StreamPhy: Streaming Inference of High-Dimensional Physical Dynamics via State Space Models

Panqi Chen[1] **Yifan Sun**[1] **Shikai Fang**[1] **Xiao Fu**[2] **Lei Cheng**[1] *
[1] College of Information Science and Electronic Engineering, Zhejiang University
[2] Department of ECE, University of Minnesota - Twin Cities

Abstract

Inferring the evolution of high-dimensional and multi-modal (e.g., spatio-temporal) physical fields from irregular sparse measurements in real time is a fundamental challenge in science and engineering. Existing approaches, including diffusion-based generative models and functional tensor methods, typically operate in offline settings, depend on full temporal observations, or incur substantial inference cost. We propose StreamPhy, an end-to-end framework that enables efficient and accurate streaming inference of full-field physical dynamics from incoming irregular sparse measurements. To realize this, StreamPhy introduces a data-adaptive observation encoder that is robust to arbitrary observation patterns, leverages a structured state-space model for memory-efficient online updates across irregular time intervals, and develops an expressive Functional Tensor Feature-wise Linear Modulation (FT-FiLM) decoder for continuous-field generation. We further establish theoretically that FT-FiLM is more expressive than the functional Tucker model, admitting a richer function class for representing complex physical dynamics. Experiments on three representative physical systems under challenging sampling patterns show that StreamPhy consistently outperforms state-of-the-art baselines, with at least 48% improvement in accuracy and up to 20–100$\times$ faster inference than diffusion-based methods.

1 Introduction

The real-time inference of spatiotemporal physical fields from streaming observations is of fundamental importance across a wide range of scientific and engineering applications. In applications such as environmental sensing [1], fluid dynamics [2], and structural health monitoring [3], observations are often sparse, irregular, and acquired under time-varying measurement patterns. These characteristics pose significant challenges for building models that can efficiently assimilate incoming data, maintain temporal consistency, and produce accurate full-field reconstructions.

Recent years have witnessed growing recognition that modern artificial intelligence (AI) techniques hold strong promise for accurately modeling spatiotemporal physical fields. One line of work [4, 5, 6, 7] draws inspiration from two-stage diffusion paradigms. Specifically, these methods first encode observations into a latent space using architectures such as vision transformers [7] or joint embedding predictive architectures [8], and then learn a generative model over the latent variables. During inference, techniques such as diffusion posterior sampling (DPS) [9] are employed to incorporate observations. Despite their strong empirical performance, these approaches typically rely on vectorized or matrix-form representations, making them readily compatible with off-the-shelf AI models (e.g., CNNs [10] and Transformers [11]) but less effective at explicitly capturing the intrinsic spatiotemporal structure of physical fields. Another line of work adopts a tensor decomposition perspective, often combined with functional representation learners such as implicit

*Correspond to lei_cheng@zju.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

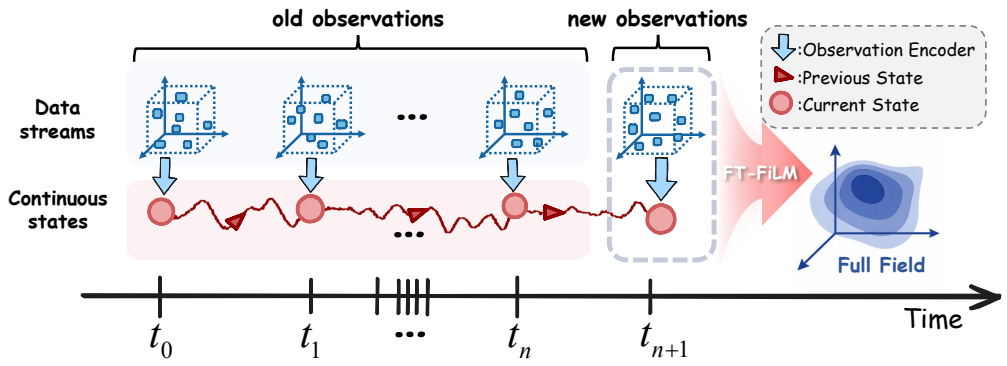

Figure 1: Semantic illustration of the proposed StreamPhy framework.

neural representations (INRs) to accommodate continuous domains [12, 13, 14, 15]. By explicitly modeling the multilinear structure of spatiotemporal data, tensor-based methods are better suited to preserve cross-dimensional relationships inherent in physical systems. Nonetheless, many existing approaches are primarily designed for batch settings rather than streaming scenarios, e.g., [14, 13]. A notable exception is [12], which supports incremental updates with arriving observations, rendering it naturally well-suited for streaming spatiotemporal reconstruction.

These methods all provide viable options for physics field estimation to a good extent, but some limitations remain. For example, diffusion-based models [16, 5, 6] are limited to offline settings and require full temporal observations during training and inference. In addition, their two-stage nature could increase the risk of error propagation. The functional tensor approaches [14, 12, 13] rely on implicit neural representation (INR) [17] and thus only fit to given data,lacking the ability to generalize to future data effectively.

Contributions. To enable streaming inference of continuous spatiotemporal dynamics from sparse and irregular observations, we propose **StreamPhy**, an end-to-end framework built upon HiPPO-based state-space model (SSM) [18, 19, 20]. SSM offer a principled and efficient mechanism for sequential modeling with online updates and long-range dependency capture, while recent advances such as Mamba [21] have demonstrated their strong long-context modeling capability. However, applying SSMs to physical data is nontrivial, as scientific observations are often irregularly sampled in both space and time, and reconstructing continuous spatial fields from latent states demands expressive functional decoding. To address these challenges, StreamPhy integrates three key components: a *data-adaptive observation encoder* that transforms arbitrarily sampled measurements into structured latent representations, a *HiPPO-based SSM* for memory-efficient temporal evolution modeling, and a novel *Functional Tensor Feature-wise Linear Modulation (FT-FiLM) decoder* for reconstructing continuous spatial fields. The proposed FT-FiLM offers greater expressive power than existing functional tensor models through flexible feature-wise modulation, enabling more effective modeling of complex spatiotemporal dynamics. A semantic illustration of StreamPhy is provided in Fig. 1.

We evaluate the proposed method on three representative physical systems with diverse sampling patterns. Experimental results demonstrate that StreamPhy consistently outperforms diffusion-based methods and online tensor approaches by a significant margin in both accuracy and efficiency, achieving at least a 48% improvement in VRMSE over state-of-the-art methods and up to $20 \sim 100\times$ faster inference compared to diffusion-based models. These results highlight the effectiveness of StreamPhy for real-time streaming inference.

Notation: Lower- and upper-case bold letters (e.g., $\mathbf{x}$ and $\mathbf{X}$) denote vectors and matrices, respectively. Upper-case bold calligraphic letters and upper-case calligraphic letters (e.g., $\boldsymbol{\mathcal{X}}$ and $\mathcal{X}$) denote tensors and sets, respectively.

2 Background

2.1 Preliminary

Tensor Decomposition: Consider a K-mode tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_K}$, where each entry $y_{\mathbf{i}}$ is indexed by $\mathbf{i} = (i_1, \ldots, i_K)$ where $i_k \in [I_K]$ for all k. The Tucker decomposition [22] factorizes $\mathcal{Y}$ into mode-wise factor matrices $\{\mathbf{U}^k \in \mathbb{R}^{I_k \times R_k}\}_{k=1}^K$ and a core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$, where R_k denotes the rank of mode k. Each entry admits the multilinear form:$y_{\mathbf{i}} \approx \text{vec}(\mathcal{G})^{\mathrm{T}}(\mathbf{u}^1_{i_1} \otimes \cdots \otimes \mathbf{u}^K_{i_K}) = \sum_{r_1=1}^{R_1} \cdots \sum_{r_K=1}^{R_K} w_{r_1,\ldots,r_K} \prod_{k=1}^K u^k_{i_k,r_k}$, where $\mathbf{u}^k_{i_k} \in \mathbb{R}^{R_k}$ is the latent representation of index i_k in mode k. The CANDECOMP/PARAFAC (CP) decomposition [23] arises as a special case with equal ranks R_k and a superdiagonal core tensor $\mathcal{G}$. Conventional tensors are defined over discrete grids, indexed by $\mathbf{i}$. Functional tensors extend this framework to continuous domains by modeling entries as evaluations of a multivariate function over $\mathbb{R}^K$, rather than discrete grids. Applying the Tucker structure yields the functional Tucker decomposition [14, 15]:

$$y(\mathbf{i}) \approx \text{vec}(\mathcal{G})^{\mathrm{T}}\big(\mathbf{u}^1(i_1) \otimes \cdots \otimes \mathbf{u}^K(i_K)\big), \tag{1}$$

where $\mathbf{i} = (i_1, \ldots, i_K) \in \mathbb{R}^K$ (where each i_k takes continuous values), and $\mathbf{u}^k(\cdot) : \mathbb{R} \to \mathbb{R}^{R_k}$ denotes the latent function for mode k. Such formulations have proven effective for modeling high-dimensional data over continuous coordinates, including climate, turbulent flow, and geospatial datasets [14, 15, 13, 12].

A state space model [20, 19, 18] (SSM) provides a general mathematical framework for modeling recurrent processes via the evolution of a latent state. We note that term "state space model" has been used broadly in prior work [24, 25, 26]; here, we use "SSM" exclusively to denote a class of structured SSMs.

In particular, we consider a continuous-time dynamical system that maps a one-dimensional input $s(t) \in \mathbb{R}$ to an output $\eta(t) \in \mathbb{R}$ through a L-dimensional latent state $\mathbf{x}(t) \in \mathbb{R}^L$, formulated as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}s(t), \\ \eta(t) &= \mathbf{c}^{\mathrm{T}}\mathbf{x}(t) + d \cdot s(t), \end{aligned} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{L \times L}$ governs the state transition dynamics, and $\mathbf{b} \in \mathbb{R}^L, \mathbf{c} \in \mathbb{R}^L, d \in \mathbb{R}$ parameterize the input-to-state and state-to-output mappings. This formulation captures temporal dependencies through latent state evolution in a continuous-time dynamical system, which can be used as a black-box representation in a deep learning model.

Prior works [18, 20, 19] leverage HiPPO theory to enable online compression of continuous signals and discrete-time sequences via projections onto orthogonal polynomial bases, thereby significantly enhancing the modeling of long-range dependencies in Eq. 2. Specifically, this framework prescribes a structured class of matrices $\mathbf{A}$ and $\mathbf{b}$, allowing the latent state $\mathbf{x}(t)$ to effectively retain the history of the input $s(t)$ in an online manner. A prominent instance, HiPPO-LegS, is defined as:

$$\mathbf{A}_{lk} = -\begin{cases} \sqrt{(2l+1)(2k+1)} & \text{if } l > k, \\ l+1 & \text{if } l = k, ; \\ 0 & \text{if } l < k. \end{cases} \quad \mathbf{b}_l = \sqrt{2l+1}. \tag{3}$$

To accommodate discrete input sequences, the continuous-time SSM can be discretized into a recurrence form using the bilinear transform [27]. Given a step size Δt, the resulting discrete-time system is

$$\mathbf{x}_t = \bar{\mathbf{A}}\mathbf{x}_{t-1} + \bar{\mathbf{b}}s_t, \tag{4}$$

$$\eta_t = \mathbf{c}^{\mathrm{T}}\mathbf{x}_t + d \cdot s_t, \tag{5}$$

where $\bar{\mathbf{A}} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}\right)$, $\bar{\mathbf{b}} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1}\Delta t\mathbf{b}$ and $\mathbf{I}$ is the identity matrix. This discretization naturally enables the model to handle irregularly sampled time intervals.

2.2 Problem Setting and Existing Methods

Problem Statement: We denote the K-mode physical full-field tensor at time step t by $\mathcal{Y}_t$, where $\mathbf{i} = (i_1, \cdots, i_K)$ denotes a continuous spatial coordinate. Without loss of generality, we consider a sequence of M observation time steps $\mathcal{T} = \{t_1, \ldots, t_M\}$ with possibly nonuniform intervals, reflecting

realistic asynchronous sampling. At each time t_m, we observe a set $\mathcal{O}_{t_m} = \{(\mathbf{i}_{n_m}, y_{t_m,\mathbf{i}_{n_m}})\}_{n_m=1}^{N_m}$, where $y_{t_m,\mathbf{i}_{n_m}}$ is the entry of $\boldsymbol{\mathcal{Y}}_{t_m}$ at location $\mathbf{i}_{n_m}$ and N_m may vary over time. The observation pattern is therefore time-varying, and we collect the entire data stream as $\mathcal{O} = \{\mathcal{O}_{t_m}\}_{m=1}^{M}$. Unlike prior settings, e.g., those in [16, 13], our streaming setting assumes that observations $\mathcal{O}_{t_m}$ arrive sequentially, requiring online updates based solely on current inputs. The objective is to perform streaming inference of the underlying continuous field $y_t(\mathbf{i}) : \mathbb{R}^K \to \mathbb{R}$ at arbitrary spatial coordinates from this irregular, time-ordered observation stream, which can be viewed as an online function approximation problem.

Existing Methods: Latent diffusion-based methods [5, 16, 6] encode observations into a latent space and learn a generative model, where reconstruction is guided by posterior sampling. However, such a paradigm typically relies on offline training and iterative sampling at inference, limiting its applicability in streaming settings. Within this framework, the functional Tucker model (FTM, see Eq. (1)) serves as an effective latent encoder for high-dimensional data [16]. Nevertheless, FTM relies on multilinear interactions between core tensors and basis functions, often requiring large tensor ranks to achieve sufficient expressivity, which leads to increased memory overhead and slower latent-space operations.

In addition, existing functional tensor-based methods [14, 12, 13, 15] leverage the multilinear structures of FTM for spatiotemporal reconstruction. Despite their effectiveness, these approaches are typically trained in an unsupervised manner on a single evolving tensor sequence, which limits their performance under extreme observations. Moreover, they often rely on high tensor ranks to capture complex dynamics and require memory replay for online updates, thereby reducing practicability. Further discussion of the related work is provided in Appx. D.

2.3 SSM-driven Framework: Motivation and Challenges

Existing paradigms are fundamentally constrained by limited computational efficiency and insufficient long-range memory, thereby limiting their ability to effectively capture long-horizon temporal dependencies. This shortcoming motivates exploring SSMs [20, 18, 19], which offer a principled and memory-efficient framework for sequential representation learning. In particular, the HiPPO framework [18] provides a rigorous mechanism for projecting historical information onto a set of orthogonal basis functions, enabling compact yet memory-preserving latent state representations. Such properties make SSMs especially well-suited for the online approximation of continuous-time dynamical systems.

Despite these advantages, online function approximation from sparse and sequential observations remains inherently challenging. The difficulty stems from the curse of dimensionality, the streaming and partially observed nature of the data, and the intrinsic complexity of spatiotemporal physical processes. Extending conventional SSM formulations to irregular spatiotemporal settings, which are characterized by non-uniform temporal sampling and incomplete spatial coverage, introduces additional challenges. Specifically, two critical issues must be addressed: (1) how to effectively encode irregular observations to consistently update latent states defined over a regular domain; and (2) how to construct a sufficiently expressive representer capable of accurately reconstructing full-field outputs from compact latent states.

To tackle these challenges, we propose **StreamPhy**, a deep state space framework that seamlessly integrates a tailored attention-based observation encoder, a HiPPO-based structured SSM, and a functional tensor-based feature-wise linear modulation module. Through this synergistic design, the proposed framework enables accurate and scalable modeling of high-dimensional continuous physical dynamics from sparse and irregular observations.

3 Methodology

3.1 Observation Encoder for Arbitrary Patterns

As in Eq. (4), prior SSMs [18, 20, 19] mainly target multivariate time series with regularly sampled observations, used either directly as s_t or encoded via RNNs or CNNs. In contrast, at each time step t_m, our observation is a set $\mathcal{O}_{t_m}$ with varying cardinality, which cannot be directly handled by existing SSM models. To address this limitation, we aim to develop

a novel observation encoder that maps each observation set into a structured latent representation compatible with the SSM framework. Motivated by the ability of attention to naturally handle variable-length inputs, we design an attention-based observation encoder to effectively aggregate irregular observations sampled from arbitrary patterns, as illustrated in Fig. 2.

Consider a time step t (we omit the subscript m for simplicity), where the observation is given by $\mathcal{O}_t$. We first decompose $\mathcal{O}_t$ into an observation value set $\{y_{t,\mathbf{i}_n}\}_{n=1}^N$ and a continuous index set $\{\mathbf{i}_n\}_{n=1}^N$. Each index $\mathbf{i}_n$ is then mapped to a functional tensor representation $\mathbf{u}_n \in \mathbb{R}^{\sum_{k=1}^K R_k}$ defined as

$$\mathbf{u}_n = \text{Concat}\big[\mathbf{u}^1_{\boldsymbol{\theta}_1}(i_1), \cdots, \mathbf{u}^k_{\boldsymbol{\theta}_k}(i_k), \cdots, \mathbf{u}^K_{\boldsymbol{\theta}_K}(i_K)\big], \quad (6)$$

where $\mathbf{u}^k_{\boldsymbol{\theta}_k}(i_k) : \mathbb{R}_+ \to \mathbb{R}^{R_k}$ is a learnable function parameterized by $\boldsymbol{\theta}_k$ that maps the continuous coordinate i_k to its latent embedding. Specifically, we adopt the idea of INR [17]: $\mathbf{u}^k_{\boldsymbol{\theta}_k}(i_k) = \text{MLP}\big([\cos(2\pi\boldsymbol{\phi}_k i_k), \sin(2\pi\boldsymbol{\phi}_k i_k)]\big)$, where $\boldsymbol{\phi}_k$ is a learnable frequency vector that modulates the input coordinate. Collecting all functional tensor representations and their corresponding observations, we obtain $\mathbf{y}_t \in \mathbb{R}^{N\times 1}$ and $\mathbf{U}_t \in \mathbb{R}^{N\times KR}$. We concatenate them and project the result into a latent representation $\mathbf{D}_t \in \mathbb{R}^{N\times D}$ via a linear transformation.

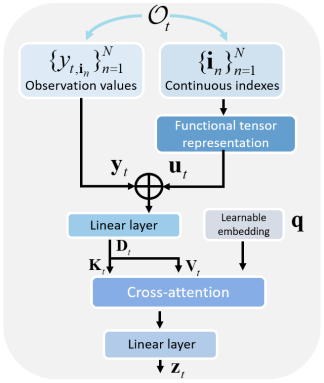

Figure 2: Details of the proposed single-head observation encoder module.

Next, we aim to aggregate the N latent samples into a structured representation that is independent of N. A straightforward approach is to compute a weighted average over $\mathbf{D}_t$ along the sample dimension N. However, such a strategy treats all observation points uniformly and thus fails to capture informative variations across samples. To address this limitation, we introduce a learnable query embedding $\mathbf{q} \in \mathbb{R}^D$ and employ a cross-attention mechanism over $\mathbf{D}_t$ to adaptively select more informative observations [11]:

$$\text{CrossAttention}(\mathbf{q}, \mathbf{K}_t, \mathbf{V}_t) = \text{Softmax}\left(\frac{\mathbf{q}^{\mathrm{T}}\mathbf{K}_t^{\mathrm{T}}}{\sqrt{D}}\right)\mathbf{V}_t \in \mathbb{R}^{1\times D}, \quad (7)$$

where $\mathbf{K}_t = \mathbf{D}_t\mathbf{W}_K \in \mathbb{R}^{N\times D}, \mathbf{V}_t = \mathbf{D}_t\mathbf{W}_V \in \mathbb{R}^{N\times D}$ and $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times D}$ are learnable projection matrices. Finally, the aggregated representation is passed through an MLP to produce the output of the observation encoder:

$$\mathbf{z}_t = \text{MLP}\big(\text{CrossAttention}(\mathbf{q}, \mathbf{K}_t, \mathbf{V}_t)\big) \in \mathbb{R}^P. \quad (8)$$

Note that $\mathbf{q}$ is shared across all time steps, enabling the encoder to produce temporally consistent and structurally stable representations.

In practice, observations are typically sampled in non-uniform patterns, where both the number and spatial distribution of samples vary over time. To mimic this setting, we adopt a stochastic masking strategy that randomly removes a subset of observations during training, thereby encouraging the model to learn robust representations.

We further extend the above formulation to a multi-head version to capture diverse dependencies with H heads. For each head h, we compute $\mathbf{q}^{(h)} = \mathbf{q}[(h-1)d_h : hd_h]$, $\mathbf{K}_t^{(h)} = \mathbf{D}_t\mathbf{W}_K^{(h)}$, $\mathbf{V}_t^{(h)} = \mathbf{D}_t\mathbf{W}_V^{(h)}$ and have $\text{head}_h = \text{Softmax}\left(\frac{(\mathbf{q}^{(h)})^{\mathrm{T}}(\mathbf{K}_t^{(h)})^{\mathrm{T}}}{\sqrt{d_h}}\right)\mathbf{V}_t^{(h)}$, where $d_h = D/H$. The outputs of all heads are concatenated and projected:$\text{MultiHead}(\mathbf{q}, \mathbf{K}_t, \mathbf{V}_t) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}_O$. Finally, we have $\mathbf{z}_t = \text{MLP}\big(\text{MultiHead}(\mathbf{q}, \mathbf{K}_t, \mathbf{V}_t)\big)$.

With the proposed observation encoder, we extend the scalar input s_t in Eq. (5) to a P-dimensional representation $\mathbf{z}_t \in \mathbb{R}^P$, yielding the following new transition function:

$$\mathbf{X}_t = \bar{\mathbf{A}}\mathbf{X}_{t-1} + \bar{\mathbf{b}} \circ \mathbf{z}_t, \quad (9)$$

where $\mathbf{X}_t \in \mathbb{R}^{L\times P}$ denotes the augmented state that tracks the evolution of $\mathbf{z}_t$ and $\circ$ denotes the outer product, producing a $L \times P$ injection term.



(a) Functional Tucker model [14, 16]. (b) Functional tensor FiLM.

Figure 3: Illustrative comparison of FTM and FT-FiLM (K=3).

3.2 Functional Tensor Feature-wise Linear Modulation as an Expressive Representer

As shown in Eq. (5), existing SSMs typically decode signals via a linear combination of latent states and observations. However, such formulations suffer from limited expressive power and are not well-suited for high-dimensional full-field generation.

When performing reconstruction, we aim to enable querying the field value at any coordinate in the continuous domain. Formally, we seek a reconstruction function of the form

$$y_t(\mathbf{i}_n) = f(\mathbf{X}_t, \mathbf{z}_t \mid \mathbf{u}_n), \tag{10}$$

where $\mathbf{u}_n$ denotes the functional representation associated with coordinate $\mathbf{i}_n$, shown in Eq. (6). Such a formulation decouples spatial representation from temporal dynamics, allowing flexible conditioning on both the latent state and spatial features. This is particularly desirable for high-dimensional field generation, where expressive interactions between $\mathbf{X}_t$, $\mathbf{z}_t$, and $\mathbf{u}_n$ are essential.

We propose a Functional Tensor FiLM (FT-FiLM) module. FT-FiLM is an extension of Feature-wise Linear Modulation (FiLM) [28] into the functional tensor domain. FiLM is known for its effectiveness in conditional reasoning and generative modeling. The key idea is to incorporate feature-wise affine modulation into a functional tensor representation, where both the scale and shift parameters are explicitly conditioned on the latent state, enabling flexible interaction between dynamics and spatial representations. Specifically, we first generate modulation parameters conditioned on both latent state $\mathbf{X}_t$ and observation latent $\mathbf{z}_t$:

$$\mathbf{\Gamma}_t = f_{\boldsymbol{\omega}_1}(\mathbf{X}_t, \mathbf{z}_t), \quad \boldsymbol{\beta}_t = f_{\boldsymbol{\omega}_2}(\mathbf{X}_t, \mathbf{z}_t), \tag{11}$$

where $f_{\boldsymbol{\omega}_1}$ and $f_{\boldsymbol{\omega}_2}$ are learnable mappings. The detailed computation of these functions is provided in Appx. B. Here, $\mathbf{\Gamma}_t \in \mathbb{R}^{V \times \sum_{k=1}^{K} R_k}$ and $\boldsymbol{\beta}_t \in \mathbb{R}^V$ denote the feature-wise scaling and shifting parameters, respectively, which are used to modulate the latent representations in a feature-wise manner.

Given $\mathbf{u}_n$ indexed at $\mathbf{i}_n$, the field value is then computed as

$$y_t(\mathbf{i}_n) = f_{\boldsymbol{\varphi}}(\mathbf{\Gamma}_t \mathbf{u}_n + \boldsymbol{\beta}_t)\,, \tag{12}$$

where $f_{\boldsymbol{\varphi}} : \mathbb{R}^V \to \mathbb{R}$ is a learnable readout function.

This formulation enables a more flexible and expressive coupling between latent dynamics and high-dimensional field representations, yielding improved capacity over the functional Tucker model (FTM) [14, 16]. Illustrative comparisons for $K = 3$ are provided in Fig. 3. From a theoretical perspective, Theorem 1 establishes that FT-FiLM is more expressive than FTM in the sense of uniform approximation, with its closure coinciding with the entire space $C(\Omega)$ (see the definition of closure in the theorem below and Appx. A for the proof and further explanations of the topological terminology). This stronger expressivity provides a more flexible basis for modeling complex dynamics.

Theorem 1 (Expressivity of FT-FiLM)**.** *Let $K \geq 2$, and let $\Omega_k \subset \mathbb{R}$ be compact intervals with nonempty interior. Set $\Omega = \Omega_1 \times \cdots \times \Omega_K$. Fix ranks $\mathbf{R} = (R_1, \ldots, R_K)$ with $R_k \geq 1$, and let $S = \sum_{k=1}^{K} R_k$. Assume the FT-FiLM width satisfies $V \geq K$. Define*

$$\mathcal{F}_{\mathrm{FTM}}(\mathbf{R}) = \Big\{ \mathbf{i} \mapsto \mathit{vec}(\boldsymbol{\mathcal{G}})^{\mathrm{T}} \big(\mathbf{u}^1(i_1) \otimes \cdots \otimes \mathbf{u}^K(i_K)\big) \;\Big|\; \boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times \cdots \times R_K},\ \mathbf{u}^k \in C(\Omega_k, \mathbb{R}^{R_k}) \Big\},$$

$$\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V) = \Big\{ \mathbf{i} \mapsto f_{\boldsymbol{\varphi}}\big(\mathbf{W}_g\, \mathbf{u}_{\mathbf{i}} + \mathbf{w}_b\big) \;\Big|\; \mathbf{u}_{\mathbf{i}} = \big[\mathbf{u}^1(i_1); \ldots; \mathbf{u}^K(i_K)\big] \in \mathbb{R}^S,$$
$$\mathbf{W}_g \in \mathbb{R}^{V \times S},\ \mathbf{w}_b \in \mathbb{R}^V,\ \mathbf{u}^k \in C(\Omega_k, \mathbb{R}^{R_k}),\ f_{\boldsymbol{\varphi}} \in \mathcal{N} \Big\},$$

where $\mathcal{N}$ is a class of continuous feedforward neural networks $f_{\boldsymbol{\varphi}} : \mathbb{R}^V \to \mathbb{R}$ satisfying the universal approximation property [29]. Let $\overline{(\cdot)}$ denote closure under the uniform norm on Ω. For a function class $\mathcal{F} \subseteq C(\Omega)$, this closure contains all continuous functions that members of $\mathcal{F}$ can approximate arbitrarily closely everywhere on Ω, i.e.,

$$\overline{\mathcal{F}} = \left\{ h \in C(\Omega) : \begin{array}{c} \textit{for every } \varepsilon > 0, \textit{ there exists } g \in \mathcal{F} \\ \textit{such that } \sup_{\mathbf{i} \in \Omega} |h(\mathbf{i}) - g(\mathbf{i})| < \varepsilon \end{array} \right\}. \tag{13}$$

Then

$$\overline{\mathcal{F}_{\mathrm{FTM}}(\mathbf{R})} \subsetneq \overline{\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)} = C(\Omega). \tag{14}$$

In Theorem 1, we require the readout network class $\mathcal{N}$ to satisfy the universal approximation property [29]. This means that, for every compact rectangle $Q \subset \mathbb{R}^V$, every continuous target $q \in C(Q)$, and every tolerance $\eta > 0$, there exists a readout $f_{\boldsymbol{\varphi}} \in \mathcal{N}$ such that

$$\sup_{\mathbf{z} \in Q} |q(\mathbf{z}) - f_{\boldsymbol{\varphi}}(\mathbf{z})| < \eta.$$

Thus the readout can approximate any continuous target uniformly on Q; its parameters and network size may depend on the target and tolerance.

On the whole, by integrating the proposed observation encoder and FT-FiLM module into the SSM, we obtain our StreamPhy framework, which extends the formulations in Eqs. (4)–(5), given by:

$$\begin{aligned} \mathbf{X}_t &= \bar{\mathbf{A}}\mathbf{X}_{t-1} + \bar{\mathbf{b}} \circ \mathbf{z}_t, \\ y_t(\mathbf{i}_n) &= f_{\boldsymbol{\varphi}}(\boldsymbol{\Gamma}_t \mathbf{u}_n + \boldsymbol{\beta}_t)\,. \end{aligned} \tag{15}$$

We summarize the overall training procedure in Algorithm 1 (see Appx. C).

4 Experiment

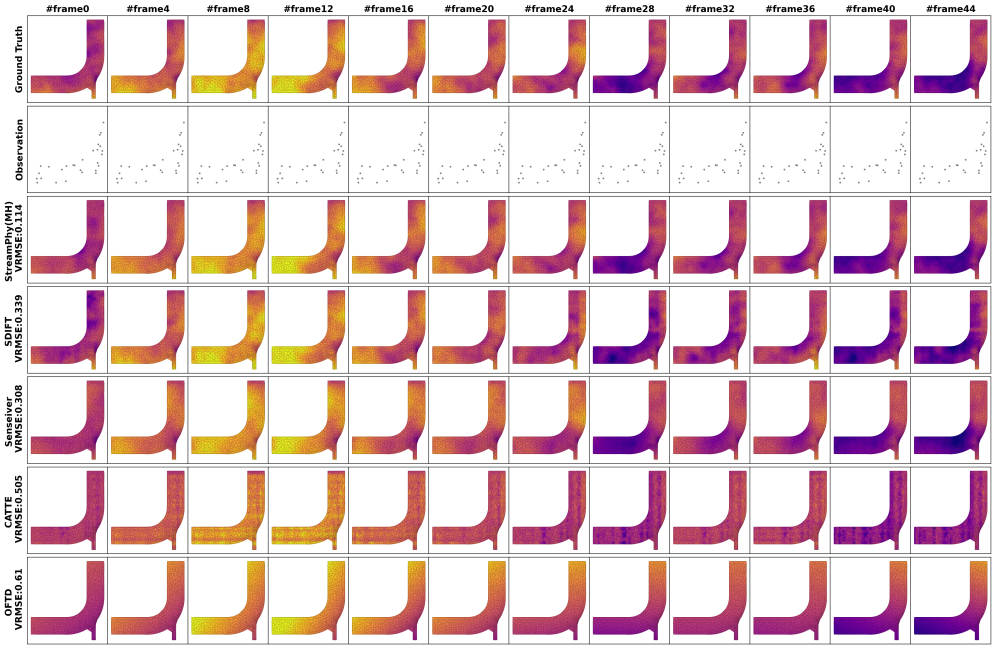

Figure 4: Reconstruction of *Turbulent Flow* dynamics under uniform sampling pattern with $\rho = 3\%$.
Datasets: We evaluate StreamPhy on three physical datasets. (1) *Turbulent Flow*(`https://zenodo.org/records/14037782`) , which captures the spatiotemporal evolution of turbulent fluid velocity fields with complex multiscale vortex dynamics and chaotic structures. We extract 600 records of size $48 \times 918 \times 1$, using 500 for training (each consisting of 48 frames

	Uniform Sampling						Slab Sampling					
	Turbulent Flow		*Ocean Sound Speed*		*Active Matter*		*Turbulent Flow*		*Ocean Sound Speed*		*Active Matter*	
	$\rho = 3\%$	$\rho = 5\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 3\%$	$\rho = 5\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 1\%$	$\rho = 3\%$
Tensor-based												
LRTFR [14]	0.5633	0.3505	0.3453	0.2176	0.3021	0.2582	0.6517	0.4897	0.4075	0.3757	0.9988	0.9537
OFTD [12]	0.5416	0.4810	0.2571	0.1470	0.5542	0.3289	0.5034	0.4901	0.5348	0.4847	0.7387	0.6721
CATTE [13]	0.6192	0.5916	0.0941	0.0850	0.1581	0.1567	0.6563	0.6257	0.1225	0.1187	0.1960	0.1920
Attention-based												
Senseiver [30]	0.2779	0.2361	0.1553	0.1248	0.2555	0.2020	0.3052	0.2578	0.2008	0.1768	0.2762	0.2430
Diffusion-based												
SDIFT [16]	0.3248	0.2821	0.1466	0.1084	0.2158	0.1564	0.4070	0.4029	0.2713	0.1452	0.4231	0.3985
SSM-based												
StreamPhy (SH)	0.1283	0.0954	0.0862	0.0760	0.0895	0.0833	0.2094	0.1979	0.1324	0.1045	0.1452	0.1278
StreamPhy (MH)	**0.0935**	**0.0696**	**0.0628**	**0.0562**	**0.0752**	**0.0732**	**0.1067**	**0.0978**	**0.0692**	**0.0634**	**0.0905**	**0.0866**

Table 1: Average VRMSEs under two sampling patterns across datasets and observation ratios.

with 918 irregular spatial observations) and 100 for testing. (2) *Ocean Sound Speed*(`https://ncss.hycom.org/thredds/ncss/grid/GLBy0.08/expt_93.0/ts3z/dataset.html`), which contains sound speed field measurements in the Pacific Ocean. We extract 1000 records of size $24 \times 5 \times 38 \times 76$ (24 frames), using 950 for training and 50 for testing; during training, only 10% of spatial points are used to simulate irregular observations. (3) *Active Matter*(`https://polymathic-ai.org/the_well/datasets/active_matter/`), which models the dynamics of rod-like active particles in a Stokes fluid via continuum simulations. We extract 928 records of size $24 \times 256 \times 256$ (24 frames), using 900 for training and 28 for testing; similarly, 10% of spatial points are used during training.

Baselines and Settings: We conduct comprehensive evaluations of StreamPhy against state-of-the-art approaches for physical field reconstruction and tensor-based methods. (1) **SDIFT** [16], a diffusion framework in functional Tucker space that reconstructs full-field multidimensional physical dynamics from irregular sparse observations by modeling latent functional representations and their temporal evolution; (2) **Senseiver** [30], an attention-based method that embeds sparse sensor measurements into a unified latent space for efficient multidimensional field reconstruction; (3) **LRTFR** [14], a low-rank functional Tucker model leveraging factorized neural representations for tensor decomposition; (4) **OFTD** [12], an online tensor decomposition approach incorporating implicit neural representations into CP model to continuously model streaming data while updating without forgetting past information; (5) **CATTE** [13], a functional temporal tensor decomposition framework that encodes continuous indices with Fourier features and neural ODEs, while automatically adapting model complexity through sparsity-inducing priors. We evaluate all methods on the task of temporal physical field reconstruction under two sampling patterns, with observation ratios $\rho \in \{1\%, 3\%, 5\%\}$. For the uniform sampling pattern, observations are evenly distributed across the entire spatiotemporal domain. At each time step t_i, a constant sampling ratio ρ is maintained, and observations are acquired sequentially in temporal order. For the slab sampling pattern, we consider a non-uniform observation scheme [1][2], while preserving the same sampling ratio ρ at each time step t_i. Following [16], we evaluate performance using the variance-scaled root mean squared error (VRMSE; definition refers to Appx. E.1), a scale-invariant measure of reconstruction accuracy. Each experiment is repeated 10 times, and we report the mean test error. All the implementation details of StreamPhy and other baselines are provided in Appx. E.

Main Evaluation Results: Note that only OFTD and our StreamPhy support online inference, whereas all other methods require the full temporal sequence before inference. We evaluate two StreamPhy variants: (i) StreamPhy with a single-head observation encoder and the masking strategy in Eq. 8 (StreamPhy(SH)), (ii) StreamPhy with a multi-head observation encoder and the masking strategy (StreamPhy(MH)). As reported in Tab. 1, two variants of StreamPhy outperform all baselines under both sampling patterns. StreamPhy with a multi-head observation encoder outperforms its single-head counterpart by effectively captures diverse dependencies across various subspaces. Qualitative results in Fig. 4 further show that StreamPhy produces more accurate and visually reconstructions of turbulent flow in an online manner. At the same observation rate, slab sampling is more challenging, and all baselines exhibit a noticeable performance drop (especially SDIFT). This is because it yields spatially concentrated and less informative measurements, which leads to poor

[2]Slab sampling refers to observing entire tensor slices at once, as in Unmanned Aerial Vehicle (UAV) sensing, where a drone captures full field slices over a trajectory or time window rather than pointwise measurements.

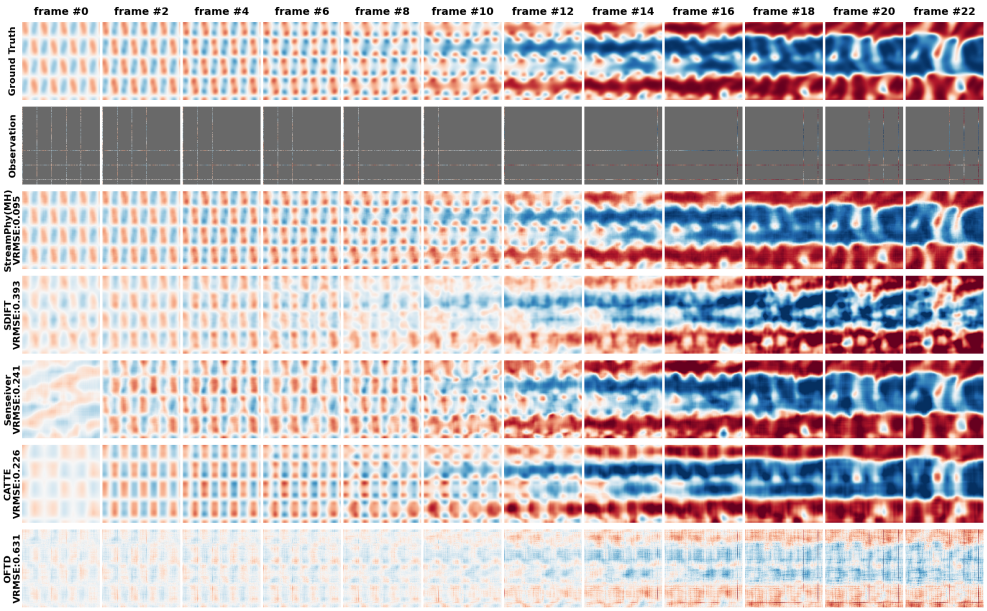

Figure 5: Reconstruction of *Active Matter* dynamics under slab sampling with $\rho = 3\%$.

	Uniform Sampling						Slab Sampling					
	Turbulent Flow		*Ocean Sound Speed*		*Active Matter*		*Turbulent Flow*		*Ocean Sound Speed*		*Active Matter*	
	$\rho = 3\%$	$\rho = 5\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 3\%$	$\rho = 5\%$	$\rho = 1\%$	$\rho = 3\%$	$\rho = 1\%$	$\rho = 3\%$
StreamPhy	**0.0935**	**0.0696**	**0.0628**	**0.0562**	**0.0752**	**0.0732**	**0.1067**	**0.0978**	**0.0692**	**0.0634**	**0.0905**	**0.0866**
StreamPhy w.o. SSM	0.8537	0.8501	0.3935	0.3923	0.9023	0.8687	0.8598	0.8556	0.3966	0.3954	0.9115	0.9107
StreamPhy w.o. mask	0.2420	0.1789	0.0899	0.0786	0.0898	0.0786	0.2841	0.2643	0.2030	0.1768	0.2401	0.1511
StreamPhy with FTM	0.1435	0.1134	0.1054	0.1034	0.1279	0.1229	0.1731	0.1574	0.1202	0.1154	0.1522	0.1454

Table 2: Ablation study on the effectiveness of the introduced SSM framework, the proposed masking strategy and FT-FiLM. All models use the multi-head observation encoder.

coverage, ill-conditioned reconstruction, and stronger dependence on long-range dependencies. As shown in Fig. 5, StreamPhy effectively alleviates these difficulties and maintains high reconstruction accuracy even though such pattern is absent during training, demonstrating strong robustness. More practical sensing conditions like real-world sensor noise, sensor drift and etc. are considered in Appx. F. Additional reconstruction visualizations are provided in Appx. G. Overall, these results validate the effectiveness of StreamPhy.

Ablation study: We conduct an ablation study to evaluate the contributions of the proposed SSM framework, the masking strategy under varying sampling patterns and sparsity levels, and the expressivity gain of FT-FiLM over FTM. We compare the full StreamPhy with three variants: (i) *w.o. SSM*, which removes the state space model (implemented by setting all $\mathbf{X}_t$ in Eq. 11 to zero), (ii) *w.o. mask*, which disables the masking mechanism in the observation encoder, and (iii) *with FTM*, which replaces FT-FiLM with FTM. All variants share the same multi-head encoder. As shown in Table 2, removing the SSM leads to a drastic performance degradation across all datasets and settings, underscoring its critical role in capturing long-range temporal dependencies from streaming observations. Disabling the masking strategy also results in consistent performance drops, particularly under slab sampling, highlighting its importance in handling irregular and partially observed inputs and in improving both accuracy and robustness. Replacing FT-FiLM with FTM further degrades performance, demonstrating the superior expressivity of FT-FiLM for high-fidelity reconstruction. Overall, StreamPhy achieves the best performance across all settings, indicating that all components are essential.

Inference Speed: We compare the inference speed of our method with SDIFT [16] on an NVIDIA RTX 4090 GPU (24 GB memory) on a single record. The results are reported in Tab. 3. It is evident that our method outperforms SDIFT across all settings, achieving a speedup of approximately $20\times$–$100\times$. This substantial acceleration arises from two main aspects. 1) StreamPhy adopts an end-to-end

Methods	*Turbulent Flow*			*Ocean Sound Speed*			*Active Matter*		
	$\rho = 3\%$	$\rho = 5\%$	#Para.	$\rho = 1\%$	$\rho = 3\%$	#Para.	$\rho = 1\%$	$\rho = 3\%$	#Para.
StreamPhy (MH)	**0.0419s**	**0.0448s**	10.3M	**0.0325s**	**0.0434s**	9.94M	**0.0479s**	**0.0498s**	11M
SDIFT [16]	5.14s	5.21s	15M	0.84s	0.89s	15M	1.31s	1.42s	12M

Table 3: Average inference speed (in seconds) for reconstruction with different observation ratios on uniform sampling patterns on a single record.

architecture that directly maps observations to sequential reconstruction of the full field, whereas SDIFT [16] relies on fully observed temporal data and performs multi-step posterior sampling. In particular, its MPDPS mechanism introduces a Gaussian process with per-frame complexity $O(T^3)$, where T denotes the number of frames, resulting in an overall complexity of $O(T^4)$. In contrast, our method scales linearly with T, i.e., $O(T)$. 2) Benefiting from the strong expressive power of the proposed FT-FiLM module, StreamPhy employs significantly more compact latent representations than SDIFT (e.g., 64 vs. 2304 (48×48) for the active matter dataset), which further contributes to its superior inference efficiency.

5 Conclusion and Limitations

In this work, we presented **StreamPhy**, an end-to-end framework for streaming inference of continuous spatiotemporal dynamics from sparse observations. By proposing a data-adaptive observation encoder that is robust to missing patterns, together with the expressive FT-FiLM module, and integrating both into a HiPPO-based state-space model, StreamPhy enables efficient and accurate streaming inference, offering a scalable solution for engineering applications. Experiments across multiple physical systems demonstrate consistent gains in both accuracy and efficiency under diverse challenging sampling patterns. Limitations of this work are provided in Appx. H.

6 Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62622127 and 62371418, and supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LR26F010004.

References

[1] Guoyong Zhang, Xiao Fu, Jun Wang, Xi-Le Zhao, and Mingyi Hong. Spectrum cartography via coupled block-term tensor decomposition. *IEEE Transactions on Signal Processing*, 68:3660–3675, 2020.

[2] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[3] Charles R Farrar and Keith Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[4] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[5] Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15(1):10416, 2024.

[6] Zeyu Li, Wang Han, Yue Zhang, Qingfei Fu, Jingxuan Li, Lizi Qin, Ruoyu Dong, Hao Sun, Yue Deng, and Lijun Yang. Learning spatiotemporal dynamics with a pretrained generative model. *Nature Machine Intelligence*, 6(12):1566–1579, 2024.

[7] Sifan Wang, Zehao Dou, Siming Shan, Tong-Rui Liu, and Lu Lu. Fundiff: Diffusion models over function spaces for physics-informed generative modeling. *arXiv preprint arXiv:2506.07902*, 2025.

[8] Helen Qu, Rudy Morel, Michael McCabe, Francois Lanusse, Alberto Bietti, Shirley Ho, and Yann LeCun. Representation learning for spatiotemporal physical systems. In *AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations*, 2026.

[9] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

[10] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[12] Xi Zhang, Yanyi Li, Yisi Luo, Qi Xie, and Deyu Meng. Online functional tensor decomposition via continual learning for streaming data completion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] Panqi Chen, Lei Cheng, Jianlong Li, Weichang Li, Weiqing Liu, Jiang Bian, and Shikai Fang. Functional complexity-adaptive temporal tensor decomposition. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] Yisi Luo, Xile Zhao, Zhemin Li, Michael K Ng, and Deyu Meng. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] Shikai Fang, Xin Yu, Zheng Wang, Shibo Li, Mike Kirby, and Shandian Zhe. Functional bayesian tucker decomposition for continuous-indexed tensor data. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Panqi Chen, Yifan Sun, Lei Cheng, Yang Yang, Weichang Li, Yang Liu, Weiqing Liu, Jiang Bian, and Shikai Fang. Generating full-field evolution of physical dynamics from irregular sparse observations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[17] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[18] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.

[19] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[20] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

[21] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[22] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on signal processing*, 65(13):3551–3582, 2017.

[23] Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA working papers in phonetics*, 16(1):84, 1970.

[24] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[25] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[26] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[27] Arnold Tustin. A method of analysing the behaviour of linear systems in terms of time series. *Journal of the Institution of Electrical Engineers-Part IIA: Automatic Regulators and Servo Mechanisms*, 94(1):130–142, 1947.

[28] 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.

[29] Namig J Guliyev and Vugar E Ismailov. Approximation capability of two hidden layer feedforward neural networks with fixed weights. *Neurocomputing*, 316:262–269, 2018.

[30] Javier E Santos, Zachary R Fox, Arvind Mohan, Daniel O'Malley, Hari Viswanathan, and Nicholas Lubbers. Development of the senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence*, 5(11):1317–1325, 2023.

[31] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

[33] Zheng Wang and Shandian Zhe. Nonparametric factor trajectory learning for dynamic tensor decomposition. In *International Conference on Machine Learning*, pages 23459–23469. PMLR, 2022.

[34] Zheng Wang, Shikai Fang, Shibo Li, and Shandian Zhe. Dynamic tensor decomposition via neural diffusion-reaction processes. *Advances in Neural Information Processing Systems*, 36, 2024.

[35] Shikai Fang, Akil Narayan, Robert Kirby, and Shandian Zhe. Bayesian continuous-time tucker decomposition. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6235–6245. PMLR, 17–23 Jul 2022.

[36] Shikai Fang, Xin Yu, Shibo Li, Zheng Wang, Robert Kirby, and Shandian Zhe. Streaming factor trajectory learning for temporal tensor decomposition. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[37] Andrei Chertkov, Gleb Ryzhakov, Georgii Novikov, and Ivan Oseledets. Optimization of functions given in the tensor train format. *arXiv preprint arXiv:2209.14808*, 2022.

[38] Panqi Chen, Lei Cheng, Jianlong Li, Weichang Li, Weiqing Liu, Jiang Bian, and Shikai Fang. Generalized temporal tensor decomposition with rank-revealing latent-ode. *arXiv preprint arXiv:2502.06164*, 2025.

[39] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

[40] Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard Turner, and Emile Mathieu. On conditional diffusion models for pde simulations. *Advances in Neural Information Processing Systems*, 37:23246–23300, 2024.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

A Proof of Theorem 1

Proof. Both model classes consist of continuous functions on Ω: functional Tucker models are finite sums of products of continuous functions, whereas FT-FiLM models are compositions of continuous factor functions, an affine map, and a continuous network. Since continuity is preserved under uniform convergence, the uniform closures of both classes are contained in $C(\Omega)$. For background on basic topological terminology, including closed sets and closure, see [31, Appendix A]. Here, closure is taken with respect to the uniform norm on $C(\Omega)$.

We first show that $\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)$ is dense in $C(\Omega)$, i.e., for every continuous target $h \in C(\Omega)$ and every tolerance $\varepsilon > 0$, there exists $g \in \mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)$ such that $\sup_{\mathbf{i}\in\Omega} |h(\mathbf{i}) - g(\mathbf{i})| < \varepsilon$. Thus every continuous function can be approximated arbitrarily accurately everywhere on Ω, without requiring exact representation. Fix $h \in C(\Omega)$ and $\varepsilon > 0$. Since $R_k \geq 1$ for every k, choose

$$\mathbf{u}^k(i_k) = (i_k, 0, \ldots, 0)^\top \in \mathbb{R}^{R_k}, \qquad k = 1, \ldots, K.$$

The condition $V \geq K$ allows us to choose $\mathbf{W}_g \in \mathbb{R}^{V\times S}$ and $\mathbf{w}_b = \mathbf{0}$ such that

$$\mathbf{W}_g \mathbf{u}_\mathbf{i} + \mathbf{w}_b = (i_1, \ldots, i_K, 0, \ldots, 0)^\top =: \iota(\mathbf{i}).$$

Let

$$D = \Omega \times [-1, 1]^{V-K}, \qquad \widetilde{h}(\mathbf{z}) = h(z_1, \ldots, z_K), \quad \mathbf{z} \in D,$$

with $D = \Omega$ when $V = K$. Then D is compact, $\widetilde{h} \in C(D)$, and $\widetilde{h}(\iota(\mathbf{i})) = h(\mathbf{i})$. Applying the universal approximation property explained immediately after Theorem 1 (see [29]) with $Q = D$, $q = \widetilde{h}$, and $\eta = \varepsilon$, we obtain $f_{\boldsymbol{\varphi}} \in \mathcal{N}$ such that

$$\sup_{\mathbf{z}\in D} \big|\widetilde{h}(\mathbf{z}) - f_{\boldsymbol{\varphi}}(\mathbf{z})\big| < \varepsilon.$$

Since $\iota(\Omega) \subseteq D$, it follows that

$$\sup_{\mathbf{i}\in\Omega} \big|h(\mathbf{i}) - f_{\boldsymbol{\varphi}}\big(\mathbf{W}_g \mathbf{u}_\mathbf{i} + \mathbf{w}_b\big)\big| < \varepsilon.$$

As h and ε were arbitrary, we conclude that

$$\overline{\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)} = C(\Omega).$$

Consequently,

$$\overline{\mathcal{F}_{\text{FTM}}(\mathbf{R})} \subseteq C(\Omega) = \overline{\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)}.$$

To establish strictness, we need to find at least one continuous function that belongs to the FT-FiLM closure $\overline{\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)}$ but not to the fixed-rank FTM closure $\overline{\mathcal{F}_{\text{FTM}}(\mathbf{R})}$, i.e., one that fixed-rank FTM cannot approximate arbitrarily closely under the uniform norm. Consider the continuous function

$$h_\star(\mathbf{i}) = \exp(i_1 i_2).$$

When $K > 2$, fix the remaining coordinates and suppress these fixed arguments in the notation below. Set

$$r = \min\{R_1, R_2\}, \qquad M = r + 1.$$

Because Ω_1 and Ω_2 have nonempty interior, we may choose equally spaced points

$$\alpha_j = a + (j-1)\delta \in \Omega_1, \qquad j = 1, \ldots, M, \qquad \delta > 0,$$

and distinct points $\beta_1 < \cdots < \beta_M$ in Ω_2.

For any $h \in \mathcal{F}_{\text{FTM}}(\mathbf{R})$, its restriction to this slice can be written as

$$h(i_1, i_2) = \mathbf{u}^1(i_1)^\top \mathbf{B} \mathbf{u}^2(i_2), \qquad \mathbf{B} \in \mathbb{R}^{R_1\times R_2},$$

where $\mathbf{B}$ is the Tucker core when $K = 2$, or its contraction with the factor vectors at the fixed coordinates when $K > 2$. The sampled matrix therefore admits the factorization

$$\mathbf{H} := \big[h(\alpha_j, \beta_\ell)\big]_{j,\ell=1}^{M} = \mathbf{A}\mathbf{B}\mathbf{C}^\top,$$

where

$$\mathbf{A}_{jp} = u_p^1(\alpha_j), \qquad \mathbf{C}_{\ell q} = u_q^2(\beta_\ell).$$

Hence

$$\operatorname{rank}(\mathbf{H}) \le \min\{R_1, R_2\} = r < M,$$

so $\det(\mathbf{H}) = 0$.

On the other hand, recall the target function $h_\star(\mathbf{i}) = \exp(i_1 i_2)$ introduced above to establish strictness. Its sampled matrix $\mathbf{M}$ satisfies

$$\mathbf{M}_{j\ell} = e^{\alpha_j \beta_\ell} = e^{a\beta_\ell} \left(e^{\delta\beta_\ell}\right)^{j-1}.$$

Thus $\mathbf{M}$ is a Vandermonde matrix with nonzero column scaling factors, and

$$\det(\mathbf{M}) = \left(\prod_{\ell=1}^{M} e^{a\beta_\ell}\right) \prod_{1\le p<q\le M} \left(e^{\delta\beta_q} - e^{\delta\beta_p}\right) \neq 0.$$

Since the determinant is continuous and $\det(\mathbf{M}) \neq 0$, there exists $\rho > 0$ such that every matrix $\mathbf{A} \in \mathbb{R}^{M\times M}$ satisfying

$$\max_{1\le j,\ell\le M} |A_{j\ell} - M_{j\ell}| < \rho$$

has nonzero determinant. Thus every singular matrix differs from $\mathbf{M}$ by at least ρ in some entry. For any $h \in \mathcal{F}_{\mathrm{FTM}}(\mathbf{R})$, its sampled matrix $\mathbf{H} = [h(\alpha_j, \beta_\ell)]_{j,\ell=1}^{M}$ is singular, as shown above. Consequently,

$$\sup_{\mathbf{i}\in\Omega} |h(\mathbf{i}) - h_\star(\mathbf{i})| \ge \max_{1\le j,\ell\le M} |h(\alpha_j, \beta_\ell) - h_\star(\alpha_j, \beta_\ell)| \ge \rho.$$

Here the sampled values are evaluated on the fixed slice when $K > 2$. This positive lower bound holds for every fixed-rank FTM function. By the definition of closure in Eq. (13), it follows that

$$h_\star \notin \overline{\mathcal{F}_{\mathrm{FTM}}(\mathbf{R})}.$$

Together with $h_\star \in C(\Omega)$ and the density of FT-FiLM, this proves

$$\overline{\mathcal{F}_{\mathrm{FTM}}(\mathbf{R})} \subsetneq \overline{\mathcal{F}_{\text{FT-FiLM}}(\mathbf{R}, V)} = C(\Omega).$$

□

Remark 2 (Scope of Theorem 1)**.** *Theorem 1 establishes a* qualitative *expressivity gap: fixed-rank functional Tucker models cannot uniformly approximate every continuous function on Ω, whereas FT-FiLM can. The density of FT-FiLM is driven by the readout network $f_{\boldsymbol{\varphi}}$; thus the theorem shows that decoupling the rank of the latent factors from the capacity of the readout, which is a key feature of FT-FiLM, removes the rank-induced approximation barrier of FTM.*

B Details of Modulation Computation

Note that,

$$\boldsymbol{\Gamma}_t = f_{\boldsymbol{\omega}_1}(\mathbf{X}_t, \mathbf{z}_t), \quad \boldsymbol{\beta}_t = f_{\boldsymbol{\omega}_2}(\mathbf{X}_t, \mathbf{z}_t), \tag{16}$$

Here, we provide detailed computations of modulation parameters $\boldsymbol{\Gamma}_t$ and $\boldsymbol{\beta}_t$.

$$f_{\boldsymbol{\omega}_1}(\mathbf{X}_t, \mathbf{z}_t) = \text{Reshape}(\text{MLP}(\text{Concat}([\text{vec}(\mathbf{X}_t), \mathbf{z_t}]))). \tag{17}$$

As shown in Eq. (17), $f_{\boldsymbol{\omega}_1}(\cdot,\cdot) : \mathbb{R}^{L\times P} \times \mathbb{R}^P \to \mathbb{R}^{V\times\sum_{k=1}^{K} R_k}$ maps a matrix together with a vector to a structured modulation matrix. Specifically, the input observation $\mathbf{X}_t \in \mathbb{R}^{L\times P}$ is first vectorized as $\text{vec}(\mathbf{X}_t) \in \mathbb{R}^{LP}$ and concatenated with the conditioning variable $\mathbf{z}_t \in \mathbb{R}^P$ to form a unified representation. This concatenated feature is then processed by a multilayer perceptron to capture nonlinear interactions between spatial observations and conditioning signals. The resulting output is subsequently reshaped via $\text{Reshape}(\cdot)$ into a tensor of dimension $V \times \sum_{k=1}^{K} R_k$, where each row corresponds to a feature-wise modulation coefficient over the V latent channels.

In this way, $f_{\boldsymbol{\omega}_1}$ generates a global modulation dictionary that parameterizes feature-wise scaling factors conditioned jointly on $\mathbf{X}_t$ and $\mathbf{z}_t$, enabling expressive cross-feature and cross-condition interactions within the FT-FiLM framework.

Also, we have

$$f_{\boldsymbol{\omega}_2}(\mathbf{X}_t, \mathbf{z}_t) = \text{MLP}(\text{Concat}([\text{vec}(\mathbf{X}_t), \mathbf{z_t}])) \tag{18}$$

As shown in Eq. (18), $f_{\boldsymbol{\omega}_2}(\cdot,\cdot) : \mathbb{R}^{L\times P} \times \mathbb{R}^P \to \mathbb{R}^V$ maps the joint representation of the observed field $\mathbf{X}_t$ and the conditioning variable $\mathbf{z}_t$ to a feature-wise bias vector. In particular, $\mathbf{X}_t$ is first vectorized as $\text{vec}(\mathbf{X}_t) \in \mathbb{R}^{LP}$ and concatenated with $\mathbf{z}_t \in \mathbb{R}^P$, forming a unified input embedding. This embedding is then passed through a multilayer perceptron to capture nonlinear dependencies between spatial observations and the conditioning signal. The output is a V-dimensional vector, where each entry corresponds to the additive shift applied to the corresponding latent feature channel.

Together with the scaling parameters produced by $f_{\boldsymbol{\omega}_1}$, this bias term enables a complete feature-wise affine modulation in the FT-FiLM layer, thereby enhancing the expressiveness of conditional field representations.

C Algorithm

In our setting, the training data is irregular and sparse observations sampled from B batches of homogeneous physical dynamics at arbitrary timesteps.

We summarize the training process of the StreamPhy in Algorithm1.

Algorithm 1 Training process

Require: Batches of training data $\{\{\mathcal{O}^b_{t_m}\}_{m=1}^M\}_{b=1}^B$, time steps $\mathcal{T} = \{t_1, \ldots, t_M\}$. Initialize the latent state $\mathbf{X}_0 = \mathbf{0}$.

while not convergence **do**
  for $t = t_1, \ldots, t_M$ **do**
    Sample a rate from $U[0.1, 1]$ and generate mask to randomly mask-out observation points according to the sampled rate.
    Obtain observation latent $\mathbf{z}_t$ from observation encoder with the mask through Eq. (8).
    Compute discretized matrices $\bar{\mathbf{A}}, \bar{\mathbf{b}}$ using Eq. (3).
    Update the current latent state using Eq. (9).
    Reconstruct full field using Eq. 12.
    Take gradients on the RMSE of reconstruction errors using Adam [32] optimizer.
  end for
end while
return Well-trained model parameters.

D Related Work

Tensor-based methods leverage low-rank structures to recover high-dimensional data from sparse observations. Recent progress in temporal tensor modeling [33, 34, 35, 36] extends classical discrete formulations to continuous-time regimes, while functional tensor approaches [15, 14, 37, 38] represent tensor modes as continuous functions with structured latent parameterizations. More recent studies [13, 12] further incorporate Neural ODEs or streaming mechanisms to model temporal dynamics. Despite these advances, such methods typically operate on individual instances, limiting their ability to exploit shared structure and generalize across heterogeneous data.

To address this limitation, generative modeling has recently emerged as an alternative paradigm. A growing body of work [9, 39, 40, 4, 6] leverages pre-trained diffusion models and demonstrates strong performance across diverse physical modeling tasks. However, these approaches generally rely on regularly sampled, well-structured data, restricting their applicability in realistic settings with sparse and irregular observations. To mitigate this issue, [5, 16] project irregular observations into a regularized latent space and model the temporal evolution of latent distributions. At inference time, full-field reconstruction is achieved by incorporating likelihood-based gradients into the score function as guidance. Nevertheless, this two-stage pipeline is prone to accumulated approximation errors and reconstruction artifacts, while incurring significant computational overhead due to repeated forward passes and gradient evaluations.

Structured state space models (SSMs) have recently emerged as an efficient and principled framework for long-range sequence modeling. Early work such as HiPPO [18] introduces a continuous-time memory mechanism based on optimal polynomial projections, providing a theoretically grounded way to compress streaming signals into finite-dimensional states. Building on this foundation, S4 [20] develops a scalable parameterization of linear state space systems, enabling efficient modeling of long sequences through structured transition matrices and fast convolutional implementations. Subsequent work further unifies recurrent, convolutional, and continuous-time formulations via linear state space layers [19], offering a flexible framework that bridges different sequence modeling paradigms. More recently, selective state space models such as Mamba [21] introduce input-dependent state transitions, significantly enhancing modeling capacity while retaining linear-time complexity. Despite their strong performance, these approaches typically assume regularly sampled inputs and are not explicitly designed for sparse and irregular observations in spatiotemporal reconstruction tasks.

E Implementation details

All the methods are implemented with PyTorch [41] and trained using Adam [32] optimizer with the learning rate tuned from $\{5e^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}\}$.

For StreamPhy, we set $L = 32, h = 4, D = 512, P = 64$ for all datasets. While $\mathbf{R} = (64, 64)$ for Turbulent Flow, $\mathbf{R} = (3, 15, 30)$ for Ocean Sound Speed and $\mathbf{R} = (128, 128)$ for Active Matter.

For OFTD, we adopt a two-stage online CP decomposition scheme with an initial window of $t_{\text{initial}} = 5$ frames and an online update step of $\Delta t = 1$. We parameterize the temporal factor using a two-layer SIREN followed by a linear output layer, with $\omega_0 = 0.3$. We train the initial fitting stage for 4000 iterations and each subsequent online update for 100 iterations, while holding out 10% of the observed entries for validation and using early stopping with patience 10. Across datasets, we set the CP rank and hidden width to $(256, 256)$ for *active matter* and $(64, 192)$ for both *SSF* and *turbulent flow*.

For LRTFR, we parameterize each mode-wise latent factor using a two-layer SIREN with a linear output layer, and reconstruct the full field through a low-rank core tensor. We train the model using Adam with learning rate 10^{-4} and weight decay 0.1 for 3001 iterations, again reserving 10% of the observed entries for validation. The hidden width is set to 192 for *active matter* and 128 for both *SSF* and *turbulent flow*. The mode-wise downsampling factors are set to $\{8, 8, 4\}$, $\{2, 8, 8, 8\}$, and $\{8, 4\}$ for the three datasets, respectively, and the rank of each mode is determined adaptively as the corresponding mode size divided by its downsampling factor.

For Senseiver, we use 32 spatial Fourier bands for positional encoding in all cases. For *active matter*, we additionally enable time as an extra input dimension, and set the encoder preprocessing width to 96, the number of latent tokens to 12, the encoder and decoder latent widths to 48, the number of encoder layers to 4, the numbers of cross-attention and self-attention heads to 4, and the number of self-attention layers per block to 3. For *SSF* and *turbulent flow*, we use the default configuration with encoder preprocessing width 64, 4 latent tokens, encoder and decoder latent widths of 16, 3 encoder layers, 2 encoder cross-attention heads, 2 encoder self-attention heads, and 1 decoder cross-attention head.

For CATTE, we set the ODE state dimension $J = 15$ and the initial number of components of the factor trajectories $R = 15$. We used two hidden layers for velocity network, with the layer width chosen from $\{128, 256, 512\}$. For LRTFR, we used two hidden layers with layer width chosen from $\{128, 256, 512\}$ to parameterize the latent function of each mode. We varied R from $\{16, 32\}$ for all baselines. For Senseiver, we used 128 channels in both the encoder and decoder, a sequence size of 256 for the Q_{in} array, and set the size of the linear layers in the encoder and decoder to 128.

For SDIFT, we first apply a functional Tucker model to decompose the tensor into factor functions and a core sequence. Each factor function is parameterized by a three-layer MLP, where each layer contains 1024 neurons and uses the sine activation function. The core sizes are set to 48×48, $3 \times 12 \times 12$, and 48×48 for the *Turbulent Flow*, *Ocean Sound Speed*, and *Active Matter* datasets, respectively. These hyperparameters are carefully selected to achieve optimal performance.

Table 4: Reconstruction performance under different types and levels of sensor noise on the Active Matter Dataset with uniform sampling ratio $\rho = 1\%$. Results are reported as mean $\pm$ standard deviation.

Noise Configuration	StreamPhy	SDIFT	CATTE	OFTD	LRTFR	Senseiver
Gaussian noise ($\sigma = 0.1$)	$\mathbf{0.077 \pm 0.007}$	0.221 ± 0.035	0.160 ± 0.033	0.583 ± 0.199	0.305 ± 0.068	0.261 ± 0.050
Gaussian noise ($\sigma = 0.3$)	$\mathbf{0.108 \pm 0.009}$	0.230 ± 0.046	0.167 ± 0.038	0.615 ± 0.180	0.306 ± 0.067	0.298 ± 0.054
Laplacian noise ($\sigma = 0.1$)	$\mathbf{0.083 \pm 0.009}$	0.235 ± 0.044	0.179 ± 0.055	0.583 ± 0.203	0.302 ± 0.069	0.269 ± 0.051
Laplacian noise ($\sigma = 0.3$)	$\mathbf{0.139 \pm 0.012}$	0.263 ± 0.042	0.196 ± 0.045	0.616 ± 0.187	0.304 ± 0.069	0.326 ± 0.065
Poisson noise ($\sigma = 0.1$)	$\mathbf{0.075 \pm 0.005}$	0.228 ± 0.042	0.163 ± 0.062	0.598 ± 0.213	0.307 ± 0.068	0.333 ± 0.085
Poisson noise ($\sigma = 0.3$)	$\mathbf{0.084 \pm 0.009}$	0.241 ± 0.049	0.181 ± 0.044	0.624 ± 0.212	0.308 ± 0.067	0.421 ± 0.111

E.1 Definition of Variance-scaled Root Mean Squared Error

Let $\{\hat{y}_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ denote the predicted and ground-truth entry, respectively. Assume that there are N points in total. The Variance-scaled Root Mean Squared Error (VRMSE) is defined as

$$\mathrm{VRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{L}(\hat{y}_i - y_i)^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{L}(y_i - \bar{y})^2}}, \tag{19}$$

where $\bar{y}$ is the mean of all points.

F Additional Experiment Results

Here, we report additional experimental results to further evaluate the robustness and computational efficiency of StreamPhy under practical sensing conditions. Specifically, we consider real-world sensor noise, sensor drift and distribution shifts, missing frames, irregular temporal arrivals, and delayed sensor measurements. We also compare the model size, training cost, and inference efficiency of StreamPhy with all baseline methods across different datasets and observation densities. Unless otherwise specified, the robustness experiments are conducted on the Active Matter Dataset under uniform spatial sampling.

Robustness to real-world sensor noise. We first evaluate the robustness of StreamPhy under different types and levels of measurement noise. Gaussian noise is used to model additive random measurement errors commonly introduced by sensors and environmental disturbances. Laplacian noise is used to simulate impulsive noise, such as occasional sensor spikes, while Poisson noise is used to simulate signal-dependent noise commonly observed in photon-counting and event-based sensing systems.

Table 4 summarizes the reconstruction performance under different noise configurations. StreamPhy consistently maintains low reconstruction errors across different noise types and noise levels, demonstrating its robustness to corrupted sensor observations.

Robustness to sensor drift and distribution shifts. Real-world sensors may exhibit temporally varying measurement biases due to device aging, calibration errors, or environmental changes. To evaluate the robustness of StreamPhy against such distribution shifts, we consider two representative types of sensor drift: gradual drift and recurring drift.

For gradual drift, we introduce an exponential temporal drift into the measurements during inference:

$$d_{\mathrm{grad}}(t) = \alpha \exp\left(\frac{t - T}{T}\right), \tag{20}$$

where t denotes the current timestamp, T denotes the total temporal horizon, and α controls the drift magnitude.

The results are reported in Table 5. StreamPhy maintains stable reconstruction performance under both mild and stronger gradual distribution shifts.

Table 5: Reconstruction performance under gradual sensor drift on the Active Matter Dataset with uniform sampling ratio $\rho = 1\%$.

Drift Magnitude	StreamPhy	SDIFT	CATTE	OFTD	LRTFR	Senseiver
$\alpha = 0.1$	$\mathbf{0.082 \pm 0.007}$	0.262 ± 0.032	0.201 ± 0.025	0.557 ± 0.231	0.317 ± 0.062	0.276 ± 0.047
$\alpha = 0.3$	$\mathbf{0.132 \pm 0.010}$	0.316 ± 0.033	0.303 ± 0.034	0.591 ± 0.216	0.386 ± 0.045	0.383 ± 0.071

Table 6: Reconstruction performance under recurring sensor drift on the Active Matter Dataset with uniform sampling ratio $\rho = 1\%$.

Drift Magnitude	StreamPhy	SDIFT	CATTE	OFTD	LRTFR	Senseiver
$\alpha = 0.1$	$\mathbf{0.081 \pm 0.011}$	0.273 ± 0.075	0.208 ± 0.053	0.566 ± 0.231	0.317 ± 0.063	0.280 ± 0.045
$\alpha = 0.3$	$\mathbf{0.147 \pm 0.027}$	0.363 ± 0.073	0.346 ± 0.089	0.606 ± 0.191	0.401 ± 0.043	0.397 ± 0.056

For recurring drift, we introduce a periodic temporal perturbation following [**?**]:

$$d_{\mathrm{rec}}(t) = \alpha \sin\left(4\pi \frac{t}{T}\right), \tag{21}$$

where α controls the magnitude of the periodic drift.

As shown in Table 6, StreamPhy remains robust under recurring temporal distribution shifts across different drift magnitudes.

Robustness to missing frames. Real-world sensing systems may suffer from incomplete temporal observations due to packet loss, communication failures, or temporary sensor outages. Following Observation Setting 2 in SDIFT [16], we consider a setting in which observations are available only for a subset of frames.

Table 7 reports the reconstruction results under uniform sampling ratios of $\rho = 1\%$ and $\rho = 3\%$. StreamPhy maintains low reconstruction errors under both sampling ratios, demonstrating its robustness to incomplete temporal observation histories.

Robustness to irregular temporal arrivals. We further evaluate the robustness of StreamPhy when observations arrive at variable temporal intervals. Specifically, we simulate irregular temporal arrivals by selecting frames with different intervals, e.g., Δt, $2\Delta t$, and $3\Delta t$. This setting represents practical scenarios involving asynchronous sensing or missing timestamps, where the temporal gaps between consecutive observations vary dynamically.

As shown in Table 8, StreamPhy maintains stable reconstruction performance despite non-uniform temporal intervals.

Robustness to delayed sensor measurements. We also consider temporal misalignment caused by delayed sensor measurements. Specifically, we fix the sensor locations across all time steps and replace either 10% or 30% of the observations at time t_i with measurements collected at the previous timestamp t_{i-1}. This setting simulates delayed communication or synchronization errors that may arise in distributed sensing systems.

The results are summarized in Table 9. StreamPhy maintains stable reconstruction performance even when a substantial fraction of the current measurements is replaced by temporally delayed observations.

Computational efficiency and model complexity. In addition to robustness under practical sensing imperfections, we compare the computational efficiency and model complexity of StreamPhy with all baseline methods. Table 10 reports the number of model parameters, training time, and inference time on the Turbulent Flow, Ocean Sound Speed, and Active Matter datasets. For inference, we additionally report the runtime under different spatial sampling ratios to evaluate the effect of observation density on computational cost.

The parameter sizes vary across methods due to their different model architectures, and each baseline is individually configured for its corresponding experimental setting. CATTE, OFTD, and LRTFR directly fit observations without a separate offline training stage; therefore, their training times are

Table 7: Reconstruction performance under missing frames on the Active Matter Dataset. Observations are available only for a subset of temporal frames. Results are reported as mean $\pm$ standard deviation.

Method	**Uniform,** $\rho = 1\%$	**Uniform,** $\rho = 3\%$
LRTFR	0.620 ± 0.484	0.598 ± 0.527
OFTD	0.623 ± 0.192	0.366 ± 0.111
CATTE	0.319 ± 0.091	0.271 ± 0.046
Senseiver	0.272 ± 0.048	0.207 ± 0.058
SDIFT	0.296 ± 0.096	0.256 ± 0.087
StreamPhy	$\mathbf{0.089 \pm 0.017}$	$\mathbf{0.081 \pm 0.018}$

Table 8: Reconstruction performance under irregular temporal arrivals on the Active Matter Dataset. Observation intervals vary among Δt, $2\Delta t$, and $3\Delta t$. Results are reported as mean $\pm$ standard deviation.

Method	**Uniform,** $\rho = 1\%$	**Uniform,** $\rho = 3\%$
LRTFR	0.934 ± 0.262	0.925 ± 0.306
OFTD	0.689 ± 0.244	0.672 ± 0.187
CATTE	0.411 ± 0.103	0.470 ± 0.188
Senseiver	0.378 ± 0.050	0.340 ± 0.093
SDIFT	0.347 ± 0.099	0.328 ± 0.089
StreamPhy	$\mathbf{0.125 \pm 0.042}$	$\mathbf{0.106 \pm 0.025}$

reported as N/A. In contrast, StreamPhy performs offline training and subsequently reconstructs physical fields through efficient forward inference.

As shown in Table 10, StreamPhy consistently achieves sub-second inference across all three datasets and different observation densities.

Table 9: Reconstruction performance under delayed sensor measurements on the Active Matter Dataset. A fraction of the observations at t_i is replaced with measurements collected at t_{i-1}. Results are reported as mean $\pm$ standard deviation.

Method	$\rho = 1\%$**, 10% Delay**	$\rho = 3\%$**, 10% Delay**	$\rho = 1\%$**, 30% Delay**	$\rho = 3\%$**, 30% Delay**
LRTFR	0.341 ± 0.097	0.266 ± 0.089	0.372 ± 0.113	0.291 ± 0.105
OFTD	0.557 ± 0.073	0.344 ± 0.066	0.581 ± 0.047	0.372 ± 0.102
CATTE	0.170 ± 0.033	0.166 ± 0.035	0.183 ± 0.047	0.172 ± 0.029
Senseiver	0.2728 ± 0.076	0.244 ± 0.044	0.293 ± 0.067	0.259 ± 0.068
SDIFT	0.234 ± 0.031	0.162 ± 0.038	0.245 ± 0.041	0.173 ± 0.080
StreamPhy	$\mathbf{0.0784 \pm 0.015}$	$\mathbf{0.0756 \pm 0.007}$	$\mathbf{0.081 \pm 0.011}$	$\mathbf{0.0791 \pm 0.009}$

Table 10: Comparison of model size, training time, and inference time across different datasets. "N/A" indicates that the corresponding method does not require a separate offline training stage.

Dataset	**Metric / Setting**	**StreamPhy**	**CATTE**	**OFTD**	**SDIFT**	**LRTFR**	**Senseiver**
Turbulent Flow	Parameters	10.3M	6.5M	5.1M	15M	4.2M	8.3M
	Training time	152 min	N/A	N/A	243 min	N/A	178 min
	Inference time ($\rho = 3\%$)	**0.041 s**	23 min	16.18 s	5.14 s	45.1 s	0.0553 s
	Inference time ($\rho = 5\%$)	**0.044 s**	26 min	17.59 s	5.21 s	53 s	0.0787 s
Ocean Sound Speed	Parameters	9.94M	5.4M	4.9M	15M	3.2M	8.5M
	Training time	154 min	N/A	N/A	209 min	N/A	211 min
	Inference time ($\rho = 1\%$)	**0.032 s**	16 min	6.09 s	0.84 s	27.5 s	0.185 s
	Inference time ($\rho = 3\%$)	**0.043 s**	16.5 min	6.48 s	0.89 s	28.0 s	0.196 s
Active Matter	Parameters	12M	8.2M	–	12M	3.2M	8.2M
	Training time	192 min	N/A	N/A	278 min	N/A	216 min
	Inference time ($\rho = 1\%$)	**0.047 s**	18 min	6.64 s	1.31 s	26.9 s	0.16 s
	Inference time ($\rho = 3\%$)	**0.049 s**	19 min	6.74 s	1.42 s	27.3 s	0.26 s

G Additional Visualizations

H Limitations

A current limitation of StreamPhy is that it does not explicitly enforce physical laws, such as strict boundary conditions or conservation constraints, during the modeling process. This is primarily a deliberate design choice motivated by our target real-world scenarios, where physical priors, boundary conditions, external forcing, or accurate governing equations may be incomplete, unavailable, or difficult to specify from sparse and heterogeneous observations. Therefore, the current framework focuses on learning continuous spatiotemporal representations directly from observational data without requiring a fully specified physical model. Nevertheless, the continuous spatial representation provided by the FT-FiLM decoder offers a natural interface for incorporating physics-based constraints when reliable physical priors are available. Since the decoder represents the physical field as a continuous function of spatial coordinates and time, its derivatives can be obtained via automatic differentiation, allowing PDE residuals, initial/boundary conditions, and conservation laws to be incorporated through PINN-style physics losses.

Another limitation is that the transition matrix $\mathbf{A}$ in the SSM is currently predefined, and its design space remains largely unexplored. In future work, we plan to integrate StreamPhy with domain-specific physical priors to improve the physical consistency and accuracy of long-range, large-scale physical field reconstruction. We also aim to investigate more flexible and potentially learnable parameterizations of $\mathbf{A}$ to further enhance the modeling capacity and performance of the SSM while preserving its ability to capture long-range dependencies.

I Impact Statement

This paper focuses on advancing physical fields modeling techniques. We are mindful of the broader ethical implications associated with technological progress in this field. Although immediate societal impacts may not be evident, we recognize the importance of maintaining ongoing vigilance regarding

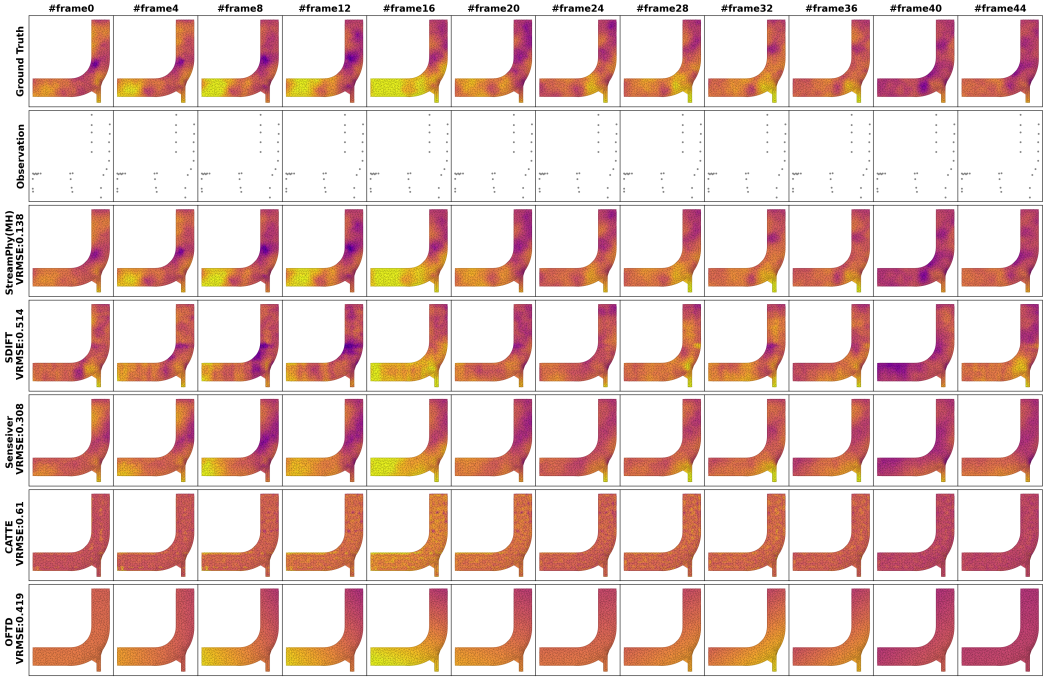

Figure 6: Reconstruction of *Turbulent Flow* dynamics under slab sampling pattern with $\rho = 3\%$.

the ethical use of these advancements. It is crucial to continuously evaluate and address potential implications to ensure responsible development and application in diverse scenarios.

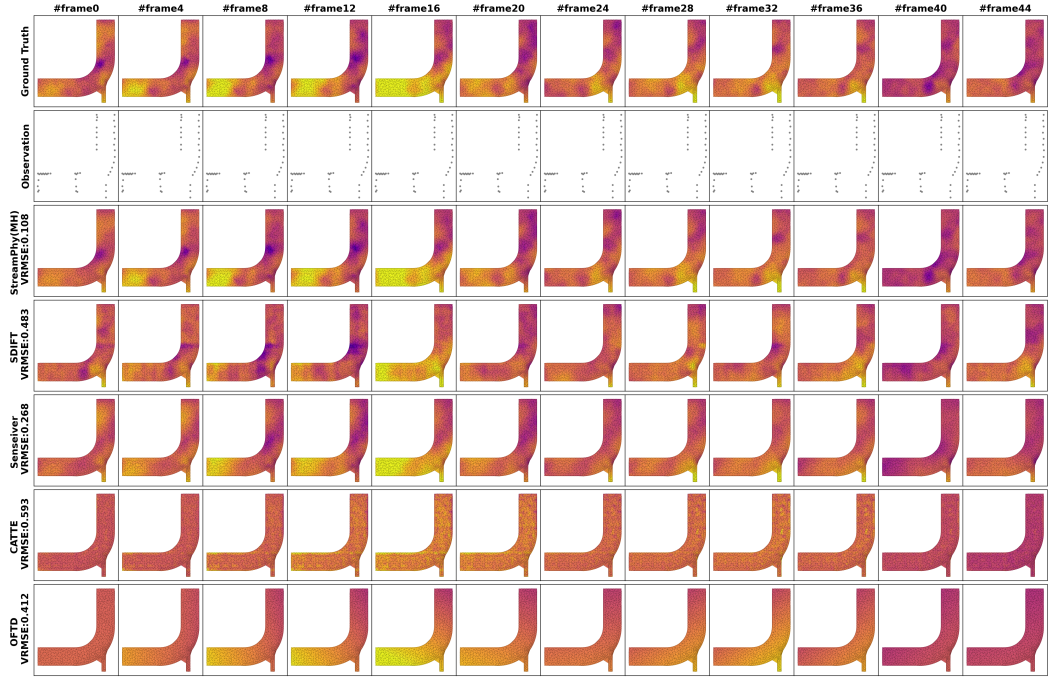

Figure 7: Reconstruction of *Turbulent Flow* dynamics under slab sampling pattern with $\rho = 5\%$.

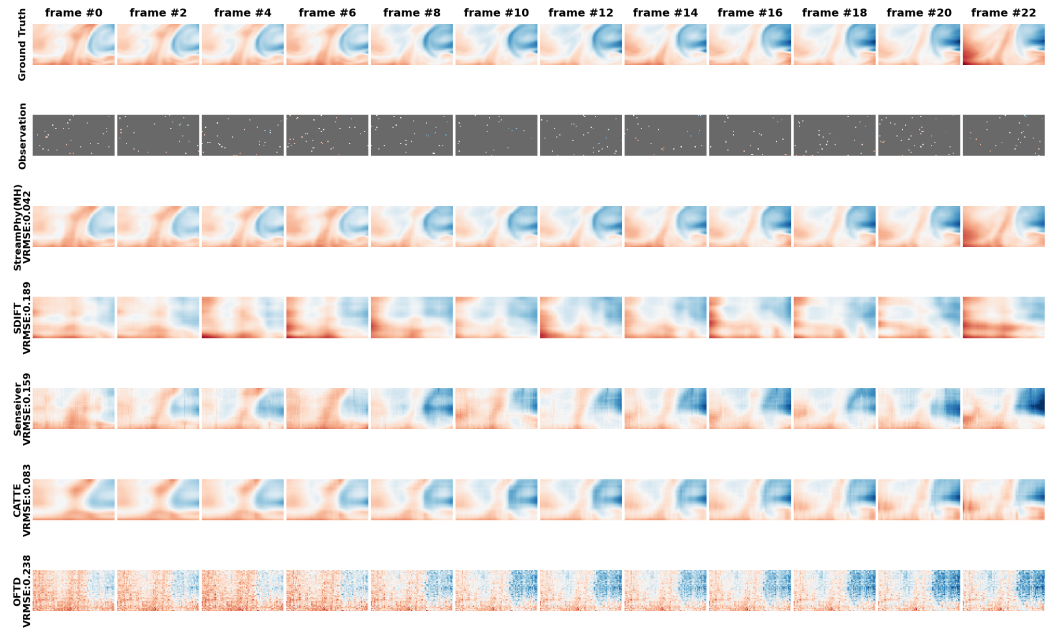

Figure 8: Reconstruction of *Ocean Sound Speed* dynamics under uniform sampling pattern with $\rho = 1\%$.

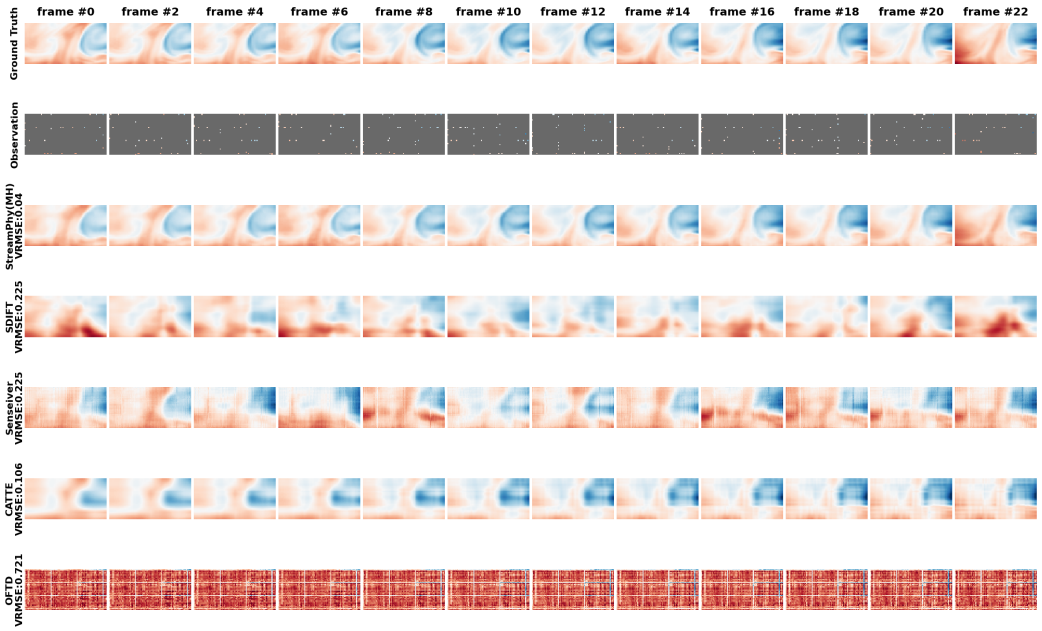

Figure 9: Reconstruction of *Ocean Sound Speed* dynamics under slab sampling pattern with $\rho = 1\%$.

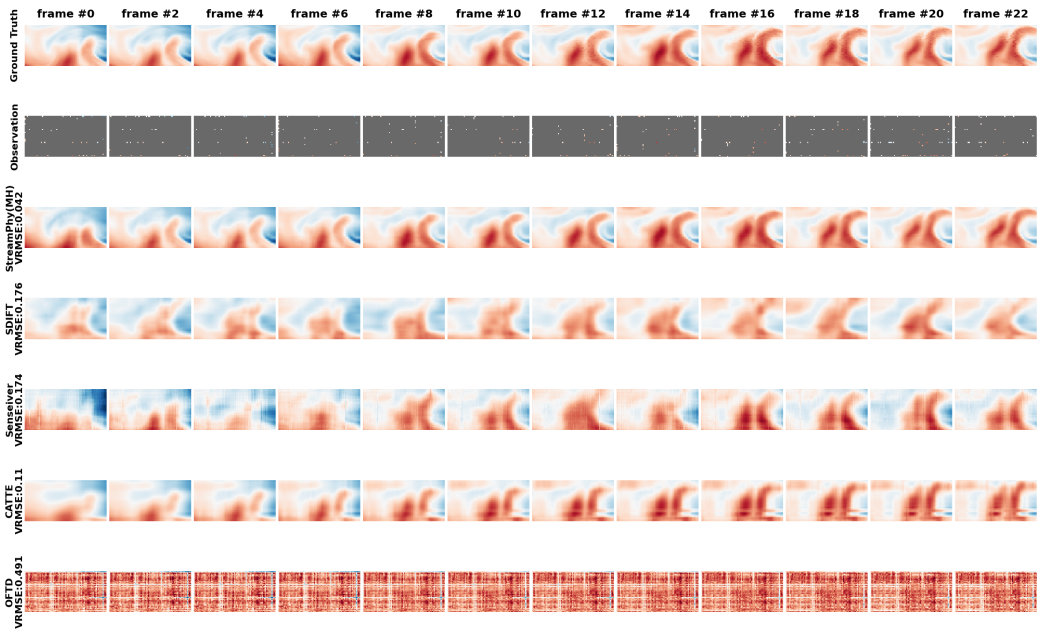

Figure 10: Reconstruction of *Ocean Sound Speed* dynamics under slab sampling pattern with $\rho = 1\%$.

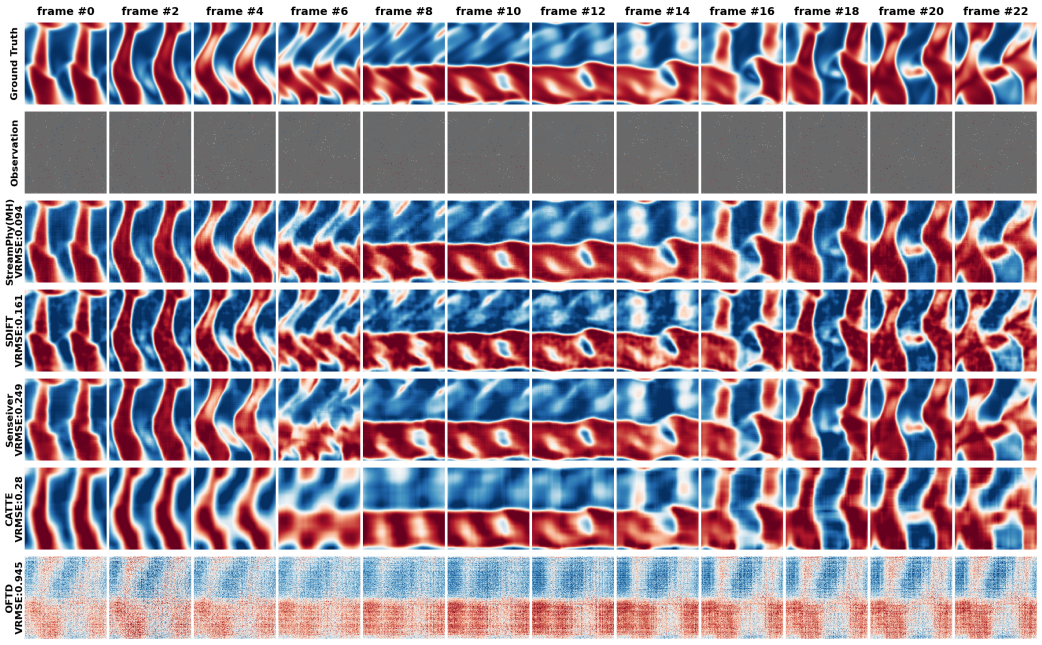

Figure 11: Reconstruction of *Active Matter* dynamics under random sampling pattern with $\rho = 1\%$.