processing all 300 time frames jointly.
$-$: M2PDE-AR diverges on Darcy transport (errors exceed those of the conditional trajectories).

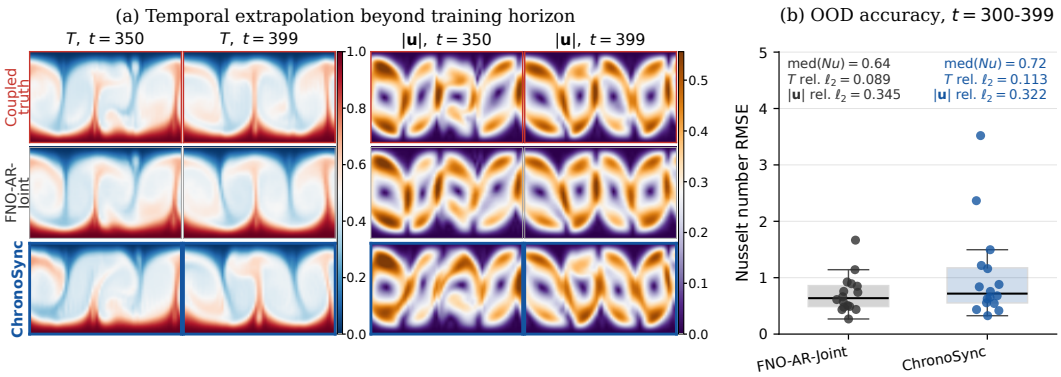


Figure 15: **Temporal extrapolation beyond the Rayleigh–Bénard training horizon.** (a) Coupled reference, FNO-AR-Joint, and ChronoSync predictions of temperature T and resultant velocity $|\mathbf{u}|$ at $t = 350$ and $t = 399$, beyond the 300 frames training horizon. (b) Out-of-distribution accuracy over $t \in [300, 399]$, reported as Nusselt-number RMSE and relative ℓ_2 errors for temperature and velocities. ChronoSync remains close to the coupled data FNO-AR-Joint oracle despite being trained only on decoupled conditional trajectories.

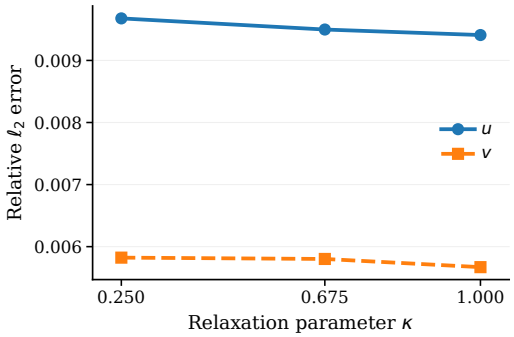


Figure 16: Sensitivity of ChronoSync to the relaxation parameter κ for the FitzHugh-Nagumo system.

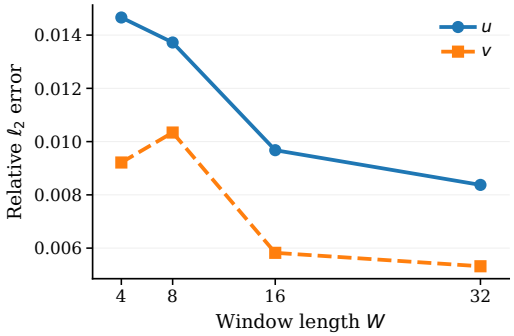


Figure 17: Sensitivity of ChronoSync to the rolling window length W for the FitzHugh-Nagumo system.

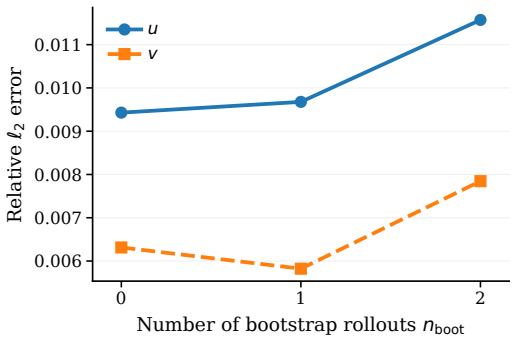


Figure 18: Sensitivity of ChronoSync to the number of bootstrap rollouts $n_{\text{boot}}^{\text{roll}}$ for the FitzHugh-Nagumo system.

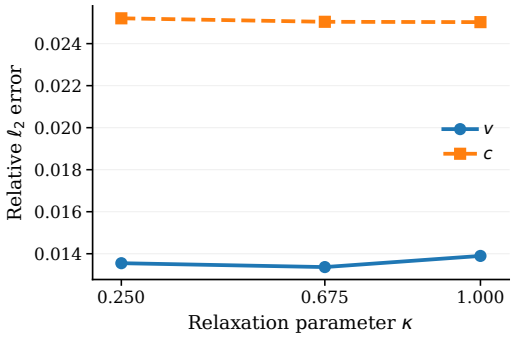


Figure 19: Sensitivity of ChronoSync to the relaxation parameter κ for the Burgers system.

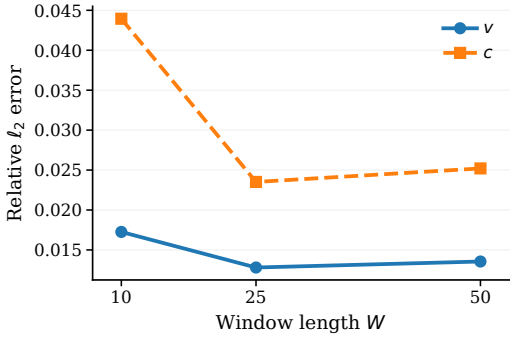


Figure 20: Sensitivity of ChronoSync to the rolling window length W for the Burgers system.

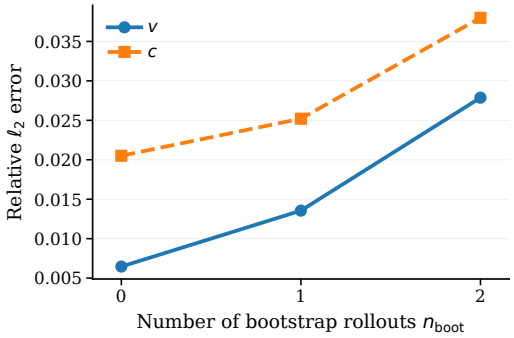


Figure 21: Sensitivity of ChronoSync to the number of bootstrap rollouts $n_{\text{boot}}^{\text{roll}}$ for the Burgers system.

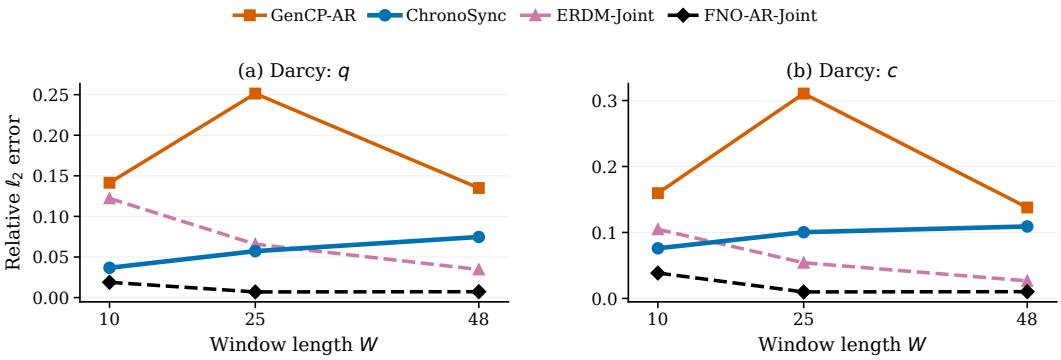


Figure 22: Sensitivity to temporal window length W for Darcy transport. Relative ℓ_2 errors are shown for (a) pressure anomaly q and (b) concentration c.

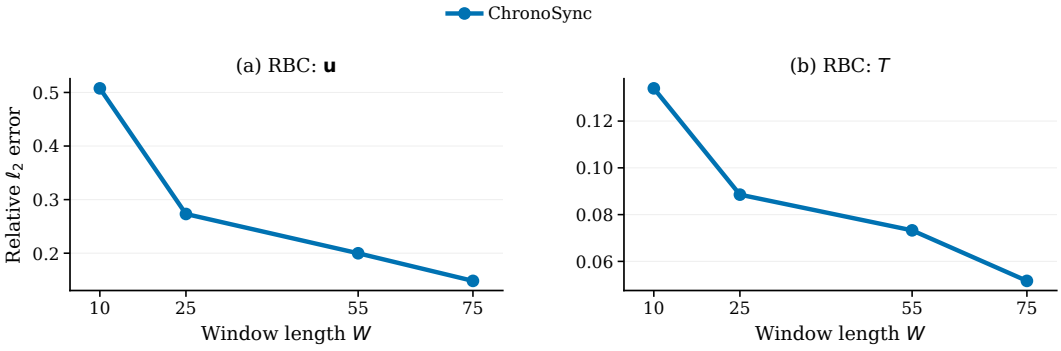


Figure 23: Sensitivity of ChronoSync to the rolling window length W for Rayleigh-Bénard convection system. Relative ℓ_2 errors are shown for (a) combined velocity $\mathbf{u}$ and (b) temperature T.