StFT: Spatio-temporal Fourier Transformer for Long-term Dynamics Prediction

Anonymous authors
Paper under double-blind review

Abstract

Simulating the long-term dynamics of multi-scale and multi-physics systems poses a significant challenge in understanding complex phenomena across science and engineering. The complexity arises from the intricate interactions between scales and the interplay of diverse physical processes, which manifest in PDEs through coupled, nonlinear terms that govern the evolution of multiple physical fields across scales. Neural operators have shown potential in short-term prediction of such complex spatio-temporal dynamics; however, achieving stable high-fidelity predictions and providing robust uncertainty quantification over extended time horizons remains an open and unsolved area of research. These limitations often lead to stability degradation with rapid error accumulation, particularly in long-term forecasting of systems characterized by multi-scale behaviors involving dynamics of different orders. To address these challenges, we propose an autoregressive Spatio-temporal Fourier Transformer (StFT), in which each transformer block is designed to learn the system dynamics at a distinct scale through a dual-path architecture that integrates frequency-domain and spatio-temporal representations. By leveraging a structured hierarchy of StFT blocks, the resulting model explicitly captures the underlying dynamics across both macro- and micro- spatial scales. Furthermore, a generative residual correction mechanism is introduced to learn a probabilistic refinement temporally while simultaneously quantifying prediction uncertainties, enhancing both the accuracy and reliability of long-term probabilistic forecasting. Evaluations conducted on three benchmark datasets (plasma, fluid, and atmospheric dynamics) demonstrate the advantages of our approach over state-of-the-art ML methods.

1 Introduction

Predicting the long-term spatio-temporal dynamics of systems governed by partial differential equations (PDEs) is a cornerstone of scientific and engineering research, with broad applications in fields such as earth system modeling, plasma science, fluid dynamics, and beyond. Traditional approaches rely heavily on numerical solvers, which discretize the domain and iteratively solve PDEs using methods including finite element, finite volume and spectral methods (Tadmor, 2012). While effective in many scenarios, these techniques face limitations when applied to multiphysics systems characterized by complex dynamics and multiscale behaviors. They require substantial computational resources and exhibit poor scalability with increasing problem size, rendering them impractical for high-dimensional, large-scale, or long-term physics systems due to excessive computational costs and memory demands.

Recent advances in deep learning have revolutionized the field of PDE modeling by introducing data-driven methodologies that significantly accelerate computations for science while maintaining high accuracy. Inspired by the universal approximation theorem (Chen & Chen, 1995), neural operators that learn the mapping between two function spaces have demonstrated great success in simulating various PDE systems across multiple scientific disciplines without retraining for new conditions (Li et al., 2020; Lu et al., 2021). Building on the success of transformers in natural language processing and computer vision (Vaswani, 2017; Dosovitskiy, 2020), transformer-based neural operators process multiple input functions while enabling arbitrary querying of output function locations, offering enhanced flexibility in handling complex functional mappings (Hao et al., 2023; Li et al., 2022). A series of neural operators have been developed to address complex scientific problems, including weather forecasting, turbulent fluid dynamics, and boiling phenomena (Pathak et al., 2022; Li et al., 2023a; Bi et al., 2023; Hassan et al., 2023; Lin et al., 2021).

Despite the success of these methods, accurate and long-term predictions of complex physical systems remain challenging, primarily due to the requirements for numerical stability, high-fidelity modeling, and reliable uncertainty quantification over extended horizons. The inherent multi-scale nature and multi-physics complexity of such systems necessitate methodologies that can efficiently represent and integrate dynamics across disparate spatial and temporal scales while simultaneously capturing the complex interactions between distinct physical processes, such as the influence of micro-scale turbulence on macro-scale flow in fluids and combustion (Peters, 2009; Natrajan et al., 2007). For large-scale atmospheric pressure systems, high-pressure ridges and low-pressure troughs play a crucial role in shaping local weather patterns; inaccurate representation of those structures can cause significant errors in forecasting rainfall, wind speed, and temperature (Wang et al., 2006; Barlow et al., 2019). In magnetically confined plasmas, multiphysics arises from the coupling of physical processes that govern plasma behavior, including electromagnetic fields, turbulence transport, thermodynamics, and particle interactions that are potentially coupled with kinetic models in high-fidelity simulations. The magnetohydrodynamic (MHD) instabilities caused by current or pressure gradients can limit burning plasma performance, and threaten fusion device integrity (von Goeler et al., 1974; Graves et al., 2012; Seo et al., 2024). Furthermore, integrating uncertainty quantification (UQ) into modeling frameworks is essential for assessing the confidence and reliability of predictions in such complex systems (Cheung et al., 2011; Scher & Messori, 2018; Kruger et al., 2024). Although neural operators present advantages over traditional approaches, they still encounter challenges associated with the demands for scientific fidelity and stability, especially when the underlying physics exhibit rapid changes or high-frequency components. These issues are further intensified in high-resolution simulations of multi-scale scenarios. Recent efforts to address these limitations include P^2C^2Net, which encodes a high-order numerical scheme with boundary condition encoding into neural networks (Wang et al., 2024a), and Dyffusion, which trains a forecasting network and an interpolation network that allows for continuous time sampling and multi-step prediction for long-range forecasting (Rühling Cachay et al., 2023). However, most existing neural operators lack built-in mechanisms for uncertainty quantification, which is particularly critical for reliable modeling of long-term dynamics, where even small errors can propagate across scales and result in significant inaccuracies.

Existing approaches for predicting spatio-temporal dynamics can be broadly be classified into two primary categories. The first category comprises models that directly forecast future states at fixed time horizons using a sequence of past observations (Wang et al., 2024b; Kontolati et al., 2024). The second category includes models that utilize an autoregressive manner, which addresses challenges of scaling and fitting complexities as a continuous-time emulator (Pathak et al., 2022; Rühling Cachay et al., 2023; Lippe et al., 2023; McCabe et al., 2023). Generally, prediction errors incurred in the short term can accumulate, leading to instability and reduced accuracy in long-term forecasts. To mitigate these issues, previous work has proposed techniques such as the pushforward trick, invariance preservation, and iterative refinement (McCabe et al., 2023; Lippe et al., 2023; Brandstetter et al., 2022). Nevertheless, the development of multi-scale modeling frameworks for long-term dynamic prediction remains crucial for capturing the interactions across scales and enhancing prediction accuracy. Concurrently, incorporating uncertainty quantification is critical for identifying the spatial and temporal regions where predictive confidence deteriorates, especially in complex systems where localized uncertainties can influence global dynamics over time.

In this work, we introduce an autoregressive **Spatio-temporal Fourier Transformer (StFT)**, for long-range forecasting of multi-scale and multi-physics systems. At each level of the spatial hierarchy, one StFT block models the physical dynamics associated with a distinct spatial scale or receptive field, as inferred from the spatiotemporal data. Each StFT block adopts a dual-path architecture: (1) **the frequency path** captures large-scale dynamics by operating in the Fourier domain, focusing on low-frequency components that are critical for modeling global behavior. (2) **the spatio-temporal path** operates in the full physical space incorporating all spatio-temporal features to capture fine-scale features. Through a hierarchical composition of StFT blocks across multiple scales, augmented by a generative residual correction block, the resulting model learns the intricate interactions both within the same scales and across different scales. Moreover, it produces uncertainty estimates at each spatial and temporal point, enabling assessment of prediction confidence throughout the forecast. The cascading StFT blocks enable our model to predict high-resolution dynamics across a spectrum of varying scales in correlated physical processes. By integrating StFT within an auto-regressive framework, our method achieves superior accuracy in long-term predictions compared to existing state-of-the-art autoregressive baselines. Our contributions are summarized as follows:

- We propose Spatio-temporal Fourier transformer (StFT), a novel ML model that learns underlying dynamics across spatial scales for multi-physics systems via a dual-path (frequency and spatio-temporal path) architecture, which effectively captures both the global, large-scale structures and local, fine-scale features.
- We propose StFT-F, which incorporates a probabilistic residual correction mechanism to refine the forecasting of StFT temporally and provide pointwise uncertainty quantification.
- We propose an overlapping tokenizer and a detokenizer that share regions between adjacent patches, improving spatial smoothing and reducing discontinuity artifacts.
- We demonstrate the effectiveness of StFT in an autoregressive framework on a diverse set of applications including the plasma, fluid, and atmospheric dynamics. Evaluating performance across variables, StFT outperforms the best baselines. Its probabilistic variant, StFT-F further improves average forecasting accuracy by 5% and produces uncertainty estimates that are empirically calibrated, as demonstrated through confidence-based evaluation.

2 Related Work

Neural Operators. Neural operator architectures and their variants have been proposed, including Fourier neural operators (Li et al., 2020; Gupta et al., 2021; Tran et al., 2021; Cao et al., 2023; Li et al., 2023b; Rahman et al., 2022), DeepONet (Lu et al., 2021; Wang et al., 2021; Jin et al., 2022; Wang et al., 2022; Kontolati et al., 2024; Prasthofer et al., 2022), transformer based operators (Hao et al., 2023; Cao, 2021; Li et al., 2022), and image-to-image operators (Gupta & Brandstetter, 2022; Long et al., 2024). U-Net, a fundamentally hierarchical structure model, which has inspired several neural operators (Rahman et al., 2022; Liu et al., 2022b; Gupta & Brandstetter, 2022), allows solutions to multi-scale PDEs by hierarchically aggregating feature representations of progressively coarser spatial resolutions. Recent work in computer vision (Liu et al., 2021; Fan et al., 2021; Zhang et al., 2022) have introduced methods for extracting multi-scale features through hierarchical architectures. However, these hierarchical models do not explicitly model or forecast the multi-scale structures of physical processes, which limit the transparency and interpretability of their representations across scales. In contrast, our method begins with a coarse approximation that captures large-scale, low-frequency phenomena, and incrementally refines the representation over layers to resolve finer details. These structured decompositions enable error diagnosis, enhance interpretability of model performance of different scales, and allow for targeted improvements with an explicit refinement mechanism.

Generative Models. Generative models, especially diffusion models have demonstrated success in various domains, including vision, audio, robotics (Ho et al., 2020; Song et al., 2020; Tian et al., 2024; Kong et al., 2020; Wolleb et al., 2022), and relevance to spatio-temporal dynamics prediction (Ho et al., 2022; Voleti et al., 2022; Singer et al., 2022; Rühling Cachay et al., 2023). As an alternative approach in generative modeling, flow matching has been introduced to support efficient sampling and has since been applied to video generation (Lipman et al., 2022; Liu et al., 2022a; Albergo & Vanden-Eijnden, 2022; Polyak et al., 2024; Esser et al., 2024). While video generation typically explores a range of creative and diverse possibilities from text or image prompts, forecasting spatial-temporal dynamics driven by PDEs necessitates more than mere statistical resemblance - it requires each prediction is firmly grounded in the underlying physics. To achieve accurate forecasting while capturing the inherent stochasticity of physical processes, our work incorporates a flow matching block following the proceeding StFT blocks. This enables the model to align its prediction distribution with the underlying physical dynamics and generate calibrated uncertainty estimates via confidence-based metrics.

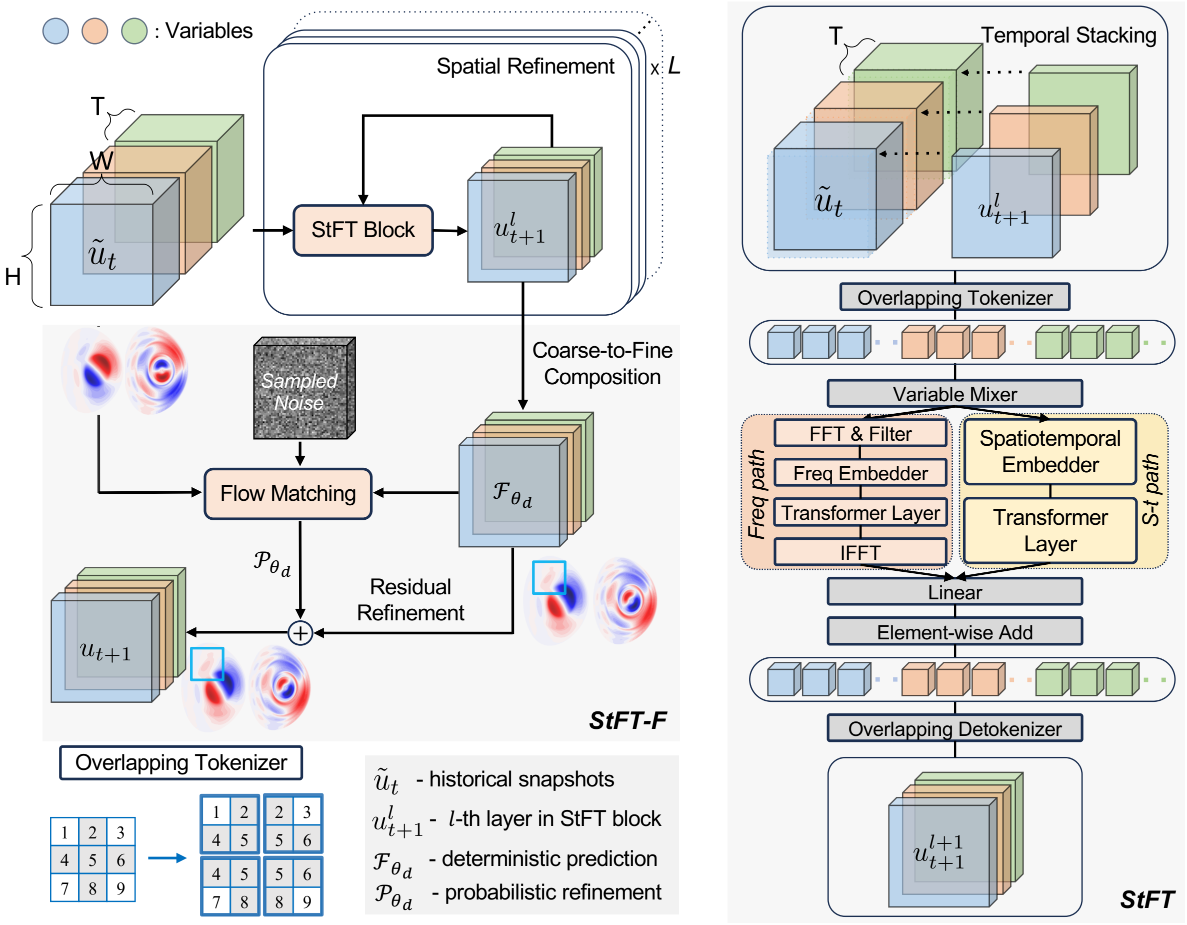

Figure 1: Left: **Overview of the proposed StFT and StFT-F model**. The model predicts u_{t+1} using the past k snapshots $\tilde{u}_t$, employing L spatial refinements from coarse to fine scales through the proposed Spatiotemporal Fourier Transformer (StFT) blocks. Bottom left: an illustration of the overlapping tokenizer, where the patch size is 2×2, and the overlapping number is 1×1. Right: **Illustration of the proposed StFT block**. First, the past snapshots $\tilde{u}_t$ and a coarser prediction from the previous layer are temporally stacked. The stacked discretizations are passed to the overlapping tokenizer to generate tokens for each variable. Next, tokens corresponding to different variables in the same spatial are mixed through a variable mixer. Two paths of transformation, frequency path and spatiotemporal path, process the frequency embeddings and the spatiotemporal embeddings respectively. The finer prediction for timestamp $t+1$ is obtained after passing through the overlapping detokenizer.

3 Method

Formulation. We consider an autoregressive formulation for long-term multi-scale spatiotemporal physical processes. We define a vector $\tilde{u}_t$ representing the historical snapshots of the multi-physics variables at timestamps from $t-k+1$ to t, in a total of k snapshots of $\tilde{u}_t = [u_t, u_{t-1}, \ldots, u_{t-k+1}]$ specifically. We formulate the probabilistic one-step forward neural operator StFT-F as

$$u_{t+1} = \mathcal{F}_{\theta_d}(\tilde{u}_t) + r_{t+1}, r_{t+1} \sim \mathcal{P}_{\theta_g}(r|\tilde{u}_t, \mathcal{F}_{\theta_d}(\tilde{u}_t)), \tag{1}$$

where $\mathcal{F}_{\theta_d}$ denote the StFT operator, a deterministic forecasting parameterized by θ_d, and $\mathcal{P}_{\theta_g}$ is the generative flow matching block parameterized by θ_g for refining the forecasting of StFT while quantifying uncertainty estimates. $\mathcal{F}_{\theta_d}$ represents the deterministic evolution of the system that encapsulates the dynamics from multi-scale spatial refinement in StFT. The residual refinement $\mathcal{P}_{\theta_g}(r|\tilde{u}_t, \mathcal{F}_{\theta_d}(\tilde{u}_t))$ captures the probabilistic nature of the residual from the generative model. It represents a prediction distribution conditioned on the current state $\tilde{u}_t$ and the deterministic prediction $\mathcal{F}_{\theta_d}(\tilde{u}_t)$, modeling the uncertainty or variations that missed by the deterministic component. The residual r_{t+1} calculates deviations from the deterministic prediction, and its distribution allows the model to account for noise or inherent stochasticity in the physical processes. Therefore, by sampling residual r_t, our model learns stochastic trajectories from data. Besides providing prediction uncertainties, these stochastic trajectories can help study the long-term behavior, stabilities, and bifurcations in stochastic systems (Lucor et al., 2003; Krämer et al., 2022).

Figure 1 presents the overview of the StFT-F, the overlapping tokenizer, and the design of the StFT block. Algorithm 1 and 2 detail the design of our model. The following subsections introduce StFT with the overlapping tokenizer/detokenizer and the residual correction mechanism based on flow matching.

3.1 StFT Block

Overlapping tokenizer. Each StFT block first tokenizes the discretized functions $\tilde{u}_t$ of shape $T \times W \times H \times C$, where T is the temporal dimension, W, H are the spatial dimensions, and C denotes the number of physical variables. We apply tokenization along the spatial dimensions for each variable. To enhance spatial continuity while minimizing visual artifacts, we propose to use an overlapping tokenizer (OLT) and detokenizer (OLDT) that allows adjacent patches to share boundaries through overlapping regions. For instance, as shown in Figure 1, a 3×3 input generates four 2×2 patches with a 1×1 overlap, where the overlapping areas (indicated in gray) are shared between patches. During detokenization, overlapping regions are reconstructed by averaging the corresponding values from neighboring patches. This strategy effectively mitigates discontinuity issues at the patch boundaries, which is particularly important for accurately representing smooth and continuous target functions. Furthermore, incorporating shared boundaries into patch embeddings enriches feature representation and extraction from fine-scale structures.

Variable mixer. Each StFT block is designed to handle a specific scale; therefore, by employing a specific patch size, we partition the input at a corresponding level of granularity. To ensure that the first block captures the coarsest features or the largest scale, we set the patch size $p_w^1 \times p_h^1$ to a large value, allowing it to model broad spatial structures effectively. As a result, $\mathcal{O}(\frac{W}{p_w^1} \times \frac{H}{P_h^1} \times C)$ patches are fed into the variable mixer, where patches corresponding to different physical processes but sharing the same spatial domain are mixed into a single token. Following this step, two transformation paths are performed: one is in the spatio-temporal domain, which operates self-attention on spatio-temporal embeddings; the other is in the frequency domain, which operates on frequency embeddings.

Frequency embeddings. The tokens are first processed by a 2D/3D Fourier transform, where only low-frequency components are retained. These low-frequency components are then passed through a frequency embedder to obtain frequency embeddings f_t. Subsequently, these frequency embeddings are fed to the standard transformer layers for mixing information and nonlinear transformation in the frequency domain. Finally, an inverse 2D/3D Fourier transform and a linear projection are applied to map the frequency embeddings back to the spatio-temporal domain.

Spatio-temporal embeddings. The same set of tokens first pass through a spatio-temporal embedder, after which the spatio-temporal embeddings e_t are processed by multiple standard transformer layers for mixing correlations and nonlinear transformation. Finally a separate linear projection is applied to get predictions for each patch of u_{t+1}.

Next, an overlapping detokenizer yields the first block prediction u_{t+1}^1. Each token represents a significant portion of the historical snapshots, encapsulating macroscopic structural features. This coarse-level partitioning reduces the complexity of modeling fine-grained details. By maintaining a lower granularity, the model prioritizes structural coherence over extraneous details, enabling it to focus on capturing and predicting global relationships between regions more effectively.

3.2 A Hierarchy of StFT Blocks

In the subsequent StFT blocks, we shift our focus to smaller scales with fine details. Consequently, we concatenate the prediction u_{t+1}^1 with the input $\tilde{u}_t$, and consider this combination as the input for the next StFT block. We further subdivide each patch from the previous StFT block into smaller patches.

The smaller patches allow for less information to be aggregated within a single patch, thereby minimizing the risk of losing local variations and enhancing the richness and informativeness of the fine-scale representation. By leveraging the finer granularity of the patches to focus on smaller regions, it allows the model to better localize features and capture their details. In addition, conditioning on the coarser prediction allows the models to iteratively refine its estimates, beginning with a broad global summary of the corresponding regions. Through the repeated subdivision of the patches, the model progressively refines its predictions across multiple scales. As shown in Figure 1, the StFT model predicts u_{t+1} and applies a total of L multi-scale spatial refinements.

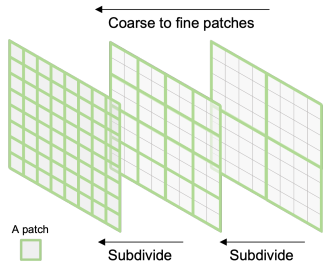

Figure 2: A patch is subdivided into smaller patches for a hierarchical learning.

3.3 Residual Refinement and Uncertainty Estimation Based on Flow Matching

Finally, the model refines its deterministic predictions through a rectified flow block, which belongs to the family of flow matching models (Liu, 2022; Liu et al., 2022a). Flow matching is formulated as an ordinary differential equation in time $\tau \in [0, 1]$, $\frac{d}{d\tau}\psi_\tau(x) = \nu_\tau(\psi_\tau(x))$, where the learnable velocity field ν_τ directs the transformation of each sample X_0 from a source distribution p_0, typically a Gaussian distribution, toward the target distribution $X_1 \sim p_1$ with p_1 representing the data distribution. If we prescribe the velocity field ν_τ such that it guides every sample along a straight-line trajectory from X_0 to X_1, it is referred to as a rectified flow. In this case, X_τ represents the linear interpolation across the entire timespan between X_0 and X_1, which can be expressed as $X_\tau = \tau X_1 + (1-\tau)X_0$. We employ a parameterized $\mathcal{M}_{\theta_g}$ to approximate ν_τ, leading to the following learning objective: $\mathcal{L}(\theta_g) = \mathbb{E}_{\tau, X_0, X_1} \left\| \mathcal{M}_{\theta_g}(X_\tau, \tau) - (X_1 - X_0) \right\|^2$. In our model, the rectified flow block takes the deterministic prediction u_{t+1} from the composition of L StFT blocks and the observations $\tilde{u}_t$ as conditioning inputs. Its objective is to generate the distribution of residuals $r_{t+1} = y - \sum_j u^j_{t+1}$, where y is the ground truth for the solution at $t+1$. Our training loss then becomes:

$$\mathbb{E}_{X_0 \sim \mathbf{N}(0,\mathbf{I}), \tau \sim (0,1)} [(\mathcal{M}_{\theta_g}(\tilde{u}_t, \mathcal{F}_{\theta_d}(\tilde{u}_t), \tau, X_\tau) - (r_{t+1} - X_0))^2], \tag{2}$$

where X_τ is the linear interpolation between the source sample X_0 and the target r_{t+1}.

```
Algorithm 1: StFT
Inputs: history $\tilde{u}_t = (u_t, \ldots, u_{t-k+1})$
Initialize: blocks $l \in [1, L]$, patch sizes $p_{h_l,w_l}$,
  truncation modes $m_{h_l,w_l}$, overlaps $o_{h_l,w_l}$, $u^0_{t+1}$ as
  None, $u_{t+1}$ as $\mathbf{0}$
$v \leftarrow$ var idx
for $l = 1, \ldots, L$ do
    $x_t \leftarrow$ TemporalStacking($\tilde{u}_t$, $u^{l-1}_{t+1}$);
    $\{x_t\}_v \leftarrow$ OLT($x_t$, $p_{h,w}$, $o_{h,w}$);
    $\{x_{t+1}\}_v$ : Invoke Freq. & S-t. paths
    $u^l_{t+1} \leftarrow$ OLDT($\{x_{t+1}\}_v$, $p_{h,w}$, $o_{h,w}$)
    $u_{t+1} \leftarrow u_{t+1} + u^l_{t+1}$
Return: $u_{t+1}$
```

```
Algorithm 2: StFT block: Freq. and S-t. paths
Frequency Path: $\bar{x}_{t,1} \leftarrow$ VariableMixer$^1$($\{x_t\}_v$)
$f_t \leftarrow$ FFTFilter($\bar{x}_{t,1}, m_h, m_w$)
$f_{i,t} \leftarrow$ FreqEmbedder($f_{i,t}$)
$f_t \leftarrow$ TransformerBlock$^1$($f_t$)
$\{x_{t+1,1}\}_v \leftarrow$ iFFT($f_t, m_h, m_w$)
$\{x_{t+1,1}\}_v \leftarrow$ Linear$^1$($\{x_{t+1,1}\}_v$)
Spatiotemporal Path:
  $\bar{x}_{t,2} \leftarrow$ VariableMixer$^2$($\{x_t\}_v$)
$e_t \leftarrow$ StEmbedder($\bar{x}_{t,2}$)
$e_t \leftarrow$ TransformerBlock$^2$($e_t$)
$\{x_{t+1,2}\}_v \leftarrow$ Linear$^2$($e_t$)
Merge: $\{x_{t+1}\}_v \leftarrow \{x_{t+1,1}\}_v + \{x_{t+1,2}\}_v$
```

4 Experiments

4.1 Datasets

In this section, we consider three spatio-temporal multi-physics systems arising from time-dependent PDEs of a variety of complexities, including a high-dimensional plasma dynamics system based on reconstructed equilibrium of DIII-D experimental discharges, a 2D incompressible Navier-Stokes equation in velocity-pressure form within a square domain driven by an external force, and a viscous shallow-water equation modeling the dynamics of large-scale atmospheric flows on a spherical domain. The problem setup and data generation are detailed in Appendix B.

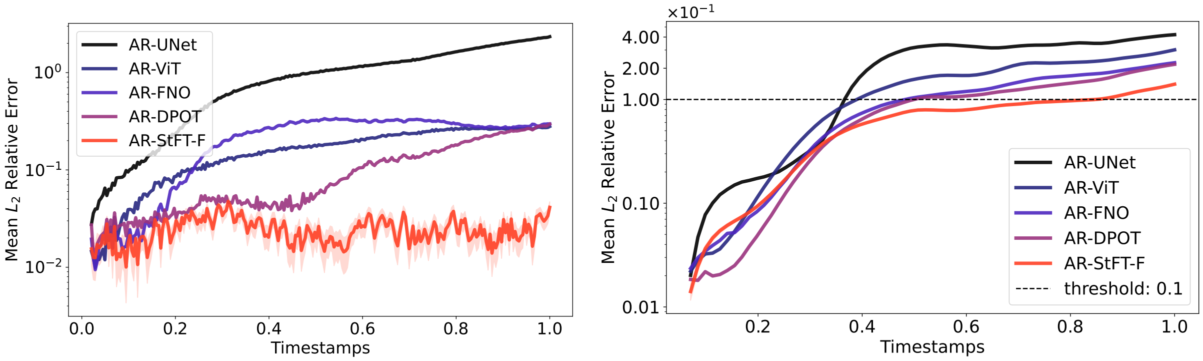

Figure 3: Results of autoregressive prediction in L_2 relative error (log scale) across the timespan: (left) perturbed parallel vector potential $\delta A_{\parallel}$ in plasma MHD, and (right) magnitude of velocity in shallow-water equation. The shaded region indicates the uncertainty distribution of σ in the relative error of StFT-F. For a given error threshold, StFT-F maintains accuracy over at least twice the time horizon compared to the baselines.

Table 1: Quantitative results of our model and baselines in the same autoregressive framework: relative L_2 error over three spatiotemporal prediction systems. AR-StFT refers to the deterministic model's results. AR-StFT-F denotes the probabilistic model with residual refinement of StFT. All models have been subjected to hyperparameter tuning to ensure fair and optimal performance comparisons.

Dataset	Variable(s)	AR-StFT	AR-StFT-F	AR-DPOT	AR-FNO	AR-ViT	AR-UNet$_b$
Plasma MHD	$\delta\phi$	2.80e-2	2.24e-2	1.04e-1	2.28e-1	1.73e-1	1.02e0
	$\delta A_{\parallel}$	2.45e-2	2.30e-2	8.36e-2	2.30e-1	3.24e-1	8.13e-1
	$\delta B_{\parallel}$	3.05e-2	2.66e-2	8.98e-2	2.33e-1	1.95e-1	7.79e-1
	δn_e	2.84e-2	2.45e-2	8.64e-2	2.33e-1	2.08e-1	1.01e0
	δn_i	3.28e-2	2.93e-2	8.76e-2	2.33e-1	2.18e-1	1.04e0
	δu_e	3.99e-2	3.73e-2	9.59e-2	3.18e-1	2.99e-1	6.96e-1
Navier-Stokes	u	3.38e-2	3.30e-2	4.67e-2	4.46e-2	5.09e-2	6.16e-2
	v	3.60e-2	3.17e-2	4.52e-2	4.57e-2	4.60e-2	6.15e-2
	p	5.16e-2	4.44e-2	6.18e-2	5.90e-2	7.03e-2	7.84e-2
Shallow-Water	$\mathbf{V}$	6.25e-2	6.53e-2	7.97e-2	9.53e-2	1.33e-1	2.02e-1

4.2 Experimental Setup

Long-term multi-physics prediction up to a horizon of 244 timesteps. Our goal is to simulate long-time trajectories given a few initial observations. This task is particularly challenging due to the multiple correlated variables present in the Navier-Stokes and plasma magnetohydrodynamics (MHD), with the test trajectories consisting of snapshots that vary from 71 to 244. We employ an autoregressive framework for all the methods: during training, each model utilizes five historical snapshots to predict the next one in a forward pass. At test time, given the initial five snapshots of a trajectory, all models autoregressively generate the entire trajectories by iteratively predicting future states based on their own previous outputs.

Baseline setup. We evaluate these datasets using the following well-known and state-of-the-art methods for comparison: autoregressive AFNO, autoregressive Fourier Neural Operator (AR-FNO), autoregressive U-Net (AR-UNet), and autoregressive vision transformer (AR-ViT). More specifically, for AFNO, we used the latest variant from the DPOT work (Hao et al., 2024), which is enhanced with a temporal aggregation layer and improved expressivity through the removal of enforced sparsity. For FNO, we use the authors' open-source implementation. For U-Net, we employ the implementation of the modified U-Net as evaluated in the recent BubbleML work (Hassan et al., 2023), where the modified U-Net was initially used in PDEBench work (Takamoto et al., 2022), and demonstrated superior performance over their baselines.

Validation and hyperparameter tuning. We divide the trajectories of each dataset into training, test, and validation sets. For each method, we identify the tunable hyperparameters, specify a range for each hyperparameter, and conduct a grid search. For our methods, we implement both StFT, the deterministic component of our model, and StFT-F, our model with a generative residual refinement block. We use the AdamW optimizer with a learning rate of $1e^{-4}$ to train those models on an A100 GPU. We ensure that all models are fairly and thoroughly trained by imposing an identical time budget across all models. More specifically, we impose 24 GPU hours on the plasma MHD and Navier-Stokes datasets, and 48 GPU hours on the shallow-water dataset. A comprehensive list of all the hyperparameters along with their respective ranges is provided in the Appendix F.

Evaluation metrics. First, we evaluate the forecasting performance by calculating the mean L_2 relative error. For StFT-F, in order to obtain the mean prediction, we generate 50 stochastic predictions at each autoregressive step, and then inject their mean into the next step. Additionally, we assess the uncertainty quantification capability of StFT-F by sampling 100 trajectories for each test case. For each trajectory, at each autoregressive step, we generate a single prediction, which is subsequently fed into the next autoregressive step to iteratively forecast the full sequence.

4.3 Main Results

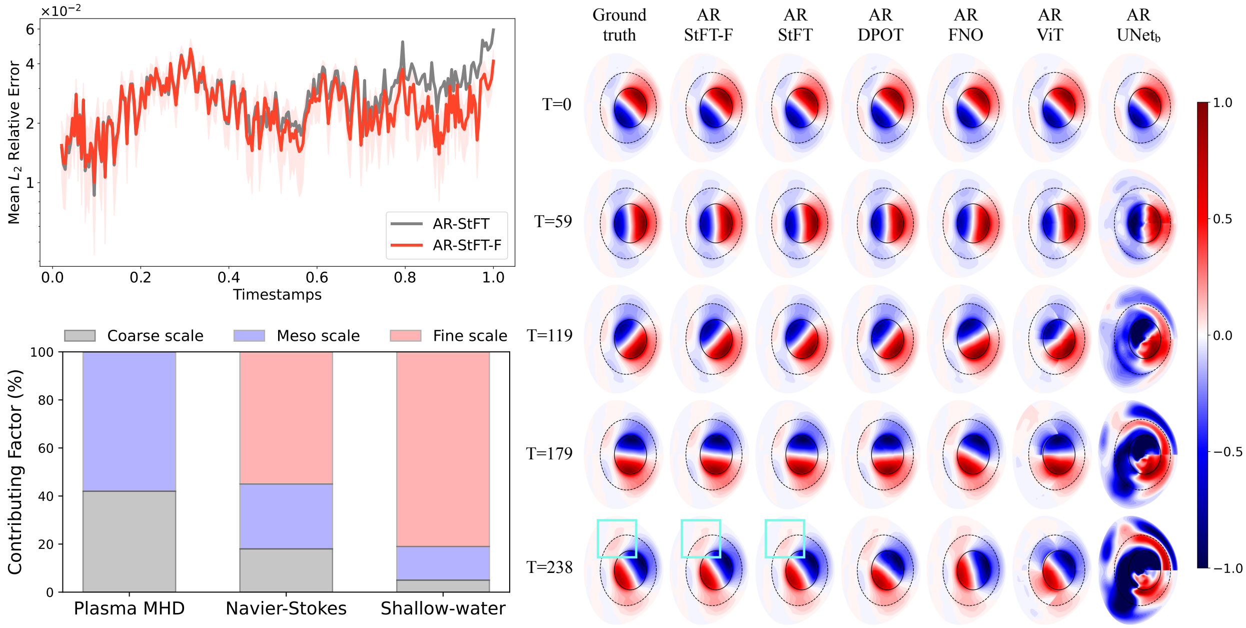

Figure 4: Left top: Comparing StFT and StFT-F in L_2 relative error across timestamps for two plasma MHD variables with uncertainty bands of StFT-F. Left bottom: The contribution of each StFT block. From bottom to top, the patch size decreases. For this plasma data, the first two levels are sufficient to capture the multi-scale structures, whereas for Navier-Stokes and shallow-water equations, the finest scale contributes more significantly. Right: Spatiotemporal evolution of perturbed electrostatic potential $\delta\phi$ predicted by autoregressive models. StFT and StFT-F remain accurate past $T = 59$, whereas baseline methods exhibit out-of-phase predictions.

Table 1 shows the forecasting performance of all the models on the three applications. We present several test trajectories and visualize their uncertainties across the forecasting time horizon estimated by StFT-F, as illustrated in Figure 5. StFT performs significantly better than all other baselines across all physical processes in the three applications. In the plasma MHD dataset, the test trajectory has a total of six coupled physics variables and 244 snapshots. On average, StFT achieves a reduction in error by a factor of three compared to the best baseline, AR-DPOT(AFNO). Although AR-FNO maintains a high resolution in its long-term prediction, it fails to capture the correct dynamics of mode evolution, leading to out-of-phase predictions as shown in Figure 4.

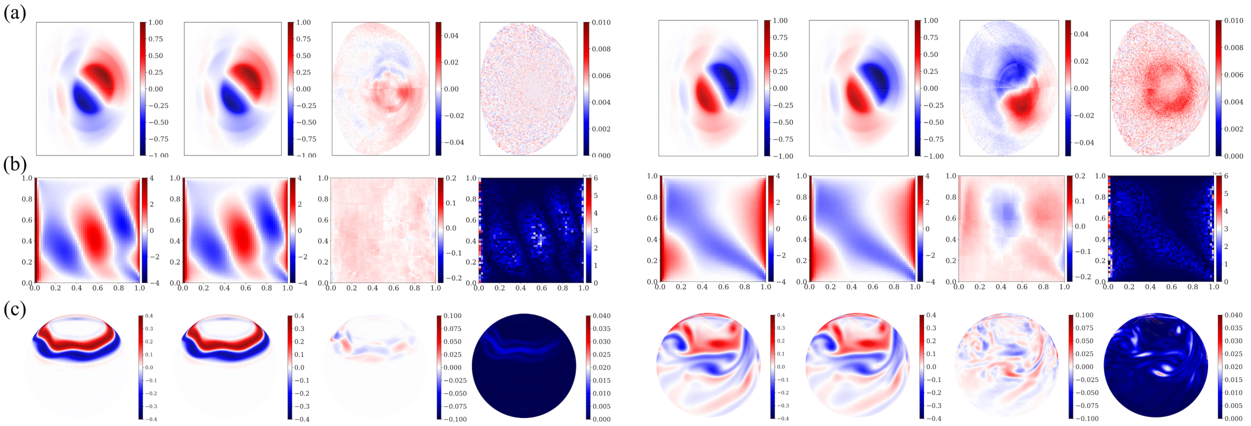

Figure 5: Evaluation of forecasting for three applications: ground truth, StFT-F prediction, residual, and uncertainty over time — shown for initial (left) and final (right) states. Variables include: (a) $\delta\phi$ in plasma MHD, (b) u in Navier-Stokes, and (c) $\mathbf{V}$ in shallow-water equations.

We examine the error growth by plotting the L_2 relative errors over time, as illustrated in Figure 3 for several representative variables. In the bottom figure for the shallow-water dataset, AR-FNO first appears to slightly better than all other methods during the short term from timestamp 0 to 0.2, and StFT-F shows superior performance soon after. For the plasma dataset, StFT-F demonstrates dominance starting from $t = 0.2$ with a stable performance, while the errors of all baseline methods begin to propagate from that point onward, resulting in a rapid decline compared to StFT-F. Notably, StFT-F exhibits long-term stability relative to the other methods. As shown in Figure 4, both of our methods accurately capture the dominant mode phase, and StFT-F generates predictions that are more closely aligned with the ground truth compared to StFT. We also compare the error over time comparing StFT and StFT-F. StFT-F begins to prevail from $t = 0.6$. For the shallow-water and Navier-Stokes datasets, on average, StFT reduces the errors by an average of 27% and 25%, respectively. In Navier-Stokes and plasma datasets, StFT-F not only surpasses StFT but also offers the additional capability of uncertainty quantification, achieving error reductions of 10% in both cases. In the shallow-water dataset, we observe a slight increase in error with StFT-F. These results indicate that our method achieves superior long-term stability and accuracy among all other methods. More visualizations and results are included in Appendix G.

4.4 Uncertainty Quantification

Table 2: Average confidence interval (90% and 95%) coverage.

Dataset	CI: 90%	CI: 95%
Plasma MHD	89.4%	92.5%
Shallow-Water	89.5%	98.0%

Figure 5 presents the distribution of the empirical standard deviation along with the mean prediction. As observed, regions with large errors correspond to those exhibiting significant uncertainties predicted by StFT-F. Additionally, it is evident that uncertainties increase with time. This aligns with our expectation, as errors accumulate during the autoregressive forecasting process. Besides the empirical evaluation relying on the standard deviation plots, we further measure the robustness of StFT's uncertainty quantification using confidence intervals as shown in Table 2. Specifically, we use the predicted uncertainty to compute empirical coverage by measuring the proportion of ground truth values that fall within a certain confidence interval around the predicted mean. This provides a more rigorous evaluation of StFT's ability to capture uncertainties. Details about CI coverage and results regarding each variable are included in Appendix E. For the 90% confidence interval, average coverage is very close to ideal value of 90%, demonstrating that StFT-F is well-calibrated around the 90% confidence interval.

4.5 Ablation Study

Multi-scale and frequency path. To evaluate the effectiveness of the hierarchical structure and the frequency path in StFT, we conduct an ablation study employing mono-scale versus multi-scale models, and employing models with and without the frequency path. Experimental results are shown in Table 3a.

Table 3: Ablation study of StFT. (a) Evaluates model structure and frequency path. (b) Shows performance impact of using more hierarchical scales.

(a) Effect of mono-/multi-scale structures and frequency path $\mathcal{F}$.

Model	**Plasma MHD**	**Shallow-Water**
Mono-scale + $\mathcal{F}$	0.0805 / 0.105	2.5729 / 0.101 / 0.0975
Multi-scale	0.0404	0.0956
Multi-scale + $\mathcal{F}$	**0.0307**	**0.0625**

(b) L2 error vs. number of scales (plasma MHD).

# Scales	**L2 Rel. Error**
1	0.0805
2	**0.0307**
3	0.0385
4	0.0391

Convergence study on number of scales. We evaluate the performance of StFT on the number of scales for the plasma MHD dataset. As shown in Table 3b, it is observed that increasing the number of scales leads to a reduction in prediction error to a certain point, with the two-layer configuration achieving the lowest error. The inclusion of additional scales results in a slight error increase, indicating a convergence in model performance. Additionally, we conduct an ablation study on the overlapping tokenizer. More details are included in Appendix C.

4.6 Additional Results

Computational and Space Complexity. We compare StFT with other baselines regarding the inference FLOPS per sample, training time and inference time per batch, and peak memory usage during training. We also compare StFT and StFT without the overlapping tokenizer, and we observed this enhancement comes with negligible impact on compute cost, as in Appendix D.

Contribution of Each Scale. To assess the contribution of each StFT block for a specific scale in fitting the training data, we quantify the weight of each block by: $\mathcal{W}_i = \frac{\|\mathbf{y}_i\|_2}{\|\mathbf{y}\|_2}$, where $\mathbf{y}_i$ represents the prediction from the i-th StFT block, and $\mathbf{y}$ denotes the ground truth. We normalize the contributions, and present the contributions in Figure 4. A greater contributing factor from the fine-scale layer in StFT is observed in the Navier-Stokes and shallow-water equations, attributed to the sharper changes and smaller scale structures inherent in the dynamics of higher-order nonlinearities.

5 Conclusion

In this paper, we propose a spatio-temporal Fourier transformer (StFT) for multi-scale and multi-physics long-term dynamics forecasting. Specifically, each StFT block is tailored to address a particular spatial scale, and through a hierarchical composition of multiple StFT blocks spanning different scales, StFT learns the interplay between multiple scales and interactions between multiple physical processes, resulting in stable and accurate long-term dynamics forecasting in an autoregressive manner. Furthermore, we propose and demonstrate the use of a generative residual correction mechanism, which enables meaningful quantification of uncertainties in the predictive model. Despite demonstrating superior forecasting ability in SciML, the model is based on regular grids, which constraints its applicability to irregular grids, and future work will focus on extending the framework to handle irregular geometries. The potential in improving its performance includes model parallelism across the multi-scale StFT blocks and extending StFT-F to end-to-end training.

References

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Mathew Barlow, William J Gutowski, John R Gyakum, Richard W Katz, Young-Kwon Lim, Russ S Schumacher, Michael F Wehner, Laurie Agel, Michael Bosilovich, Allison Collow, et al. North american extreme precipitation events and related large-scale meteorological patterns: a review of statistical methods, dynamics, modeling, and trends. *Climate Dynamics*, 53:6835–6875, 2019.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 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(2):023068, 2020.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Lno: Laplace neural operator for solving differential equations. *arXiv preprint arXiv:2303.10528*, 2023.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6(4):911–917, 1995.

Sai Hung Cheung, Todd A Oliver, Ernesto E Prudencio, Serge Prudhomme, and Robert D Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96 (9):1137–1149, 2011.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis, 2024. *URL https://arxiv. org/abs/2403.03206*, 1, 2024.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6824–6835, 2021.

JP Graves, IT Chapman, S Coda, M Lennholm, M Albergante, and M Jucker. Control of magnetohydrodynamic stability by phase space engineering of energetic ions in tokamak plasmas. *Nature communications*, 3(1):624, 2012.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pp. 12556–12569. PMLR, 2023.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

Sheikh Md Shakeel Hassan, Arthur Feeney, Akash Dhruv, Jihoon Kim, Youngjoon Suh, Jaiyoung Ryu, Yoonjin Won, and Aparna Chandramowlishwaran. Bubbleml: a multi-physics dataset and benchmarks for machine learning. *arXiv preprint arXiv:2307.14623*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Pengzhan Jin, Shuai Meng, and Lu Lu. Mionet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

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.

Nicholas Krämer, Jonathan Schmidt, and Philipp Hennig. Probabilistic numerical method of lines for time-dependent partial differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 625–639. PMLR, 2022.

SE Kruger, Jarrod Leddy, EC Howell, Sandeep Madireddy, C Akcay, T Bechtel Amara, J McClenaghan, LL Lao, David Orozco, SP Smith, et al. Thinking bayesian for plasma physicists. *Physics of Plasmas*, 31 (5), 2024.

Zhijie Li, Wenhui Peng, Zelong Yuan, and Jianchun Wang. Long-term predictions of turbulence by implicit u-net enhanced fourier neural operator. *Physics of Fluids*, 35(7), 2023a.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023b.

Chensen Lin, Zhen Li, Lu Lu, Shengze Cai, Martin Maxey, and George Em Karniadakis. Operator learning for predicting multiscale bubble growth dynamics. *The Journal of Chemical Physics*, 154(10), 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

Xinliang Liu, Bo Xu, and Lei Zhang. Ht-net: Hierarchical transformer based operator learning model for multiscale pdes. 2022b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Da Long, Zhitong Xu, Guang Yang, Akil Narayan, and Shandian Zhe. Arbitrarily-conditioned multi-functional diffusion for multi-physics emulation. *arXiv preprint arXiv:2410.13794*, 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

D Lucor, D Xiu, C-H Su, and GE Karniadakis. Predictability and uncertainty in cfd. *International Journal for Numerical Methods in Fluids*, 43(5):483–505, 2003.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators. *arXiv preprint arXiv:2306.10619*, 2023.

Vinay K Natrajan, Eiichiro Yamaguchi, and Kenneth T Christensen. Statistical and structural similarities between micro-and macroscale wall turbulence. *Microfluidics and Nanofluidics*, 3:89–100, 2007.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

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

Norbert Peters. Multiscale combustion and turbulence. *Proceedings of the Combustion Institute*, 32(1):1–25, 2009.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Michael Prasthofer, Tim De Ryck, and Siddhartha Mishra. Variable-input deep operator networks. *arXiv preprint arXiv:2205.11404*, 2022.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

Sebastian Scher and Gabriele Messori. Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, 144(717):2830–2841, 2018.

Jaemin Seo, SangKyeun Kim, Azarakhsh Jalalvand, Rory Conlin, Andrew Rothstein, Joseph Abbate, Keith Erickson, Josiah Wai, Ricardo Shousha, and Egemen Kolemen. Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature*, 626(8000):746–751, 2024.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Eitan Tadmor. A review of numerical methods for nonlinear partial differential equations. *Bulletin of the American Mathematical Society*, 49(4):507–554, 2012.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Sibo Tian, Minghui Zheng, and Xiao Liang. Transfusion: A practical and effective transformer-based diffusion model for 3d human motion prediction. *IEEE Robotics and Automation Letters*, 2024.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

Stodiek von Goeler, W Stodiek, and N Sauthoff. Studies of internal disruptions and m= 1 oscillations in tokamak discharges with soft—x-ray tecniques. *Physical Review Letters*, 33(20):1201, 1974.

Bin Wang, Brian Hoskins, and Bin Wang. Large-scale atmospheric dynamics. *The Asian Monsoon*, pp. 357–415, 2006.

Qi Wang, Pu Ren, Hao Zhou, Xin-Yang Liu, Zhiwen Deng, Yi Zhang, Ruizhi Chengze, Hongsheng Liu, Zidong Wang, Jian-Xun Wang, Ji-Rong Wen, Hao Sun, and Yang Liu. P^2c^2net: Pde-preserved coarse correction network for efficient prediction of spatiotemporal dynamics. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 68897–68925. Curran Associates, Inc., 2024a. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/7f605d59a0dbde101518b552cb616ddf-Paper-Conference.pdf`.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Improved architectures and training algorithms for deep operator networks. *Journal of Scientific Computing*, 92(2):35, 2022.

Sifan Wang, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J Pappas, and Paris Perdikaris. Bridging operator learning and conditioned neural fields: A unifying perspective. *arXiv preprint arXiv:2405.13998*, 2024b.

Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C Cattin. Diffusion models for medical anomaly detection. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 35–45. Springer, 2022.

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3417–3425, 2022.

A Appendix

B Problem Setup and Datasets

B.1 Plasma magnetohydrodynamic (MHD) equations

We consider magnetohydrodynamic (MHD) equations that characterize the plasma instabilities in fusion tokamaks. The coupled multi-physics system includes the continuity equation solving charge density δn, Poisson's equation solving $\delta\phi$, the Ampere's law to solving $\delta u_{\|}$, the Faraday's law with the assumption $E_{\|} = 0$ to solving $\delta A_{\|}$, and the perpendicular force balance equation to solving $\delta B_{\|}$. The first continuity equation for gyrocenter charge density is expressed as,

$$\begin{aligned}&\frac{\partial \delta n}{\partial t}+\mathbf{B}_0\cdot\nabla\left(\frac{n_0\delta u_{\|}}{B_0}\right)-n_0\mathbf{v}_*\cdot\frac{\nabla B_0}{B_0}+\delta\mathbf{B}_{\perp}\cdot\nabla\left(\frac{n_0 u_{\|0}}{B_0}\right)\\&-\frac{\nabla\times\mathbf{B_0}}{eB_0^2}\cdot\left(\nabla\delta P_{\|}+\frac{\left(\delta P_{\perp}-\delta P_{\|}\right)\nabla B_0}{B_0}\right)\\&+\nabla\cdot\left(\frac{\delta P_{\|}\mathbf{b}_0\nabla\times\mathbf{b}_0\cdot\mathbf{b}_0}{eB_0}\right)-\frac{\mathbf{b}_0\times\nabla\delta B_{\|}}{e}\cdot\nabla\left(\frac{P_0}{B_0^2}\right)\\&-\frac{\nabla\times\mathbf{b}_0\cdot\nabla\delta B_{\|}}{eB_0^2}P_0=0,\end{aligned}\tag{3}$$

where n is the density, B is the magnetic field, $u_{\|}$ is the parallel flow velocity, and P is the pressure. The perturbed quantities are denoted by δ with the equilibrium states including temperature, density, magnetic field and the flux surface from the reconstruction of DIII-D experiments. Here, $\delta n=\delta n_e+q_i\delta n_i/q_e$ stands for the difference of ion and electron density, and $\delta u_{\|}=\delta u_{\|e}+q_i\delta u_{\|i}/q_e$ denotes the difference of ion and electron flow. We have $\mathbf{v}_*=\mathbf{b}_0\times\nabla\left(\delta P_{\|}+\delta P_{\perp}\right)/\left(n_0 m_e\Omega_e\right)$, where m_e is the electron mass, and $\Omega_e=eB_0/m_e$ is the electron cyclotron frequency. The perturbed electron parallel flow $\delta u_{\|}$ can be solved from Ampere's law,

$$\delta u_{\|}=\frac{1}{\mu_0 e n_0}\nabla_{\perp}^2\delta A_{\|},\tag{4}$$

where μ_0 is the permeability of vacuum. $\delta A_{\|}$ is the perturbed vector potential. In the single fluid model, $E_{\|}=0$ is assumed. Then $\delta A_{\|}$ can be solved from

$$\frac{\partial A_{\|}}{\partial t}=\mathbf{b}_0\cdot\nabla\phi,\tag{5}$$

and the electrostatic potential ϕ can be solved from gyrokinetic Poisson's equation (the quasi-neutrality condition)

$$\frac{c^2}{v_A^2}\nabla_{\perp}^2\phi=\frac{e\delta n}{\epsilon_0},\tag{6}$$

where c is the speed of light, v_A is the Alfvén velocity, and ϵ_0 is the dielectric constant of vacuum. The parallel magnetic perturbation δB is given by the perpendicular force balance,

$$\frac{\delta B_{\|}}{B_0}=-\frac{\beta_e}{2}\frac{\delta P_{\perp}}{P_0}=-\frac{\beta_e}{2}\frac{\partial P_0}{\partial\psi_0}\frac{\delta\psi}{P_0}.\tag{7}$$

The perturbed pressure in the fluid limit can be calculated by

$$\begin{aligned}\delta P_{\perp}&=\frac{\partial P_0}{\partial\psi_0}\delta\psi-2\frac{\delta B_{\|}}{B_0}P_0,\\\delta P_{\|}&=\frac{\partial P_0}{\partial\psi_0}\delta\psi-\frac{\delta B_{\|}}{B_0}P_0.\end{aligned}\tag{8}$$

In these equations, ψ_0 and $\delta\psi$ is the equilibrium and perturbed magnetic flux, and the evolution of $\delta\psi$ is solved from

$$\frac{\partial\delta\psi}{\partial t}=-\frac{\partial\phi}{\partial\alpha_0},\tag{9}$$

where α_0 is from the Clebsch representation of $\mathbf{B}$ field, and $\mathbf{B}_0=\nabla\psi_0\times\nabla\alpha_0$. We run a linear gyrokinetic simulation with a $100\times250\times24$ mesh in radial, poloidal and parallel directions. The time step is set to $\Delta t=0.005R_0/C_s=1.483\times10^{-8}$s. We keep both $n=0,1$ modes, generate a trajectory of $128,000$ time steps, and collect the data every 100 snapshots. We focus on emulating the dynamics of electrostatic potential $\delta\phi$, parallel vector potential $\delta A_{\|}$, electron number density δn_e, ion number density δn_i, and electron velocity δu_e in their trajectories.

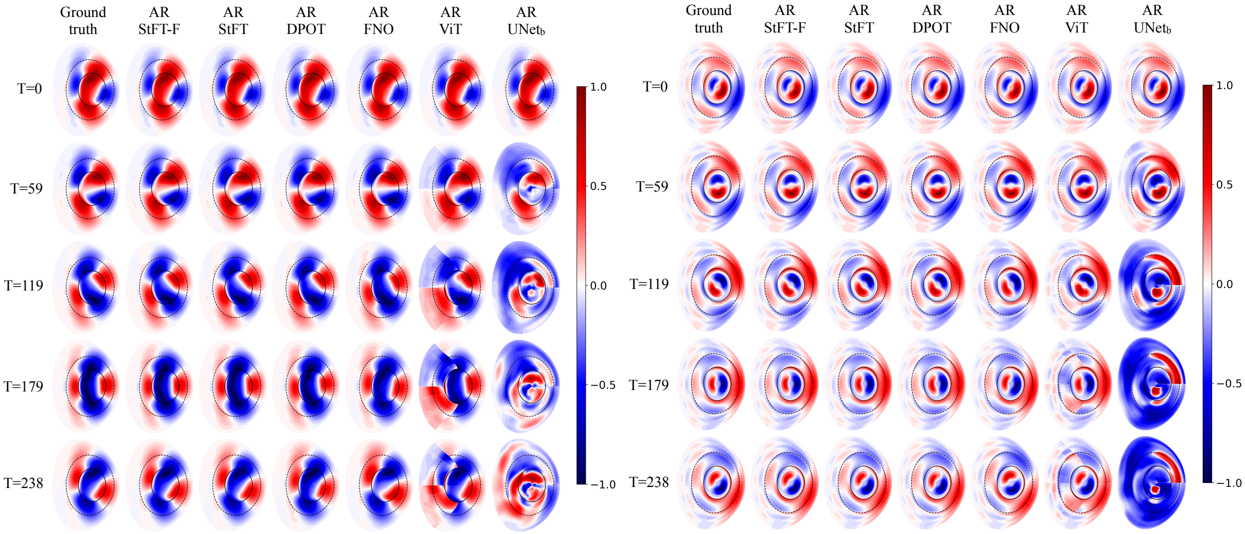

Figure 6: Temporal evolution of normalized perturbed parallel vector potential $\delta A_\parallel$ and perturbed electron density δn_e contours predicted by different models: StFT-F, StFT, FNO, ViT and U-Net. Significant phase differences between the predictions of the models appear after $T = 59$, where StFT and StFT-F perform stable across the forecasting time horizon.

B.2 2D incompressible Navier-Stokes equations

We consider the 2D incompressible Navier-Stokes (NS) equation on a rectangular domain $(x, y) \in [0, 1]^2$,

$$\begin{aligned}
&\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} + \frac{1}{\mathrm{Re}}\nabla^2 u + f(x, y),\\
&\frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} + \frac{1}{\mathrm{Re}}\nabla^2 v + f(x, y),\\
&\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,
\end{aligned} \tag{10}$$

where u and v represent the velocity components in the x and y directions, and p represents the pressure. $f(x, y)$ is the source term, and we set it to $e^{-100((x-0.5)^2+(y-0.5)^2)}$. The Reynolds number is set to 1000. We run a finite difference solver to compute the solutions on a 50×50 spatial grid, with the temporal domain discretized into a total of 101 timestamps over $T \in [0, 20]$. We generated a total of 100 trajectories by sampling the four boundary conditions uniformly from $(0.1, 0.6)$.

B.3 Spherical shallow-water equations

We consider the viscous shallow-water equations modeling the dynamics of large-scale atmospheric flows:

$$\begin{aligned}
&\frac{D\mathbf{V}}{Dt} = -f\mathbf{k} \times \mathbf{V} - g\nabla h + \nu\nabla^2\mathbf{V},\\
&\frac{Dh}{Dt} = -h\nabla \cdot \mathbf{V} + \nu\nabla^2 h, \quad x \in \Omega,\ t \in [0, 1],
\end{aligned} \tag{11}$$

where $\mathbf{V}$ is the velocity vector tangential to the spherical surface, $\mathbf{k}$ is the unit vector normal to the surface, h is the thickness of the fluid layer, $f = 2\Xi \sin\phi$ is the Coriolis parameter (Ξ being the Earth's angular velocity), g is the gravitational acceleration, and ν is the diffusion coefficient. The equations are defined over a spherical domain $\Omega = (\lambda, \phi)$, with longitude λ and latitude ϕ.

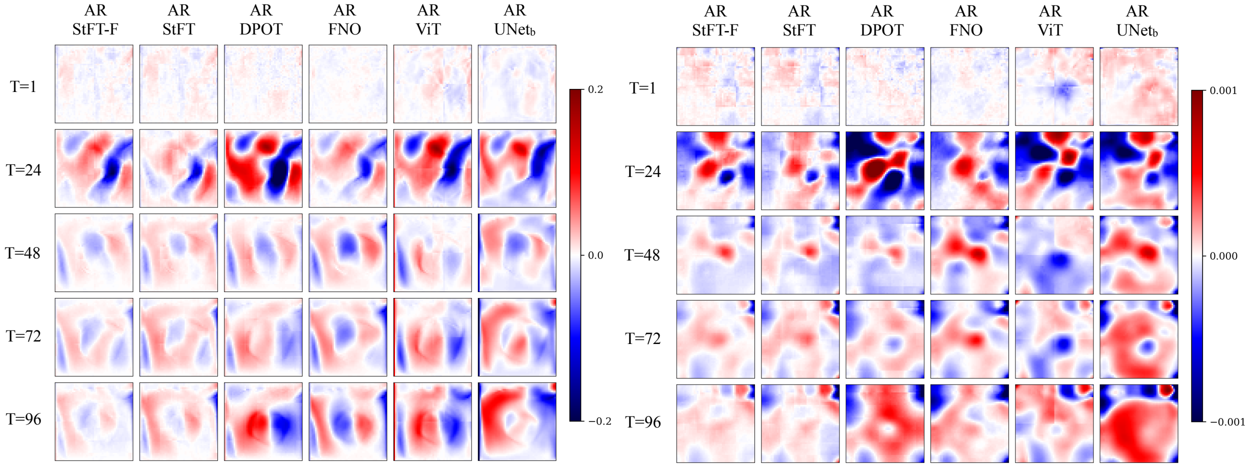

Figure 7: 2D incompressible Navier-Stokes equation: pointwise error of the predicted evolution of velocity component u and pressure p contours across different models: StFT-F, StFT, FNO, ViT and U-Net. For long-term predictions, StFT and StFT-F demonstrate lower residuals compared to other models.

As an initial condition, a zonal flow typical of a mid-latitude tropospheric jet is defined for the velocity component u as a function of latitude ϕ:

$$u(\phi, t=0) = \begin{cases} 0, & \phi \le \phi_0, \\ \frac{u_{\max}}{n} \exp\left[\frac{1}{(\phi-\phi_0)(\phi-\phi_1)}\right], & \phi_0 < \phi < \phi_1, \\ 0, & \phi \ge \phi_1, \end{cases}$$

where $u_{\max}$ is the maximum zonal velocity, ϕ_0 and ϕ_1 represent the southern and northern boundaries of the jet in radians, and $n = \exp[-4/(\phi_1 - \phi_0)^2]$ normalizes $u_{\max}$ at the midpoint of the jet. To induce barotropic instability, a localized Gaussian perturbation is added to the height field, expressed as:

$$h'(\lambda, \phi, t=0) = \hat{h}\cos(\phi)\exp\left[-\left(\frac{\lambda}{\alpha}\right)^2\right]\exp\left[-\left(\frac{\phi_2 - \phi}{\beta}\right)^2\right],$$

where $-\pi < \lambda < \pi$, and parameters $\hat{h}, \phi_2, \alpha$, and β control the shape and location of the perturbation. The parameters α and β are sampled from uniform distributions $\alpha \sim U[0.1, 0.5]$ and $\beta \sim U[0.03, 0.2]$. We ran the solver from Dedalus Burns et al. (2020) on a 256×256 spherical grid, and the temporal dimension is discretized into 72 timestamps. We have a total of 200 trajectories by sampling α and β.

C Ablation Study

C.1 The Hierarchical Structure and The Frequency Path

To evaluate the effectiveness of the hierarchical structure and the frequency path in StFT, we conduct an ablation study of our model. First, we only keep one layer of StFT while removing the hierarchical structure. Second, we keep the hierarchical structure, and remove the frequency path in each hierarchical layer.

Table 4 shows the L_2 relative errors averaged over all the variables. Note that l_1 is the coarsest level, and l_3 is the finest level. $\mathcal{F}$ stands for the frequency path in StFT blocks. We find that multi-scale $+\mathcal{F}$" outperforms both ablation cases. These results demonstrate the effectiveness of the hierarchical composition of StFT blocks and the frequency path in StFT block. With the hierarchical composition, for Plasma MHD, the error drops to 0.0307 from 0.0805, and for shallow-water equations (SWE), the error drops to 0.0625 from 0.0975. We observe that the fine-level layer setting in the SWE achieves the best performance among the single layer

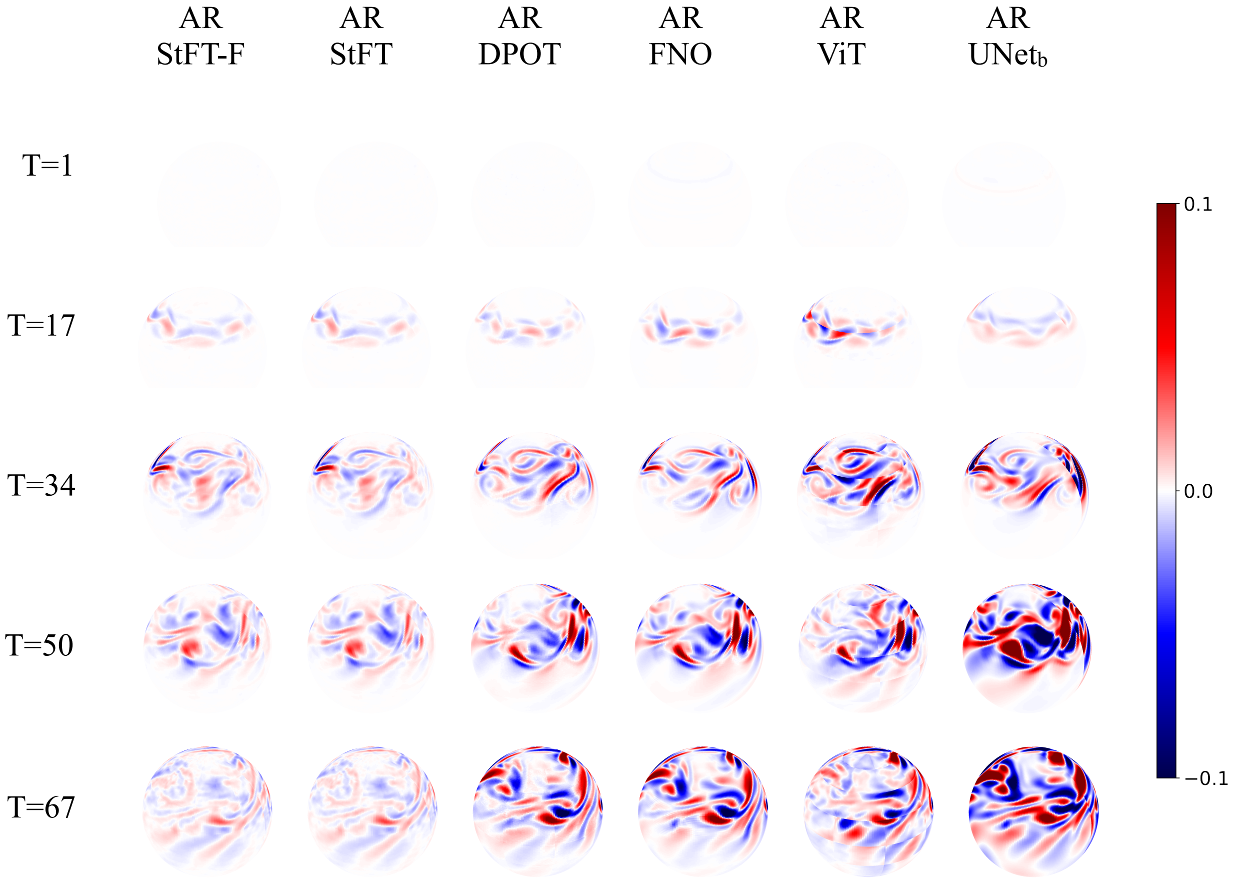

Figure 8: Spherical shallow-water equations: pointwise error of the temporal evolution of velocity field predicted by all the autoregressive models: StFT-F, StFT, FNO, ViT and U-Net. The prediction error exhibits a temporal growth trend, with our model StFT and StFT-F consistently demonstrate lower residuals over the forecasting time horizon.

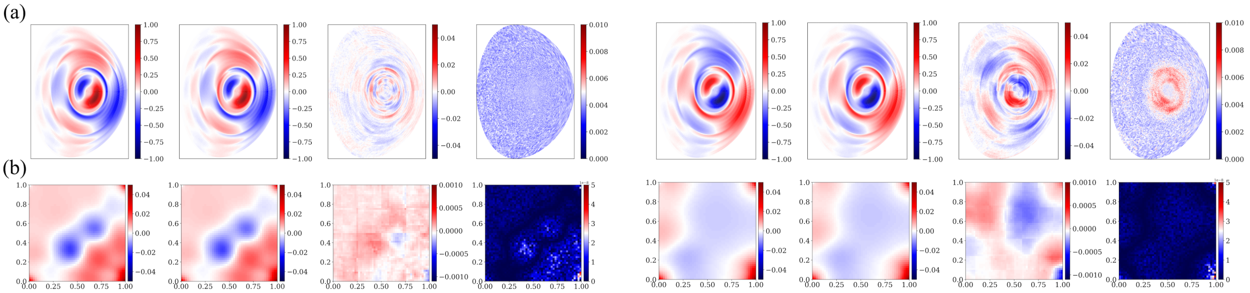

Figure 9: Additional evaluation of forecasting: ground truth, StFT-F prediction, residual, and uncertainty over time - shown for initial (left) and final (right) states. Variables include: (a) δn_e in plasma MHD, (b) p in Navier-Stokes.

Table 4: Ablation study results. We run models with combinations of StFT blocks and the frequency path. l_1 is the coarsest level, and l_3 is the finest level. F stands for the frequency path in StFT blocks.

Dataset	Setting(s)	AR-StFT
Plasma MHD	$l_1 + \mathcal{F}$	0.0805
	$l_2 + \mathcal{F}$	0.105
	$l_1 + l_2$	0.0404
	$l_1 + l_2 + \mathcal{F}$	**0.0307**
Shallow-Water	$l_1 + \mathcal{F}$	2.5729
	$l_2 + \mathcal{F}$	0.101
	$l_3 + \mathcal{F}$	0.0975
	$l_1 + l_2 + l_3$	0.0956
	$l_1 + l_2 + l_3 + \mathcal{F}$	**0.0625**

results, and the multi-layer settings further decrease the prediction error. The frequency path in StFT also plays a crucial role, where in Plasma MHD, the error drops to 0.0307 from 0.0404, and in the SWE dataset, the error drops to 0.0625 from 0.0956.

C.2 The Overlapping Tokenizer

We conduct an ablation study on the overlapping tokenizer design, evaluating both predictive performance and computational cost, as reported in Tables 5 and 6.We evaluate the computational cost including the inference FLOPS per sample, training/inference time per batch and peak memory usage. We used a fixed batch size of 20 for all models. The results are summarized in the following two tables. StFT-O stands for StFT with the overlapping tokenizer and StFT-NO for the opposite. All experiments were performed on an NVIDIA A100 GPU. Incorporating the overlapping tokenizer leads to substantial improvements in accuracy, in plasma MHD and shallow-water datasets, with errors reductions of 68% and 10% respectively. Notably, this enhancement comes with negligible impact on computational complexity, inference time and peak memory usage.

Table 5: Effect of the overlapping tokenizer on the prediction error of plasma MHD and shallow-water datasets.

Method	Plasma MHD	Shallow-Water
With overlapping tokenizer	**0.0307**	**0.0625**
Without overlapping tokenizer	0.0986	0.0700

Table 6: Computational complexity comparison of StFT with and without overlapping tokenizer.

Dataset	Method	GFLOPs	Training / Iter (s)	Inference / Iter (s)	Peak Memory (GB)
MHD	StFT-O	0.704	0.181	0.0455	9.49
MHD	StFT-NO	0.704	0.157	0.0454	9.49
SWE	StFT-O	4.30	0.189	0.0319	12.2
SWE	StFT-NO	4.30	0.189	0.0319	12.2

C.3 Convergence on the Number of Scales

We evaluate the performance of StFT on the number of scales for the plasma MHD dataset. Specifically, we run StFT with one scale (patch size of 128), three scales (patch sizes of 128, 64, and 48), and four scales (patch sizes of 128, 64, 48, and 32). As shown in Table 7, it is observed that increasing the number of scales leads to a reduction in prediction error to a certain point, with the two-layer configuration achieving the lowest error (0.0307). Beyond this, the inclusion of additional scales results in a slight increase in error, indicating a convergence in model performance, and further increasing the number of scales may not provide performance gain.

Table 7: Effect of the number of scales on L2 relative error of the overall prediction on the plasma MHD.

Number of Scales	L2 Relative Error
1	0.0805
2	0.0307
3	0.0385
4	0.0391

D Computational and Space Complexity

We compare StFT with other baselines regarding the inference FLOPS per sample, training time per batch, inference time per batch, and peak memory usage (during training). In plasma MHD and shallow-water datasets, compared to FNO, StFT is 20x smaller, and 8x smaller in computation (FLOPS) respectively. While achieving the highest prediction accuracy, StFT has the same order of magnitude of FLOPs as UNet and ViT in plasma MHD. In the shallow-water equation (SWE), however UNet is much more computationally expensive - 11x more than StFT – while its relative error is 3x larger than StFT. StFT is also efficient in both training and inference time. For plasma MHD, StFT achieves a 30% reduction in training time and 50% reduction in inference time compared to FNO. For SWE, StFT is 40% faster in training and 60% faster than FNO in inference. Although StFT incorporates dual paths operating in the frequency domain and the spatio-temporal domain, its peak memory usage remains comparable to that of FNO in the plasma MHD, and is reduced by 56% compared to FNO in SWE, and 26% less than UNet in SWE. Figure 10 compares the inference time and FLOPs versus mean L2 relative error for all models.

Table 8: Comparison of methods on plasma MHD and shallow-water datasets regarding the computational and space complexity.

Dataset	Method	GFLOPs	Training / Iter (s)	Inference / Iter (s)	Peak Memory (GB)
MHD	StFT	0.704	0.181	0.0455	9.49
MHD	DPOT	1.10	0.0676	0.0120	3.20
MHD	FNO	13.8	0.262	0.0900	10.1
MHD	UNET	0.462	0.0263	0.00230	1.38
MHD	ViT	0.338	0.0171	0.00237	1.65
SWE	StFT	4.30	0.189	0.0319	12.2
SWE	DPOT	1.61	0.0427	0.0133	8.43
SWE	FNO	34.6	0.325	0.0886	30.2
SWE	UNET	48.5	0.107	0.0257	16.5
SWE	ViT	0.90	0.0130	0.00347	7.45

E Signficance of UQ on StFT-F

For the Plasma MHD dataset, we measure the average coverage for the first 20 autoregressively predicted snapshots (each row represents one physical variable), as shown in Table 9. In the last row, we report the average coverage across all physical variables. For the 90% confidence interval, average coverage is very close to the ideal value of 90%, demonstrating that StFT-F is well-calibrated around the 90% confidence interval. For the 95% confidence interval, the average coverage is under-confident by 2.5%, suggesting the intervals may be slightly narrow. For the shallow-water dataset, we provide the results in Table 10, where we measure

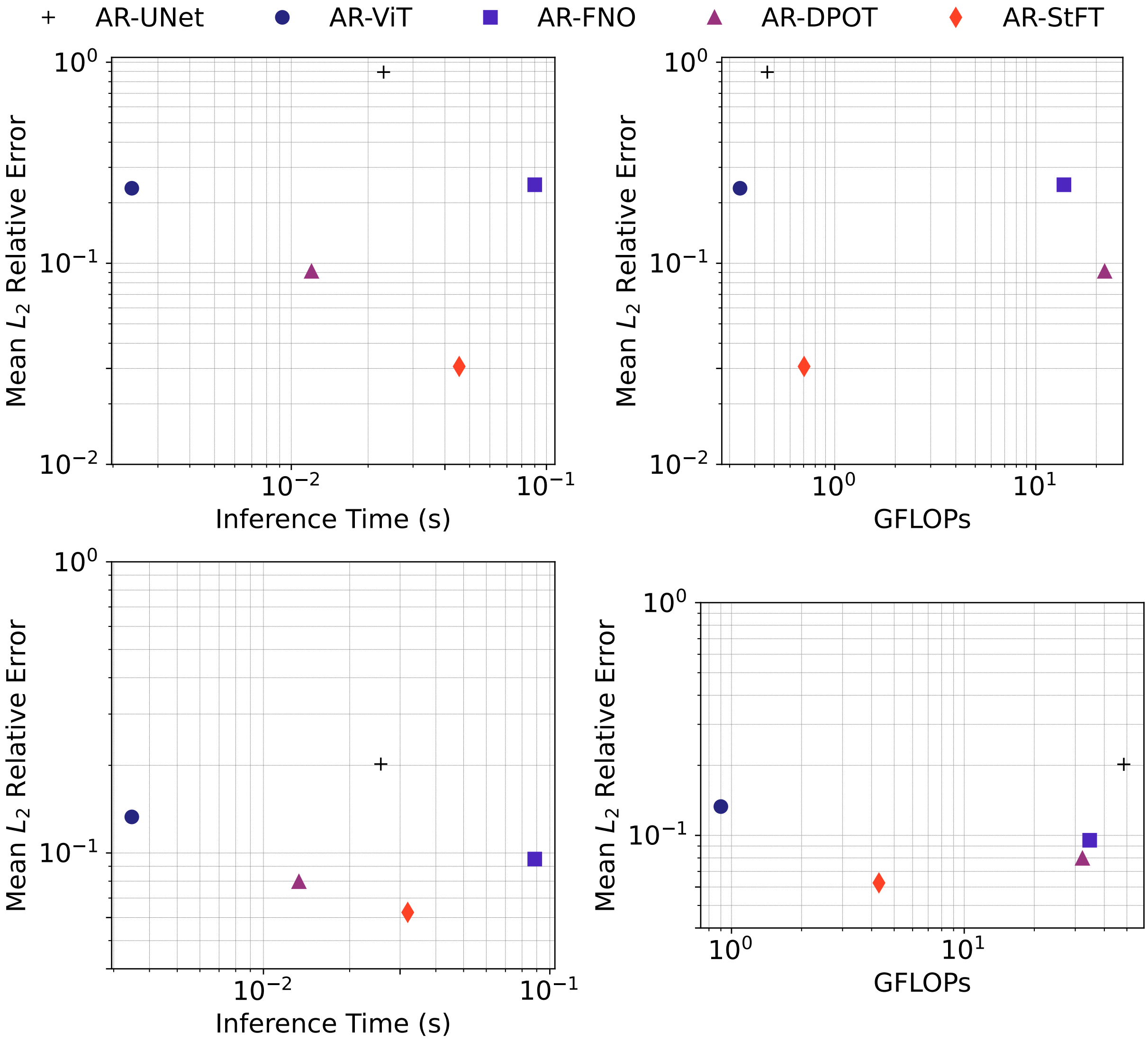

Figure 10: Comparison of models in terms of mean L2 relative error versus inference time (left) and FLOPs (right) for plasma MHD (top) and shallow-water (bottom).

the average coverage for the first 10 predicted snapshots. For the 90% confidence interval, average coverage is also very close to the ideal value of 90%, which shows that StFT-F is also well-calibrated for the shallow-water dataset.

Table 9: Confidence interval coverage on the plasma MHD dataset.

Variable	**CI: 90%**	**CI: 95%**
$\delta\phi$	0.906	0.937
$\delta A_{\|}$	0.977	0.988
$\delta B_{\|}$	0.900	0.930
δn_e	0.933	0.962
δn_i	0.867	0.910
δu_e	0.781	0.823
Average Coverage	**89.4%**	**92.5%**

Table 10: Confidence interval coverage on the shallow-water dataset.

Variable	**CI: 90%**	**CI: 95%**
V	0.895	0.980

Table 11: Training/validation/test data splits and the training budget for all models measured on an A100 GPU.

Dataset	**Total**	**Split (Train / Val / Test)**	**Training Budget**
Plasma MHD	1 traj. (1,221 snapshots)	927 / 50 / 244	24h
Navier-Stokes	100 traj.(101 snaps/traj)	80 / 10 / 10	24h
Shallow-water	200 traj. (72 snaps/traj)	160 / 20 / 20	48h

F Experimental Details

Training/validation/test data sets. For the plasma MHD data, we split the trajectory of 1221 snapshots into a training set (the first 927 snapshots), a validation set (the middle 50 snapshots), and a test set (the last 244 snapshots). For the Navier-Stokes dataset, we have a total of 100 trajectories (101 snapshots for each trajectory), and split them into 80 trajectories for training, 10 for validation, and 10 for testing. For the shallow-water dataset, we have a total of 200 trajectories (72 snapshots for each trajectory), and split them into 160 trajectories for training, 20 for validation, and 20 for testing. To ensure a fair comparison, we impose a fixed training budget across all models. Specifically, a 48-hour limit mesaured on one A100 GPU was set for the shallow-water equation dataset, while a 24-hour limit is applied to both the plasma MHD and Navier-Stokes equation datasets. Table 11 summarizes the datasets and the training budget for all models.

Generative residual correction block. We follow a two-step training protocol in training StFT-F: first, we train StFT thoroughly with the training budget, and then we train the generative residual correction block for another 200 epochs. We employ a rectified flow to learn distributions of the residuals given the prediction of StFT and the history snapshots. We implement a similar structure to the Diffusion Transformer (DiT) as the backbone model (Peebles & Xie, 2023). In each DiT block, we apply adaptive layer normalization before a self-attention layer and an MLP layer. We use adaLN-Zero for time conditioning. For the history snapshots $\tilde{u}_t$ and the prediction $\mathcal{F}_{\theta_d}(\tilde{u}_t)$ of StFT, these conditions are incorporated as extra input tokens.

Hyperparameters. For StFT on the plasma dynamics, 3D FFT is used to encode the spatio-temporal inputs in the frequency path. We use the patch size of 128 for the the first StFT block and and 64 for the second StFT block. The overlapping size is set to 1. The hidden dimension is set to 128. The depth for each StFT block is set to 6. We keep the lowest 8 frequencies for each spatial dimension. For the rectified flow block, the depth is set to 8, and the hidden dimension is set to 128. For the Navier-Stokes equation, StFT uses a patch size of 25 for the coarse block (the first block), 13 for the middle block, and 8 for the last block. The overlapping size is set to 0, and the frequency path is not used. For each block in the hierarchical structure, the depth is set to 8, and we use a hidden dimension of 512. In the rectified flow block, we use a depth of 4 and set the hidden dimension to 128. For the shallow-water equation, three levels of StFT blocks are employed, and their patch sizes are set to 128, 64, and 32 respectively. For each block, the depth is set to 6, and the hidden dimension is set to 512. We use the 2D FFT to encode the spatio-temporal inputs, and the lowest 8 frequencies are kept for each spatial dimension. The overlapping size is set to 1. For the rectified flow model, we use a depth of 8 and a hidden dimension of 128.

Table 12: Hyperparameter search range for each method.

Method	Hyperparameter Search
DPOT	Hidden dimension: $[256, 512]$ Patch size: $[8, 16, 32]$ Depth: 6 Heads: 4
AR-FNO	Modes: $[16, 20, 24]$ Layers: $[4, 5]$ Hidden dimension: 256
AR-ViT	Hidden dimension: $[256, 512]$ Patch size: $[16, 32, 64]$
AR-UNet	Bottleneck hidden dimension: $[64, 128, 256, 512]$

Baselines. For all the baselines, we run all models with the same training time budget, as detailed in Table 11. For DPOT, we vary the hidden dimension from $[256, 512]$, and the patch size from $[8, 16, 32]$. We set the depth as 6, and the number of heads as 4, which is the default setting from the author's implementation. For AR-FNO, the number of modes are selected through a search in $[16, 20, 24]$, the number of layers are searched in $[4, 5]$, and the hidden dimension is set to 256. For AR-ViT, we vary the hidden dimension from $[256, 512]$, and the patch size from $[16, 32, 64]$. For AR-UNet, the hidden dimension of bottleneck embeddings are searched in $[64, 128, 256, 512]$. Table 12 summarizes the hyperparameter choices for each method.

G More Visualization Results

Figure 6 illustrates the ground truth and predicted temporal evolution of normalized perturbed parallel vector potential $\delta A_{\parallel}$ and perturbed electron density δn_e in plasam MHD using AR-StFT, AR-StFT-F, AR-DPOT, AR-FNO, AR-ViT and AR-UNet methods. StFT and StFT-F perform stable across the forecasting time horizon. Figure 7 and 8 show the pointwise error of all the models compared to the ground truth data in the Navier-Stokes and shallow-water equations, where StFT and StFT-F demonstrate lower residuals compared to other baseline models.

In addition, we compare StFT with StFT-F for the autoregressive prediction across different timestamps for velocity u in Navier-Stokes equation, shown in Figure 11. A slightly larger error is observed between $t = 0.4$ and 0.6, which can be attributed to the training objective of StFT. As StFT is optimized using a pointwise loss function, it is encouraged to produce predictions that closely match the most likely outcome. In contrast, StFT-F is designed to learn the full distribution of the target, potentially introducing higher variance in its predictions. This distributional modeling, while beneficial for uncertainty quantification, may result in marginally increased errors as it captures characteristics beyond the mean behavior.

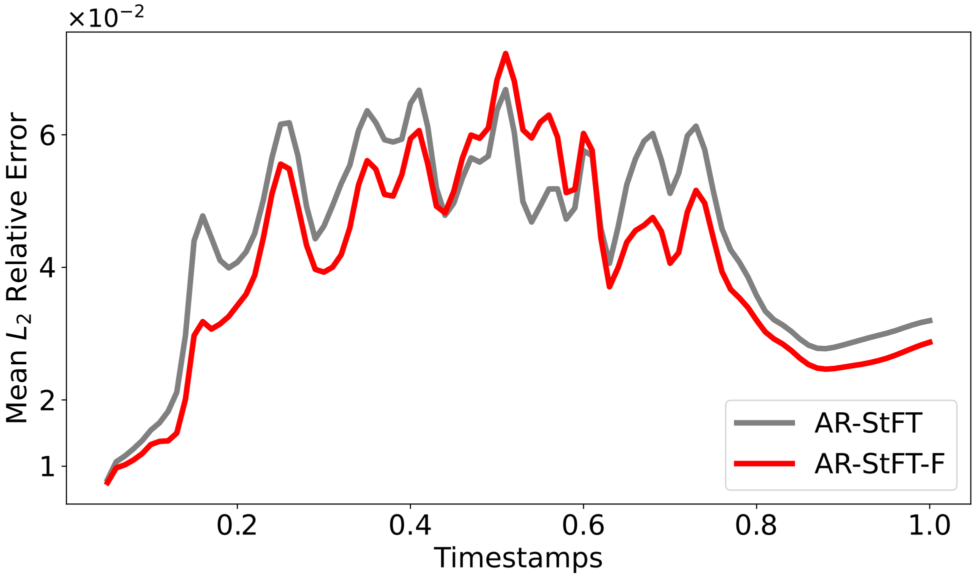

Figure 11: Results of comparing StFT and StFT-F for the autoregressive prediction in L_2 relative error across different timestamps for velocity u in Navier-Stokes equation. The shaded region represents the standard deviation distribution of the relative error of StFT-F. However, the uncertainty is negligible that it is not visually discernible. StFT-F demonstrates better performance in the latter stages of the forecasting time horizon compared to StFT.