MATEY: multiscale adaptive transformer models for spatiotemporal physical systems

Anonymous authors
Paper under double-blind review

Abstract

Accurate representation of the multiscale features in spatiotemporal physical systems using vision transformer (ViT) architectures requires extremely long, computationally prohibitive token sequences. To address this issue, we propose two adaptive tokenization schemes that dynamically adjust patch sizes based on local features: one ensures convergent behavior to uniform patch refinement, while the other offers better computational efficiency. Moreover, we present a set of spatiotemporal attention schemes, where the temporal or axial spatial dimensions are decoupled, and evaluate their computational and data efficiencies. We assess the performance of the proposed multiscale adaptive model, MATEY, in a sequence of experiments. The results show that adaptive tokenization schemes achieve improved accuracy without significantly increasing the length of the token sequence. Compared to a full spatiotemporal attention scheme or a scheme that decouples only the temporal dimension, we find that fully decoupled axial attention is less efficient and expressive, requiring more training time and model weights to achieve the same accuracy. Finally, we demonstrate in two fine-tuning tasks featuring different physics that models pretrained on PDEBench data outperform the ones trained from scratch, especially in the low data regime with frozen attention.

1 Introduction

Developing foundation models for physical systems is vital for energy generation, earth sciences, and power and propulsion systems. These models offer faster solutions than physics-based simulations and can generalize better across multiple systems than single-purpose AI approaches. However, their application to physical systems, often characterized by multiple sub-processes at different scales, is still in the early stages. For instance, fluid flowing around a cylinder creates a von Kármán vortex street, a highly dynamic flow with rapidly evolving vortices. Accurate solutions of such multiscale systems require a very high resolution representation to capture the most complex features across space and time. However, for scientific machine learning as for modeling and simulation, using very high resolutions to achieve accurate solutions incurs significant computational cost. This is particularly true for developing foundation models using vision transformer (ViT)-based architectures, as using the standard self-attention mechanism for extremely long spatiotemporal sequences can become prohibitively computationally expensive.

Efficient representation of multiscale features in high-resolution inputs has been an active research topic in computer vision. Three broad approaches can be characterized. First, multiscale models like Swin Transformer (Liu et al., 2021) and MViTv2 (Li et al., 2022) introduce multiple stages with decreasing resolution and increasing feature dimension for efficient hierarchical representations. Second, computational techniques have been developed that facilitate training on long sequences (e.g., sequence parallelism across GPUs (Jacobs et al., 2023)) or reduce the effective sequence length in the attention kernel (e.g., decomposing attention along axial directions (Ho et al., 2019)). Third, the actual sequence length can be directly shortened by pruning and merging tokens ((Haurum et al., 2023; Meng et al., 2022; Yin et al., 2022; Bolya & Hoffman, 2023)), though this strategy may lead to critical information loss (Liu et al., 2024).

These techniques have recently been adopted in scientific machine learning (sciML) for physical systems. For example, the atmosphere foundation model Aurora (Bodnar et al., 2024) uses Swin Transformer, while axial attention is applied by MPP (McCabe et al., 2023). Despite the progress, computational constraints remain a bottleneck, as existing approaches do not yet handle high-fidelity solutions of applications such as computational fluid dynamics, in which input sequences can easily exceed billions of tokens. More efficient algorithms are needed to enable the development of foundation models for multiscale multiphysics systems.

In this work, we develop a multiscale adaptive foundation model, MATEY (see Figure 1), that provides two key algorithmic contributions to address the challenges posed by spatiotemporal physical systems. First, inspired by the adaptive mesh refinement (AMR) technique, we introduce an adaptive tokenization method that dynamically adjusts patch sizes across the system based on local features, which provides as much as a 2× reduction in compute for similar or higher accuracy. Second, we present a set of spatiotemporal attention schemes based on the axial attention (Ho et al., 2019) that differ in their decomposition of long spatiotemporal sequences and identify the cost in time-to-accuracy for axial attention. Finally, we assess the fine-tuning performance of models pretrained on PDEBench (Takamoto et al., 2022) in two highly out-of-distribution settings, colliding thermals and magnetohydrodynamics (MHD), that include additional physical variables not included in pretraining and observe the pretrained models outperforming random initialized models.

2 Related work

Scientific foundation models Several research directions have been explored for building foundation models for physical systems, including multiple physics pretraining (McCabe et al., 2023) with PDEBench data, input augmentation with PDE system configurations (Hang et al., 2024), robust pretraining schemes (Hao et al., 2024), fine-tuning effectiveness investigations (Subramanian et al., 2024), and data-efficient multiscale ViT architectures (Herde et al., 2024). While these work made remarkable progress, they do not directly address the issue of token sequence length, which becomes a computation bottleneck when applying ViTs to high dimension or high resolution data.

Multiscale ViTs While most multiscale ViTs achieve hierarchical representations via multi-stage attention blocks at different resolutions (e.g., MViTv2 (Li et al., 2022) and Swin Transformer (Liu et al., 2021)), there are a few focusing on tokenization schemes, such as(Yin et al., 2022; Fan et al., 2024; Zhang et al., 2024; Havtorn et al., 2023). Among these, the single-stage MSViT with dynamic mixed-scale tokenization (Havtorn et al., 2023), which leverages a learnable gating neural network for selecting the token refinement, is most related to our work. This approach requires a tailored gate loss function and an adaptive trimming scheme to handle the high overhead cost, which in return hurts gate training accuracy. In contrast, the tokenization scheme in MATEY adaptively adjusts the patch sizes directly based on local feature scales, which is simpler and more direct.

Axial attentions The quadratic scaling nature of attention makes it computationally prohibitive for extremely long token sequences from multidimensional systems. To address this challenge, (Ho et al., 2019) proposed the axial attention, which decomposes the full attention into a sequence of attention operations along each axis. It reduces the attention cost from $\mathcal{O}(N^{2d})$ to $\mathcal{O}(N^{d+1})$, for a given d-dimensional system with N^d tokens. ViViT (Arnab et al., 2021) factorized the spatiotemporal attention into spatial- and temporal-dimensions for video classification. (McCabe et al., 2023) applied the axial attention in the Axial ViT (AViT) for spatiotemporal solutions of physical systems. While these spatiotemporal attention schemes can reduce the sequence length and hence the attention cost, their impact on accuracy in physical systems is unclear.

3 MATEY, explained

We propose multiscale adaptive foundation models, MATEY, to predict two-dimensional spatiotemporal solutions of multiple physical systems. The architecture of MATEY is illustrated in Figure 1. Given a sequence of T past solutions of some physical system leading up to time t, MATEY predicts the solution at a future time $t + t_{\text{lead}}$ by learning from sequences of solutions for multiple physical systems. Specifically, MATEY

learns a model $\mathbf{f_w}$ such that $\mathbf{u}_{t+t_{\text{lead}}} \approx \mathbf{f_w}(\mathbf{u}_{t-T+1}, \ldots, \mathbf{u}_t; t_{\text{lead}})$ by training parameters $\mathbf{w}$ to minimize the loss of the prediction from the solution sequence $\mathbf{U} = [\mathbf{u}_{t-T+1}, \ldots, \mathbf{u}_t]$ against the future solution with a lead time $\mathbf{u}_{t+t_{\text{lead}}}$. In the following paragraphs, we give detailed descriptions for each component in MATEY.

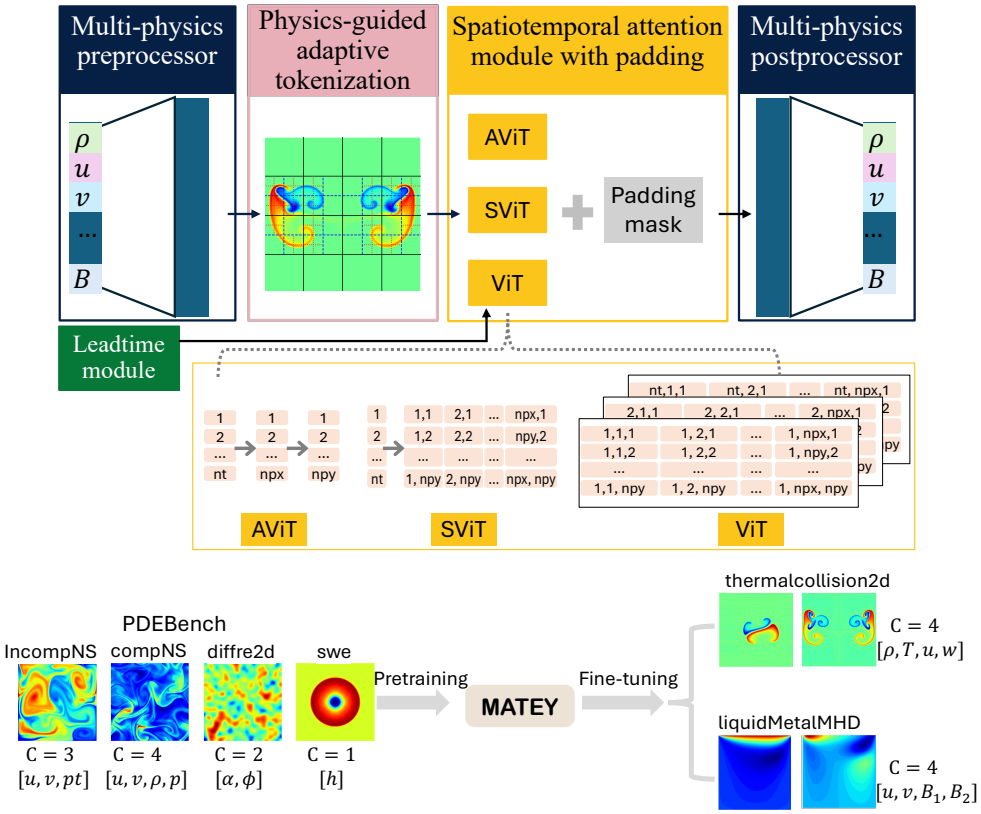

Figure 1: MATEY: multiscale adaptive foundation models for spatiotemporal physical systems.

Multi-physics preprocessor, postprocessor, and training To accommodate multiple physical systems with different sets of variables at different spatial resolutions, we adopt the multi-physics preprocessor and postprocessor used in MPP (McCabe et al., 2023). For system k with C_k variables, the preprocessor first encodes solutions $\mathbf{u}_t(x, y) \in \mathbb{R}^{C_k}$ to a latent space $\mathbb{R}^{C_{\text{uni}}}$, where $C_{\text{uni}} \gg C_k$ is shared among all systems. Specifically, letting H and W denote the resolution in the x and y directions, respectively, the preprocessor encodes the solution $\mathbf{U}_k \in \mathbb{R}^{T \times H \times W \times C_k}$ of system k into the unified latent representation $\mathbf{U} \in \mathbb{R}^{T \times H \times W \times C_{\text{uni}}}$. $\mathbf{U}$ is then tokenized into sequences $\boldsymbol{Z}^0 \in \mathbb{R}^{nt \times npx \times npy \times C_{\text{emb}}}$ in the tokenization module, which consists of convolutional blocks. Here $nt = T/p_t$, $npx = H/p_x$, and $npy = W/p_y$ are the number of patches in each dimension with prescribed patch size $[p_t, p_x, p_y]$. After passing through L attention blocks, the input token sequence $\boldsymbol{Z}^0$ leads to the attention output $\boldsymbol{Z}^L \in \mathbb{R}^{nt \times npx \times npy \times C_{\text{emb}}}$. The last temporal snapshot of $\boldsymbol{Z}^L$ is then decoded in the postprocessor into the prediction $\mathbf{u}_{\text{pred}} \in \mathbb{R}^{H \times W \times C_k}$. In this work, the preprocessor is a linear map, the tokenization module is implemented as a convolutional neural network (CNN), and the final decoding postprocessor uses 2D transposed convolutional blocks. To train the model from solutions with different resolutions, we follow the approach in MPP by performing system-based sampling in the training process and fusing information from samples across different systems via multi-GPU training with PyTorch Distributed Data Parallelism (DDP) and gradient accumulation.

Attention mechanisms — AViT, SViT, and ViT The standard ViT attention mechanism takes into account the attention across the entire set of spatiotemporal dimensions, which results in a high attention cost when extremely long spatiotemporal token sequences (e.g., from high-resolution spatiotemporal data) are considered. To address this issue, various factorized attention mechanisms have been proposed, such as

AViT (Ho et al., 2019; McCabe et al., 2023) and a spatio-temporal decoupled attention (Arnab et al., 2021), referred to as SViT here. These attention mechanisms mainly consist of the same multihead self-attention (MHSA) and feed forward multi-layer perceptron (MLP) but differ in their attention block architecture. When L attention blocks are cascaded, the standard attention block in ViT is given as

$$\begin{aligned}
&\widehat{\boldsymbol{Z}}^0 = \boldsymbol{Z}^0 + \boldsymbol{E}_{\text{pos}}, \quad \boldsymbol{Z}^0 = [\boldsymbol{z}_1^0, \boldsymbol{z}_2^0, \ldots, \boldsymbol{z}_N^0], \\
&\boldsymbol{Z}^1 = \text{MLP}(\widetilde{\boldsymbol{Z}}^1) + \widetilde{\boldsymbol{Z}}^1, \qquad \widetilde{\boldsymbol{Z}}^1 = \text{MHSA}(\widehat{\boldsymbol{Z}}^0) + \widehat{\boldsymbol{Z}}^0 + \text{MLP}(t_{\text{lead}}), \\
&\boldsymbol{Z}^\ell = \text{MLP}(\widetilde{\boldsymbol{Z}}^\ell) + \widetilde{\boldsymbol{Z}}^\ell, \qquad \widetilde{\boldsymbol{Z}}^\ell = \text{MHSA}(\boldsymbol{Z}^{\ell-1}) + \boldsymbol{Z}^{\ell-1}, \qquad \ell = 2, \ldots, L
\end{aligned} \tag{1}$$

where $[\boldsymbol{z}_1^0, \ldots, \boldsymbol{z}_N^0]$ denotes the full spatiotemporal token sequence of length N with each token $\boldsymbol{z}_i^0 \in \mathbb{R}^{C_{\text{emb}}}$, $\boldsymbol{E}_{\text{pos}}$ is a positional embedding term, and each MHSA and MLP is followed by an `InstanceNorm1d` module. In ViT, the token sequence includes all spatiotemporal patches, meaning $N = nt \cdot npx \cdot npy$, resulting in an overwhelming cost of $\mathcal{O}((nt \cdot npx \cdot npy)^2)$ operations for attention. In contrast, SViT decouples the attention into $npx \cdot npy$ time-attention blocks and nt space-attention blocks cascaded sequentially, as in "$\text{MHSA}_{\text{time}} \rightarrow \text{MHSA}_{\text{space}} \rightarrow \text{MLP}$",

$$\begin{aligned}
&\text{Time sequences:} \quad \boldsymbol{Z}_i^{\ell-1} = \left[\boldsymbol{z}_{(i-1)\cdot nt+1}^{\ell-1}, \boldsymbol{z}_{(i-1)\cdot nt+2}^{\ell-1}, \ldots, \boldsymbol{z}_{(i-1)\cdot nt+nt}^{\ell-1}\right], i = 1, \ldots, npx \cdot npy \\
&\text{Attention in time:} \quad \boldsymbol{Z}_i^{\ell-\frac{1}{2}} = \text{MHSA}_{\text{time}}\left(\boldsymbol{Z}_i^{\ell-1}\right) + \boldsymbol{Z}_i^{\ell-1}, \quad i = 1, \ldots, npx \cdot npy \\
&\text{Space sequences:} \quad \check{\boldsymbol{Z}}_t^{\ell-\frac{1}{2}} = \left[\boldsymbol{z}_t^{\ell-\frac{1}{2}}, \boldsymbol{z}_{t+nt}^{\ell-\frac{1}{2}}, \ldots, \boldsymbol{z}_{t+nt\cdot(npx\cdot npy-1)}^{\ell-\frac{1}{2}}\right], \quad t = 1, \ldots, nt, \\
&\text{Attention in space:} \quad \widetilde{\boldsymbol{Z}}_t^{\ell} = \text{MHSA}_{\text{space}}\left(\check{\boldsymbol{Z}}_t^{\ell-\frac{1}{2}}\right) + \check{\boldsymbol{Z}}_t^{\ell-\frac{1}{2}}, \quad t = 1, \ldots, nt, \\
&\text{Feed forward ML:} \quad \boldsymbol{Z}^{\ell} = \text{MLP}\left(\widetilde{\boldsymbol{Z}}^{\ell}\right) + \widetilde{\boldsymbol{Z}}^{\ell}, \quad \ell = 1, \ldots, L,
\end{aligned} \tag{2}$$

which reduces the MHSA cost to $npx \cdot npy \cdot \mathcal{O}(nt^2) + nt \cdot \mathcal{O}((npx \cdot npy)^2)$. The position embedding and the lead time MLP are omitted in (2) for simplicity. AViT further decomposes the space-attention in SViT into two axial directions following the same approach, which leads to a cost of $npx \cdot npy \cdot \mathcal{O}(nt^2) + nt \cdot npy \cdot \mathcal{O}(npx^2) + nt \cdot npx \cdot \mathcal{O}(npy^2)$. The decomposition in both AViT and SViT neglects some spatiotemporal correlations, and thus gives shorter token sequence length for each attention block, at the cost of introducing additional attention blocks. These extra attention blocks moderately increase the model size, as shown in Table 1. Note that within the same size category considered in Table 1, AViT and ViT are larger than ViT due to the additional MHSA, while AViT and ViT have similar sizes because AViT reuses the same attention blocks for different spatial directions. In MATEY, we implement the three attention mechanisms – AViT, SViT, and ViT – and evaluate their performance on test problems to study how the lost spatiotemporal correlations affect the quality of the solution and to assess the impact of decoupled attentions with additional attention blocks on the learning efficiency for multi-physics foundation models.

Adaptive tokenization Smaller patch sizes are preferred for better representation accuracy, as ViTs can capture long-range correlations between patches well but lack inductive biases within patches. However, features in physical systems often cross multiple length scales and exhibit strong spatiotemporal inhomogeneities. Consequently, constant patch sizes that are small enough to provide good accuracy in the necessary regions of such systems result in impractically long token sequence lengths over the entire domain. To address this issue, we propose an adaptive ViT that dynamically adjusts the tokenization patch sizes according to local physical features. To maximize expressiveness, we start with coarse patching and identify the most complex patches in each sample based on a simple metric, such as the variance of local features. The identified patches are further refined to the sub-token-scale (STS) to improve representation accuracy. Adaptive patch size leads to patches of varying length across samples, which are handled with padding masks. Patch position and patch area bias are represented following the embedding method in (Bodnar et al., 2024).

For a given solution field $\boldsymbol{u}_t \in \mathbb{R}^{H \times W \times C}$, tokenization at a constant patch size $[p_x, p_y]$ is achieved through a CNN block and leads to a patch grid of size $(npx, npy) = (H/p_x, W/p_y)$. For adaptive tokenization, we apply varying patch sizes in space based on local complexity represented by the patch variance. For a solution $\boldsymbol{u}_t \in \mathbb{R}^{H \times W \times C}$ and an initial coarse patch size $[p_{x_1}, p_{y_1}]$, a variance tensor $\boldsymbol{v}_t \in \mathbb{R}^{npx_1 \times npy_1}$

($npx_1 = H/p_{x_1}$ and $npy_1 = W/p_{y_1}$) is calculated from solutions inside each patch of the reshaped solution $\widetilde{\boldsymbol{u}}_t \in \mathbb{R}^{npx_1 \times npy_1 \times p_{x_1} \times p_{y_1} \times C}$ as

$$\boldsymbol{v}_t(i,j) = \frac{1}{C \cdot p_{x_1} \cdot p_{y_1}} \sum_{c=1}^{C} \sum_{k=1}^{p_{x_1}} \sum_{l=1}^{p_{y_1}} \left(\widetilde{\boldsymbol{u}}_t(i,j,k,l,c) - \frac{1}{p_{x_1} \cdot p_{y_1}} \sum_{k=1}^{p_{x_1}} \sum_{l=1}^{p_{y_1}} \widetilde{\boldsymbol{u}}_t(i,j,k,l,c) \right)^2. \tag{3}$$

Patches with variance values greater than a prescribed threshold are then selected for further refinement at a smaller patch size. Specifically, let STS-IDs denote the index set of patches to be refined, then

$$\text{STS-IDs} := \{(i,j) | \boldsymbol{v}_t(i,j) > \gamma_{\text{sts}} \cdot \boldsymbol{v}_{t,\max}\}, \quad N_{\text{sts}} := |\text{STS-IDs}|, \tag{4}$$

where $\gamma_{\text{sts}} \in [0,1]$ is a user-specified hyperparameter, $\boldsymbol{v}_{t,\max}$ is the maximal variance among all patches, and N_{sts} is the number of patches to be refined. The selected patches are refined to patches of a smaller size $[p_{x_{\text{sts}}}, p_{y_{\text{sts}}}]$, referred to as "STS tokens" in this work, where $\boldsymbol{Z}^0_{\text{sts},i} = \left[\boldsymbol{z}^0_{\text{sts},1}, \boldsymbol{z}^0_{\text{sts},2}, \ldots, \boldsymbol{z}^0_{\text{sts},p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}}\right]_i$ ($i = 1, \ldots, N_{\text{sts}}$). The STS tokens can be combined with the coarse tokens in two ways, as shown in Figure 2. In the first approach, referred to as "Adap_Mul" (for adaptive multi-resolution tokenization), we consider the coarse and STS tokens as separate sequences, passing through the attention blocks serially. In the second approach, referred to as "Adap_Mix" (for adaptive mixed-resolution tokenization), we replace the selected coarse patches with the sequence of STS tokens directly appended to the end of the sequence.

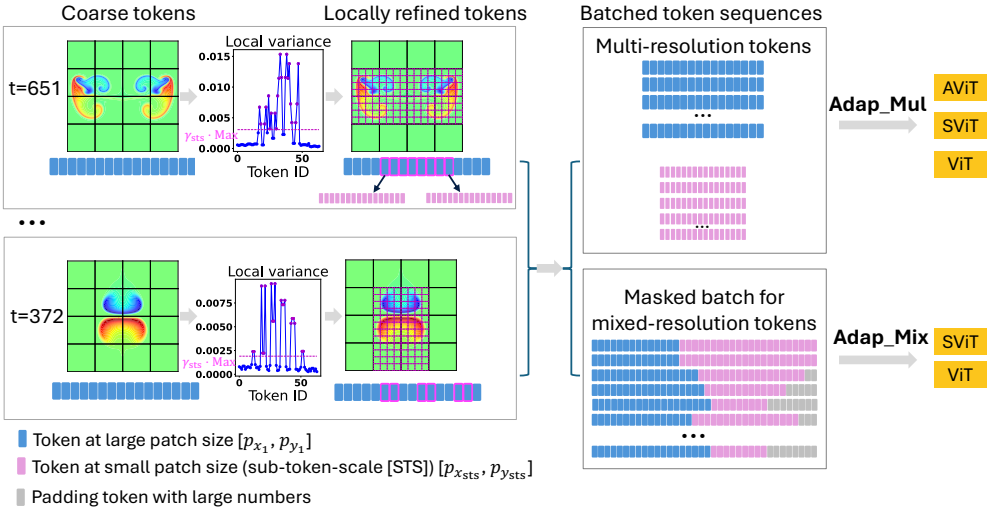

Figure 2: Adaptive tokenization that dynamically adjusts patch sizes based on local features. There are three essential parameters: $[p_{x_1}, p_{y_1}]$, $[p_{x_{\text{sts}}}, p_{y_{\text{sts}}}]$ and γ_{sts}. The parameter $[p_{x_1}, p_{y_1}]$ denotes the initial coarse patch size, $[p_{x_{\text{sts}}}, p_{y_{\text{sts}}}]$ represents the refined patch size, and $\gamma_{\text{sts}} \in [0,1]$ determines which patches to refine. We select patches with local variances greater than γ_{sts} times the maximum variance across all patches (see Equation (4)).

After spatiotemporal attention, the decoding of adaptive patch sequences into solution fields within the multi-physics postprocessor is performed using transposed convolutional blocks, tailored to each corresponding scale. For Adap_Mul, the patches at different resolutions/sizes are deconvoluted separately and then summed to the final output, $\widehat{\boldsymbol{u}}_t$. Specifically, for a coarse attention output $\boldsymbol{Z}^L_{\text{coarse}} = \left[\boldsymbol{z}^L_1, \boldsymbol{z}^L_2, \ldots, \boldsymbol{z}^L_{npx_1 \times npy_1}\right]$ and STS attention outputs $\boldsymbol{Z}^L_{\text{sts},i} = \left[\boldsymbol{z}^L_{\text{sts},1}, \boldsymbol{z}^L_{\text{sts},2}, \ldots, \boldsymbol{z}^L_{p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}}\right]_i$ ($i = 1, \ldots, N_{\text{sts}}$), "Adap_Mul" performs the following operations:

$$\begin{aligned}
\text{Reconstruction from coarse patches:} \quad & \widehat{\boldsymbol{u}}_t = \text{ConvTranspose2d}_1(\boldsymbol{Z}^L_{\text{coarse}}), \\
\text{Reconstruction from STS patches:} \quad & \widehat{\boldsymbol{u}}_{t,\text{sts},i} = \text{ConvTranspose2d}_2(\boldsymbol{Z}^L_{\text{sts},i}) \\
& \widehat{\boldsymbol{u}}_{t,\text{sts}} = [\widehat{\boldsymbol{u}}_{t,\text{sts},1}, \ldots, \widehat{\boldsymbol{u}}_{t,\text{sts},N_{\text{sts}}}] \\
\text{Fusion of multi-resolution solutions:} \quad & \widehat{\boldsymbol{u}}_t[\text{STS-IDs}] = \widehat{\boldsymbol{u}}_t[\text{STS-IDs}] + \widehat{\boldsymbol{u}}_{t,\text{sts}}.
\end{aligned} \tag{5}$$

On the other hand, the Adap_Mix approach fuses the coarse and STS patch sequences into the full sequences at the coarse and fine STS scales, respectively, reconstructs the solutions via transposed convolutions at corresponding resolutions separately, and then merges them to achieve multi-resolution solutions. This approach guarantees consistency with the coarse patch solution when $\gamma_{\text{sts}} = 1.0$ (no refinement) and the fine patch solution when $\gamma_{\text{sts}} = 0.0$ (refining all patches). Let $\boldsymbol{Z}'^{L}_{\text{coarse}} = [\boldsymbol{z}^L_1, \boldsymbol{z}^L_2, \ldots, \boldsymbol{z}^L_{npx_1 \times npy_1 - N_{\text{sts}}}] \in \mathbb{R}^{(npx_1 \times npy_1 - N_{\text{sts}}) \times C_{\text{emb}}}$ denote the coarse portion of the mixed-resolution attention output, and let $\boldsymbol{Z}^L_{\text{sts},i} = [\boldsymbol{z}^L_1, \boldsymbol{z}^L_2, \ldots, \boldsymbol{z}^L_{p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}}]_i$ $(i = 1, \ldots, N_{\text{sts}})$ denote the STS portion. Adap_Mix performs the following operations:

1. Reconstruct the full coarse patches $\boldsymbol{Z}^L_{\text{coarse}} \in \mathbb{R}^{npx_1 \times npy_1 \times C_{\text{emb}}}$ via

$$\begin{aligned} \boldsymbol{Z}^L_{\text{coarse}}[\text{Kep-IDs}] &= \boldsymbol{Z}'^{L}_{\text{coarse}}, \\ \boldsymbol{Z}^L_{\text{coarse}}[\text{STS-IDs}] &= [\text{Mean}(\boldsymbol{Z}^L_{\text{sts},1}), \text{Mean}(\boldsymbol{Z}^L_{\text{sts},2}), \ldots, \text{Mean}(\boldsymbol{Z}^L_{\text{sts},N_{\text{sts}}})], \end{aligned} \tag{6}$$

   where Kep-IDs is the complementary indexing tensor to STS-IDs, representing all coarse patches kept in the sequence.

2. Reconstruct the full fine patches $\boldsymbol{Z}^L_{\text{fine}} \in \mathbb{R}^{H/p_{x_{\text{sts}}} \times W/p_{y_{\text{sts}}} \times C_{\text{emb}}}$ via

$$\begin{aligned} \boldsymbol{Z}'^{L}_{\text{fine}}[\text{STS-IDs}, :, :] &= [\boldsymbol{Z}^L_{\text{sts},1}, \boldsymbol{Z}^L_{\text{sts},2}, \ldots, \boldsymbol{Z}^L_{\text{sts},N_{\text{sts}}}], \\ \boldsymbol{Z}'^{L}_{\text{fine}}[\text{Kep-IDs}, :, :] &= \text{repeat}\left(\boldsymbol{Z}'^{L}_{\text{coarse}}, p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}\right), \\ \boldsymbol{Z}^L_{\text{fine}} &= \text{reshape}\left(\boldsymbol{Z}'^{L}_{\text{fine}}\right). \end{aligned} \tag{7}$$

   where $\boldsymbol{Z}'^{L}_{\text{fine}} \in \mathbb{R}^{(npx_1 \times npy_1) \times (p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}) \times C_{\text{emb}}}$ is an intermediate supporting tensor.

3. Reconstruct solution fields $\widehat{\boldsymbol{u}}_{t,\text{coarse}} \in \mathbb{R}^{H \times W \times C}$ and $\widehat{\boldsymbol{u}}_{t,\text{fine}} \in \mathbb{R}^{H \times W \times C}$ from coarse patches and fine patches, respectively:

$$\widehat{\boldsymbol{u}}_{t,\text{coarse}} = \text{ConvTranspose2d}_1(\boldsymbol{Z}^L_{\text{coarse}}), \quad \widehat{\boldsymbol{u}}_{t,\text{fine}} = \text{ConvTranspose2d}_2(\boldsymbol{Z}^L_{\text{fine}}). \tag{8}$$

4. Fusion of solutions from step 3 to get the multi-resolution solution fields $\widehat{\boldsymbol{u}}_t \in \mathbb{R}^{H \times W \times C}$:

$$\widehat{\boldsymbol{u}}_t[\text{Kep-IDs}] = \widehat{\boldsymbol{u}}_{t,\text{coarse}}[\text{Kep-IDs}], \quad \widehat{\boldsymbol{u}}_t[\text{STS-IDs}] = \widehat{\boldsymbol{u}}_{t,\text{fine}}[\text{STS-IDs}]. \tag{9}$$

Among the two adaptive approaches, Adap_Mul is simpler to implement, requiring minimal code modifications, supports the AViT attention mechanism, and does not increase the maximum sequence lengths. In contrast, Adap_Mix produces relatively longer sequences and lacks AViT support but has the potential significant benefit of better capturing cross-scale correlations than the decoupled Adap_Mul. Furthermore, by varying γ_{sts} from 1.0 to 0.0, Adap_Mix guarantees a smooth transition from the coarse patch solution at $[p_{x_1}, p_{y_1}]$ to the fine patch solution at $[p_{x_{\text{sts}}}, p_{y_{\text{sts}}}]$ (see Figure 5).

Pretraining and fine-tuning We pretrain the models on PDEBench data, which includes five basic 2D systems: incompressible flows, compressible flows, turbulent flows, reaction-diffusion systems, and shallow water equations. We consider two fine-tuning cases: 1) colliding thermals between a cold and a warm bubbles from MiniWeather simulations (Norman, 2020) and 2) lid-driven cavity MHD flows (Fambri et al., 2023). As discussed in detail in Appendix A.1, these fine-tuning datasets were selected to be meaningfully out-of-distribution, not only in flow regime but also in including thermal and electromagnetic components that are not represented at all in the pretraining data. Training was performed on the Frontier and Perlmutter supercomputers at the Oak Ridge Leadership Computing Facility (OLCF) and National Energy Research Scientific Computing Center (NERSC), respectively.

4 Experiments

We design three experiments to evaluate 1) the performance of three spatiotemporal attention schemes, AViT, SViT, and ViT, 2) the impact of adaptive tokenization, and 3) the effectiveness of pretrained models on two fine-tuning tasks that feature physics different from the pretraining data. In these experiments, we set $p_t = 1$ and $C_{\text{uni}} = C_{\text{emb}}/4$, and employ square patches (i.e., $p_x = p_y$, $p_{x_1} = p_{y_1}$, and $p_{x_{\text{sts}}} = p_{y_{\text{sts}}}$) by default.

4.1 Spatiotemporal attention schemes

We evaluate AViT, SViT, and ViT for three model sizes: Tiny (Ti), Small (S), and Base (B) with 3, 6, 12 heads and hidden dimension $C_{\text{emb}} = 192$, 384, and 768, respectively (Touvron et al., 2022), as shown in Table 1, on the colliding thermals dataset. In the same size category, AViT and SViT are about 30% larger than ViT due to the additional attention block. More details about the experiment are presented in Appendix A.2.

Table 1: Number of model parameters in AViT, SViT, and ViT for three model sizes, Tiny, Small, and Base, detailed in Section 4.1. ViT results in about 30% fewer model parameters than AViT and SViT because the latter two require additional attention blocks.

	Tiny	Small	Base
AViT	7.5M	29.9M	119.3M
SViT	7.6M	30.0M	119.3M
ViT	5.8M	22.8M	90.9M

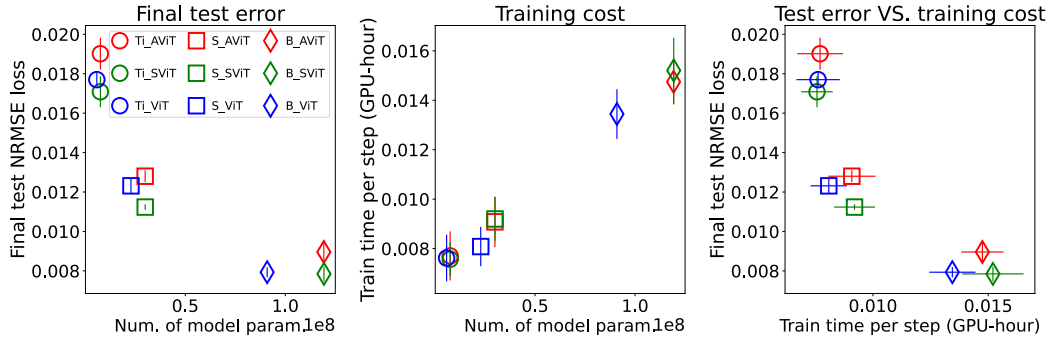

Figure 3: Learning efficiency of AViT, SViT, and ViT at three model sizes regarding final predictive error and training time cost: SViT and ViT are observed to be more expressive and computationally efficient than AViT in the experiment, as they require fewer model parameters and less training time to achieve the same test accuracy.

Figure 3 compares the final test error, defined as the normalized root-mean-square error (NRMSE), and the training time, represented as GPU hour per step, for the nine models. For the same size category, SViT (green) achieves the lowest error, followed by ViT (blue), and then AViT (red). In terms of training time, SViT takes longer than AViT, while ViT is the least expensive. ViT processes longer token sequences and hence is expected to have a higher single-unit attention cost, whereas AViT and SViT have multiple attention units with shorter token sequence lengths. The results reported in Figure 3 show that the ViT has the lowest cost, which implies that the number of attention blocks plays a more important role than the token sequence length in terms of training cost in this example. This observation is due to the fact that the spatiotemporal token sequence length ($16 \times 8 \times 8$) in this example is relatively short. We expect ViT to become more expensive than AViT and SViT when more refined or higher dimensional solutions are considered, in which longer token sequences are required. In general, we find that SViTs and ViTs are more

expressive and computationally efficient than AViTs, in that they achieve the lower predictive errors with fewer model parameters and less training time.

4.2 Adaptive tokenization

We start the evaluation of our adaptive tokenization methods in a single collision trajectory between two thermal bubbles. Figure 4 compares the temperature contours of the true solution at $t = 590$ with the predicted solutions from Ti-SViT models at constant patch sizes, ps=16×16 and ps=32×32, and adaptive tokenization (Adap_Mul with $p_{x_1} = p_{y_1} = 32$, $p_{x_{\text{sts}}} = p_{y_{\text{sts}}} = 16$, and $\gamma_{\text{sts}} = 0.2$). The predicted solution from ps=32×32 exhibits abrupt changes with clear edges for the local structures inside the patches, while the finer resolution model at ps=16×16 captures smoother, finer structures but requires many more patches. In contrast, our adaptive tokenization methods capture smooth, fine structures comparable to ps=16×16 while requiring much shorter sequences.

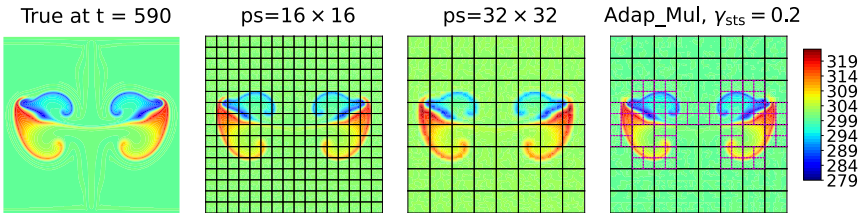

Figure 4: Predicted temperature contours at $t = 590$ from Ti-SViT models with constant patch sizes ps=16×16 and ps=32×32 and adaptive tokenization (Adap_Mul with $p_{x_1} = p_{y_1} = 32$, $p_{x_{\text{sts}}} = p_{y_{\text{sts}}} = 16$, and $\gamma_{\text{sts}} = 0.2$). Adap_Mul predicts smoother, finer local structures that are overlooked in ps=32×32, similar to the more expensive ps=16×16.

Adap_Mix in ViT and SViT Adap_Mix with $(p_{x_1}, p_{x_{\text{sts}}}, \gamma_{\text{sts}})$ is designed to ensure convergence in γ_{sts} values. When $\gamma_{\text{sts}} \to 1$, no refinement is conducted and the output is converged to ps=$p_{x_1} \times p_{x_1}$. Conversely, when $\gamma_{\text{sts}} \to 0$, all patches are refined and the output is converged to ps=$p_{x_{\text{sts}}} \times p_{x_{\text{sts}}}$. To examine such convergence behavior, we conduct a set of runs with varying $(p_{x_1}, p_{x_{\text{sts}}}, \gamma_{\text{sts}})$ values, together with runs at constant patch sizes, ps=32×32, ps=16×16, and ps=8×8. Figure 5 shows the final NRMSE test loss versus the average sequence length of patches per time step, $L_{\text{avg,mix}}$. For a given trajectory of spatiotemporal solutions with T steps, the average sequence length is defined as

$$L_{\text{avg,mix}} = \frac{1}{T}\sum_{t=1}^{T} L_t = \frac{1}{T}\sum_{t=1}^{T}\left[(npx_1 \cdot npy_1 - N_{\text{sts},t}) + N_{\text{sts},t} \cdot \left(\frac{p_{x_1}}{p_{x_{\text{sts}}}} \cdot \frac{p_{y_1}}{p_{y_{\text{sts}}}}\right)\right], \tag{10}$$

where L_t represents the length of the mixed patch sequence and $N_{\text{sts},t}$ is the number of patches selected based on Equation (4) at time t. Clearly, the predictive error of Adap_Mix evolves from the corresponding coarse patch results of ps=$p_{x_1} \times p_{x_1}$ to ps=$p_{x_{\text{sts}}} \times p_{x_{\text{sts}}}$ when γ_{sts} varying from 1.0 to 0.0, in two settings $(p_{x_1} = 32, p_{x_{\text{sts}}} = 16)$ and $(p_{x_1} = 16, p_{x_{\text{sts}}} = 8)$ and for both ViT and SViT. More interestingly, Adap_Mix at some γ_{sts} values even achieves a lower predictive error than the fine patch case ps=$p_{x_{\text{sts}}} \times p_{x_{\text{sts}}}$ with a much shorter sequence length (e.g., with $2\times$ reduction), clearly indicating the advantages of the adaptive tokenization approach.

Adap_Mul in ViT, SViT, and AViT In contrast to Adap_Mix, which combines the coarse and locally refined patches into a hybrid sequence that is fed into attention blocks together, Adap_Mul treats the two, i.e., $\boldsymbol{Z}^L_{\text{coarse}} = \left[\boldsymbol{z}^L_1, \boldsymbol{z}^L_2, \ldots, \boldsymbol{z}^L_{npx_1 \times npy_1}\right]$ and $\boldsymbol{Z}^L_{\text{sts},i} = \left[\boldsymbol{z}^L_{\text{sts},1}, \boldsymbol{z}^L_{\text{sts},2}, \ldots, \boldsymbol{z}^L_{p_{x_1}/p_{x_{\text{sts}}} \times p_{y_1}/p_{y_{\text{sts}}}}\right]_i$ $(i = 1, \ldots, N_{\text{sts}})$, separately. Adap_Mul maintains this separation for both attention and solution reconstruction and views the reconstructed solutions from the refined patches as a local STS correction. The computing cost scales either linearly for MLP or quadratically for attention with sequence length in various model components.

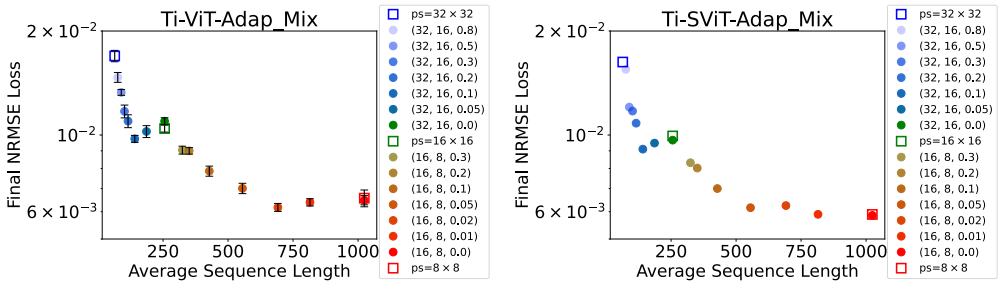

Figure 5: Final NRMSE loss for Tiny ViT (left) and SViT (right) with adaptive tokenization — **Adap_Mix** with hyperparamters (p_{x_1}, $p_{x_{\text{sts}}}$, γ_{sts})— and constant patch sizes against average sequence length, $L_{\text{avg,mix}}$ (Equation (10)). Error bars, representing standard deviations from 3 runs, are shown for ViT. Adap_Mix with γ_{sts} varying from 1.0 to 0.0 shows a clear convergent transition from the coarse constant patch size ps=$p_{x_1} \times p_{y_1}$ to the fine constant patch size ps=$p_{x_{\text{sts}}} \times p_{y_{\text{sts}}}$. More interestingly, Adap_Mix is shown to achieve lower prediction errors than the more expensive ps=$p_{x_{\text{sts}}} \times p_{x_{\text{sts}}}$ cases despite requiring only half of the average sequence length.

To represent the cost, we define the linear and quadratic indices for ViT and SViT as in

$$
\begin{aligned}
L_{\text{lin}} &= \frac{1}{T}\sum_{t=1}^{T} L_t = \frac{1}{T}\sum_{t=1}^{T}\left[npx_1 \cdot npy_1 + N_{\text{sts},t} \cdot \left(\frac{p_{x_1}}{p_{x_{\text{sts}}}} \cdot \frac{p_{y_1}}{p_{y_{\text{sts}}}}\right)\right], \\
L_{\text{quad}} &= \frac{1}{T}\sum_{t=1}^{T} L_{\text{quad},t} = \frac{1}{T}\sum_{t=1}^{T}\left[(npx_1 \cdot npy_1)^2 + N_{\text{sts},t} \cdot \left(\frac{p_{x_1}}{p_{x_{\text{sts}}}} \cdot \frac{p_{y_1}}{p_{y_{\text{sts}}}}\right)^2\right].
\end{aligned}
\tag{11}
$$

For AViT, the index $L_{\text{quad},t}$ needs to be adjusted as

$$
L_{\text{quad},t} = \left(npx_1^2 \cdot npy_1 + npx_1 \cdot npy_1^2\right) + N_{\text{sts},t} \cdot \left[\left(\frac{p_{x_1}}{p_{x_{\text{sts}}}}\right)^2 \cdot \frac{p_{y_1}}{p_{y_{\text{sts}}}} + \frac{p_{x_1}}{p_{x_{\text{sts}}}} \cdot \left(\frac{p_{y_1}}{p_{y_{\text{sts}}}}\right)^2\right], \tag{12}
$$

Figure 6 shows the final NRMSE test loss against the two cost estimation indices L_{lin} (left) and L_{quad} (right) of Adap_Mul in ViT (top) and SViT (bottom) at varying values of (p_{x_1}, $p_{x_{\text{sts}}}$, γ_{sts}). Clearly, as γ_{sts} decreases for ViT and SViT, predictive errors are significantly reduced. Simultaneously, the linear cost increases progressively from coarse to refined and beyond, while the increase in quadratic cost remains negligible. Moreover, the accuracy does not converge to the refined case when $\gamma_{\text{sts}} = 0$, despite the higher linear cost. This indicates that Adap_Mul is better suited for scenarios involving attention on long sequences. Figure 7 presents similar results for Adap_Mul in AViT. Compared with ViT and SViT, the accuracy improvement in AViT is less pronounced in our experiments with $(p_{x_1}, p_{x_{\text{sts}}}) = (32, 16)$ and $(16, 8)$ (top), possibly due to the extremely short sequence lengths of 2 in AViT. Notably, significant accuracy gains are observed when $p_{x_{\text{sts}}}$ is reduced from 16 to 8 for $p_{x_1} = 32$ (bottom).

Comparing the two approaches for adaptive tokenization, we find that **Adap_Mix** provides better predictive accuracy, likely due to considering cross-scale correlations, and guarantees convergence toward the solution with uniformly refined tokens. In contrast, **Adap_Mul** is dramatically more cost effective for attention operations with quadratic complexity and easier to implement than Adap_Mix. AViT does not interact well with adaptive tokenization approaches when the STS sequence length $p_{x_1}/p_{x_{\text{sts}}}$ is too short.

4.3 Effectiveness of pretraining in colliding thermals and MHD fine-tuning tasks

We examine the transferrability of pretrained models to fine-tuning systems with distinct physics and different set of variables, as in Table A1. Specifically, we aim to address three broad questions: (1) Is pretraining effective when the downstream tasks have a distinct set of physical variables? (2) How does limited fine-tuning of non-attention blocks compare to full fine-tuning? and (3) How does the amount of fine-tuning data affect convergence? To address these three questions, we design a sets of experiments, starting from

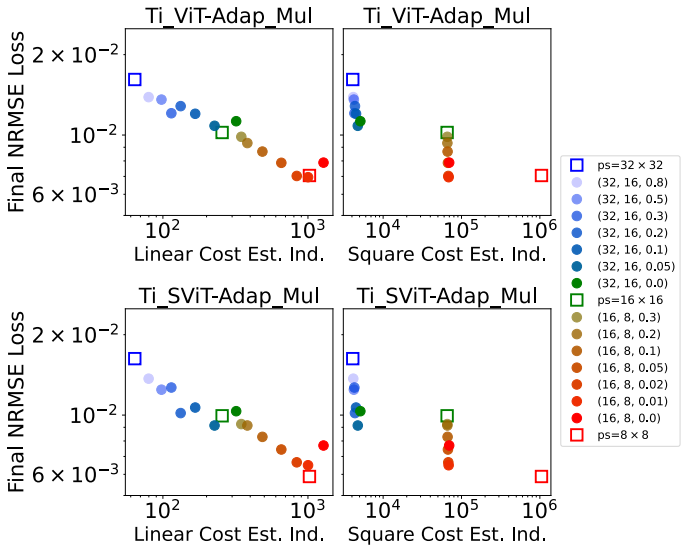

Figure 6: Final NRMSE loss for Tiny ViT (top) and SViT (bottom)with adaptive tokenization — **Adap_Mul** with hyperparamters (p_{x_1}, $p_{x_{\text{sts}}}$, γ_{sts}) — and constant patch sizes against linear cost estimation index (left) and square cost estimation index (right). As γ_{sts} moving from 1.0 to 0.0, predictive errors decrease dramatically; simultaneously, the linear cost in **Adap_Mul** transitions from ps=$p_{x_1} \times p_{y_1}$ to ps=$p_{x_{\text{sts}}} \times p_{y_{\text{sts}}}$, while quadratic cost increase is negligible.

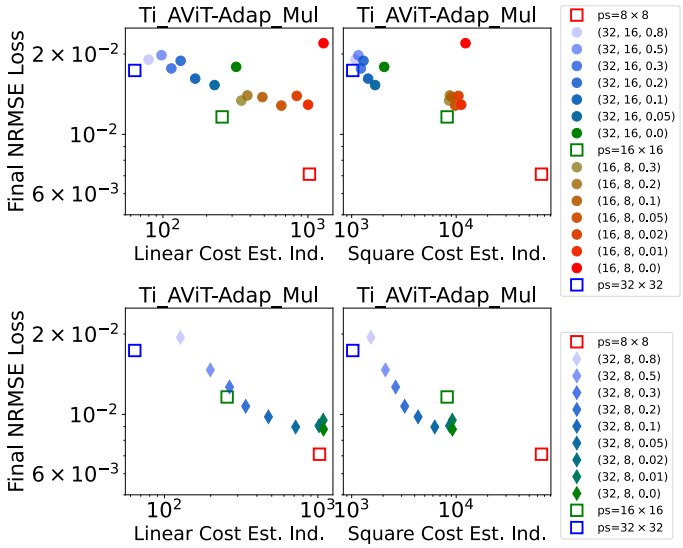

Figure 7: Final NRMSE loss for Tiny AViT with adaptive tokenization — **Adap_Mul** with hyperparamters (p_{x_1}, $p_{x_{\text{sts}}}$, γ_{sts})m— and constant patch sizes against linear cost estimation index (left) and square cost estimation index (right). Compared with ViT and SViT, AViT requires smaller STS patches to achieve similar accuracy improvements due to axial decomposition, demonstrating notable prediction accuracy gains with $p_{x_{\text{sts}}} = 8$ instead of $p_{x_{\text{sts}}} = 16$, for $p_{x_1} = 32$.

models pretrained on PDEBench or randomly initialized models ("*_INIT"), and fine-tune them on colliding thermals and MHD datasets with distinct physical variables. For fine-tuning each model, we either allow all model parameters to be tunable ("ALL") or freeze the attention blocks and limit training to the preprocessor, the tokenization module, and the postprocessor ("PREPOST"). Finally, for each initial model and fine-tuning configuration, we train four models with increasing amounts of fine-tuning data.

For the colliding thermals dataset, Figure 8 compares the test loss with full and limited fine-tuning using pretrained and randomly initialized models. The different training data sizes ranging from one set of colliding thermals time-trajectory to 24 sets of trajectories. The fine-tuning task is to predict the solution of the physical system at a lead time of t_{lead} uniformly sampled between 1 and 50 steps. An example of the true and predicted solutions in these four training configurations is illustrated in Figure 9.

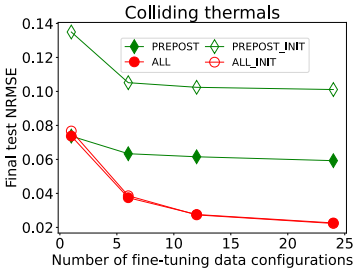

Figure 8: NRMSE loss for test set at different training data sizes in fine-tuning of colliding thermals at a maximum lead time of 50 steps, with full ("ALL") and limited ("PREPOST") fine-tuning using pretrained and randomly initialized models ("*_INIT").

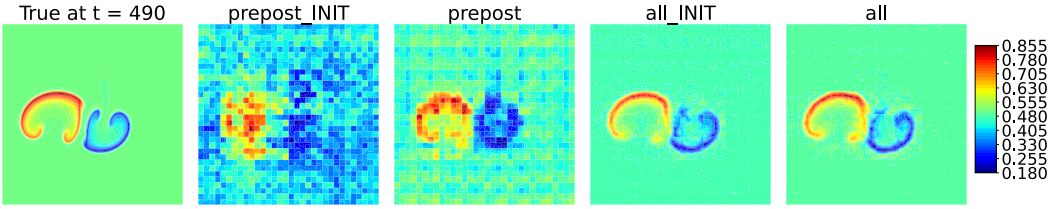

Figure 9: Temperature contours of true solution vs predicted solutions from four fine-tuned models (on 12 trajectories) at $t = 490$ from Ti-SViT models for a lead time of 40 in the collision of two thermal bubbles.

For the limited fine-tuning test with the colliding thermals dataset, the pretrained models achieve significantly lower error than starting from scratch with randomly initialized parameters. Moreover, while this advantage persists as the number of fine-tuning data increases, it is most pronounced in the low data configuration of learning from a single trajectory. Indeed, we find that limited fine-tuning with the pretrained models generalizes well even when learning from one trajectory, seeing only moderate improvements when run on the largest dataset size considered. Overall, the lower converged error from pretrained models suggests the frozen attention blocks clearly learned transferable knowledge during pretraining. For full fine-tuning, the accuracy is much better than limited fine-tuning as a result of the model being more expressive. The difference between the pretrained and randomly initialized models is much lower, being minor in the case of a single data configuration during training and vanishing as the amount of data increases.

For the MHD dataset, Figure 10 shows the final test NRMSE errors in lid-driven cavity flows after fine-tuning against data sizes when starting from pretrained and randomly initialized models for limited and full fine-tuning. The training dataset sizes used for fine-tuning range from 1 to 12 simulation configurations, with each configuration including approximately 1900 samples. The fine-tuning task is to predict the flow solution at a lead time of t_{lead} uniformly sampled between 1 and 100 steps. Contour plots from the true solution and the predicted solution from each training configuration are depicted in Figure 11.

Overall, the fine-tuning performance is a result of model expressibility, training data size, and the similarity between training and testing tasks. As with the colliding thermals dataset, pretrained models outperformed the randomly initialized models for both full and limited fine-tuning runs. However, the reduced expressibility of the limited fine-tuning configuration consistently shows an accuracy gap, even with more training data, as they cannot fully represent the data complexity. In contrast, full fine-tuning leads to more expressive models that can capture all training data information when trained on limited data but often show high test errors; as more training data is provided, they generalize well and lead to a convergent improved test error. In our

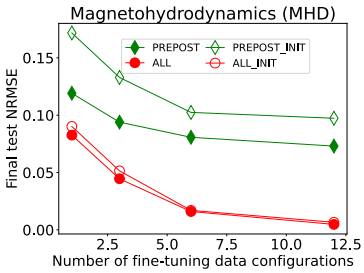

Figure 10: NRMSE loss for test set at different training data sizes in fine-tuning of lid-driven cavity MHD flows dataset at a maximum lead time of 100 steps, with full ("ALL") and limited ("PREPOST") fine-tuning using pretrained and randomly initialized models ("*_INIT").

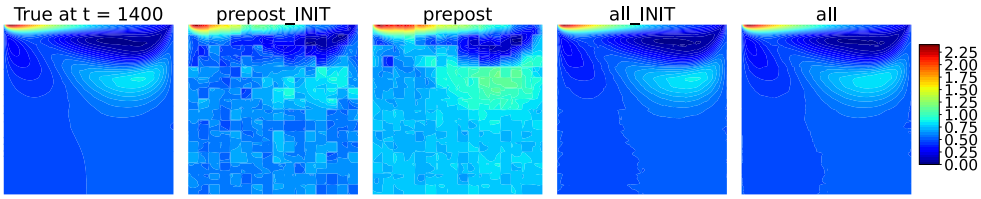

Figure 11: Contours of true horizontal magnetic field values B_x vs predicted solutions from four fine-tuned models (on 12 trajectories) at $t = 1400$ from Ti-SViT models for a lead time of 80 in lid-driven cavity MHD flows.

fine-tuning, the randomly initialized models perform well in testing even with a single data configuration, likely due to the similarity between training and testing tasks. Future work will explore more challenging scenarios with increased heterogeneity within the fine-tuning data.

While studies like McCabe et al. (2023) have demonstrated impressive outperformance from fine-tuning of pretrained models versus randomly initialized models, these fine-tuning tests were performed on data that, while distinct, was fully governed by physical equations and characterized by physical variables that were represented in the training data. Yet for a model that aims to be foundational for multiphysical systems, we argue that assessing model performance in more realistic settings, where equations like Navier-Stokes are coupled with those from other domains of physics, is a more informative test of the effectiveness of pretraining. Accordingly, we assess fine-tuning performance on physical systems that incorporate fluid flows, which are well-represented in PDEBench, with thermodynamics and electromagnetism, which are not. As reasonably anticipated, we find that advantages of pretraining are reduced in this more complex setting.

5 Discussion

In this paper, we make three contributions that will advance the development of foundation models for multiscale physical systems. First, we find that while some data efficiency is lost in a fully decoupled spatiotemporal attention scheme such as AViT, SViT provides an intriguing balance of computational and data efficiency versus the standard ViT approach. Yet using SViT alone does not sufficiently address the computational challenges associated with attention for high spatial resolutions. Second, we instead suggest that our adaptive tokenization scheme provides a promising approach for working with high resolution data. This sort of adaptivity has the potential to be both flexible and expressive enough to deal with the dynamic and sparse nature of the multiscale features in physical data. Third, we suggest an alternative path to evaluate foundation models for multiscale physical systems that focuses on fine-tuning problems involving out-of-distribution physics governed by different equations with distinct sets of physical variables. In two such settings, colliding thermals and magnetohydrodynamics, we find that while pretraining does provide an advantage, its impact is much more muted compared to fine-tuning on the same set of variables, suggesting additional effort is required to obtain truly foundational models in this space.

Acknowledgments

References

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer, 2021. URL https://arxiv.org/abs/2103.15691.

Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 2024.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4599–4603, June 2023.

F. Fambri, E. Zampa, S. Busto, L. Río-Martín, F. Hindenlang, E. Sonnendrücker, and M. Dumbser. A well-balanced and exactly divergence-free staggered semi-implicit hybrid finite volume / finite element scheme for the incompressible MHD equations. *Journal of Computational Physics*, 493:112493, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112493. URL https://www.sciencedirect.com/science/article/pii/S0021999123005880.

Qihang Fan, Quanzeng You, Xiaotian Han, Yongfei Liu, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. ViTAR: Vision Transformer with Any Resolution, 2024. URL https://arxiv.org/abs/2403.18361.

Zhou Hang, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-Conditional Transformers Are Universal PDE Solvers. *arXiv preprint arXiv:2405.17527*, 2024.

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.

Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas B. Moeslund. Which Tokens to Use? Investigating Token Reduction in Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 773–783, October 2023.

Jakob Drachmann Havtorn, Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. PMSViT: Dynamic Mixed-scale Tokenization for Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 838–848, 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient Foundation Models for PDEs. *arXiv preprint arXiv:2405.19101*, 2024.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. DeepSpeed Ulysses: System Optimizations for Enabling Training of Extreme Long Sequence Transformer Models, 2023. URL https://arxiv.org/abs/2309.14509.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4804–4814, 2022.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models, 2024. URL https://arxiv.org/abs/2402.17177.

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.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12309–12318, 2022.

Matt Norman. mrnorman/miniWeather, September 2024. URL https://github.com/mrnorman/miniWeather. original-date: 2018-07-24T02:29:06Z.

Matthew R Norman. miniWeather. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2020.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. In *European Conference on Computer Vision*, pp. 497–515. Springer, 2022.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10809–10818, 2022.

Enzhi Zhang, Isaac Lyngaas, Peng Chen, Xiao Wang, Jun Igarashi, Yuankai Huo, Mohamed Wahib, and Masaharu Munetomo. Adaptive Patching for High-resolution Image Segmentation with Transformers, 2024. URL https://arxiv.org/abs/2404.09707.