A. appendix

A.1. LLM USAGE

During the manuscript writing and revision process, we used a Large Language Model (LLM) to assist. Specifically, LLM was used to improve the accuracy and readability of the language, and to help ensure the overall structure and clarity of the paper. This tool primarily assisted with tasks such as sentence reconstruction, grammatical proofreading, and improving text coherence.

A.2. Experimental Details

Pre-training. We pre-trained the model on 8 NVIDIA RTX 4090 GPUs using the Adam optimizer with an initial learning rate of 1.0×10^{-3} and a cyclic learning rate schedule (cycle), including 200 warm-up epochs. The total training lasted 1000 epochs with a batch size of 32. To mitigate the effects of varying dataset sizes, training weights were assigned to each dataset. During training, we used $T = 10$ time steps to predict the next frame, maintaining consistency with the original settings of most datasets. The details are shown in Table 7.

Fine-tuning. In the fine-tuning stage, we loaded the pre-trained weights and performed 200-epoch and 500-epoch fine-tuning on each subset. The key module of the model is the nested MoE layer, whose parameters are shared across different frequency components along the channel dimension, enabling cross-level expert collaboration.

A.3. Data Preprocessing and Sampling

Data Padding and Masking. Different PDE datasets vary in resolution, number of variables, and geometric configurations. If we directly sample from the raw data, the resulting batch will have large variations in size, leading to unbalanced training loads and reduced efficiency in modern multi-GPU training. Here, we adopt the padding and masking strategy from DPOT. First, we select a fixed resolution $H = 128$, which matches a considerable portion of the datasets. Datasets with lower resolutions are upsampled to H via interpolation, while those with higher resolutions are randomly downsampled or interpolated to H. Second, to unify the number of variables across different PDEs, we pad all datasets along the channel dimension (e.g., filling with ones) to match the maximum number of channels. For datasets with irregular geometries, an additional mask channel is introduced to encode the specific geometric configuration of each PDE instance.

Balanced Data Sampling. When training with multiple PDE datasets, differences among datasets can lead to unbalanced training progress and inefficiency. To address this issue, we adopt the sampling strategy from DPOT, which balances the sampling probabilities across datasets during training. Our goal is to ensure that each dataset is represented equally throughout the training process. Let $|D_k|$ denote the number of samples in the k-th dataset ($1 \le k \le K$), and assign a weight w_k to each dataset to indicate its relative importance. Then, the sampling probability from dataset D_k is defined as:

$$p_k = \frac{w_k}{K\,|D_k|\,\sum_k w_k}$$

We can observe that the sampling probability depends on the weight w_k rather than the dataset size $|D_k|$, which helps mitigate gradient imbalance caused by dataset size disparities.

A.4. Limitations and Conclusions

We use DPOT as our primary baseline and adopt its data processing strategies, including adding noise, data padding, and balanced data sampling. However, our core model differs from DPOT, which is based on AFNO, while our network architecture employs a nested MoE. MoE-POT, our work, also incorporates a MoE architecture, but uses only a single-layer MoE and primarily improves upon the frequency convolution in AFNO. In contrast, our proposed nested MoE architecture processes PDE features at both macroscopic and microscopic levels, achieving an effective fusion of frequency domain and spatiotemporal domain features.

Due to resource constraints, our model is currently only implemented with one parameter size, but this version has already demonstrated good accuracy and generalization ability. Combining the results of scaling experiments and interpretability analysis, we validate the model's effectiveness and show that it can be scaled to versions with different parameter sizes. Considering the diversity of expert and activation numbers, future work can explore optimal parameter configurations to further improve model performance.

A.5. Detailed Information of Datasets

We list the configurations of the PDE datasets used for pre-training along with detailed descriptions of the governing partial differential equations:

FNO-v: This dataset focuses on the temporal evolution of the two-dimensional incompressible fluid vorticity field $w(x,t)$, where $(x,t) \in [0,1]^2 \times [0,T]$. The dynamics are governed by the two-dimensional Navier–Stokes equations in the vorticity–streamfunction formulation:

$$\partial_t w + u \cdot \nabla w = \nu \Delta w + f(x), \qquad \nabla \cdot u = 0, \tag{16}$$

where u denotes the velocity field, ν is the viscosity coefficient, Δ represents the Laplace operator, and $f(x)$ denotes the external forcing term. By varying the viscosity ν, the dataset provides fluid dynamics simulations under different flow regimes, enabling the study of how viscosity influences the evolution of vortex structures.

Table 7. Setting of the Attention Module.

Dim	Ratio	Layers	Heads	Routed$_1$	Shared$_1$	Top-k_1	Routed$_2$	Shared$_2$	Top-k_2	Model Size	Activated Size
512	1	2	4	1	6	2	1	6	2	83M	13M

Table 8. Train and test set sizes of the PDE datasets used for pre-training.

	FNO-ν			PDEBench CNS-(η, ζ), DR, SWE						PDEArena		CFDBench
	1e-5	1e-4	1e-3	1,0.1	1,0.01	0.1,0.1	0.1,0.01	DR	SWE	NS	NS-cond	-
Train set size	100	9800	1000	9000	9000	9000	9000	900	900	6500	3100	9000
Test set size	200	200	200	1000	1000	1000	1000	100	100	1300	600	1000

PDEBench-CMS: This dataset focuses on the numerical simulation of compressible fluid mechanics (CMS). The goal is to predict the temporal evolution of the velocity field $u(x,t)$, the pressure field $p(x,t)$, and the density field $\rho(x,t)$ over the spatio-temporal domain $(x,t) \in [0,1]^2 \times [0,1]$. The data are generated based on the governing equations of compressible fluid dynamics, which consist of the conservation of mass, momentum, and energy:

$$\partial_t \rho + \nabla \cdot (\rho u) = 0, \tag{17}$$

$$\rho\left(\partial_t u + u \cdot \nabla u\right) = -\nabla p + \eta \Delta u + \left(\zeta + \tfrac{\eta}{3}\right) \nabla(\nabla \cdot u), \tag{18}$$

$$\partial_t \left(\tfrac{3}{2} p + \tfrac{\rho u^2}{2}\right) = -\nabla \cdot \left[\left(\varepsilon + p + \tfrac{\rho u^2}{2}\right) u - u \cdot \sigma'\right], \tag{19}$$

where η denotes the shear viscosity coefficient and ζ the bulk viscosity coefficient.ε is the energy density and σ' is the stress tensor.

PDEBench-SWE: The dataset is derived from PDEBench and focuses on the numerical simulation of the Shallow Water Equations (SWE). The objective is to predict the water depth field $h(x,t)$ over the spatiotemporal domain $(x,t) \in [-1,1]^2 \times [0,5]$. The SWE is a set of approximate governing equations widely used in ocean dynamics, flood modeling, and geomorphological evolution studies. The governing equations are given as follows:

$$\partial_t h + \nabla \cdot (hu) = 0, \tag{20}$$

$$\partial_t (hu) + \nabla \cdot \left(\tfrac{1}{2} h u^2 + \tfrac{1}{2} g r h^2\right) = -g r h \nabla b, \tag{21}$$

PDEBench-DR: The dataset is derived from PDEBench and focuses on the numerical simulation of diffusion–reaction (DR) systems. The objective is to predict the density field $u(x,t)$ over the spatiotemporal domain $(x,t) \in [-2.5, 2.5]^2 \times [0,1]$. The governing equation is given by:

$$\partial_t u = D \nabla^2 u + R(u), \tag{22}$$

where D is the diffusion coefficient and $R(u)$ denotes the nonlinear reaction term.

PDEArena: The dataset is derived from PDEArena and focuses on the numerical simulation of incompressible Navier–Stokes (NS) flows. The objective is to predict the velocity field $u(x,t)$, pressure field $p(x,t)$, and density field $\rho(x,t)$ over the spatiotemporal domain $(x,t) \in [0,32]^2 \times [0,24]$.

The 2D incompressible Navier–Stokes equations are given by:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u}, \tag{23}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{24}$$

where $\mathbf{u} = (u, v)^\top$ is the velocity field, p is the pressure, and ν is the kinematic viscosity.

NS-cond introduces additional physical conditions such as forcing fields $\mathbf{f}(\mathbf{x}, t)$ or spatially varying viscosity $\nu(\mathbf{x})$:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu(\mathbf{x}) \Delta \mathbf{u} + \mathbf{f}(\mathbf{x}, t), \tag{25}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{26}$$

Here, $\mathbf{f}(\mathbf{x}, t)$ denotes external forcing and $\nu(\mathbf{x})$ can vary spatially.

CFDBench: The dataset is derived from CFDBench and focuses on the numerical simulation of incompressible or weakly compressible flows in irregular geometries. The objective is to predict the velocity field $u(x,t)$ and the pressure field $p(x,t)$ over domains with complex boundaries. The governing equations are given as follows:

$$\partial_t (\rho u) + \nabla \cdot (\rho u^2) = -\nabla p + \nabla \cdot \mu(\nabla u + \nabla u^\top), \tag{27}$$

$$\nabla \cdot (\rho u) = 0, \tag{28}$$

where ρ is the fluid density, u is the velocity field, p is the pressure, and μ denotes the viscosity coefficient.

A.6. Open access to data and code

The code is available at: https://github.com/Event-AHU/OpenFusion/tree/main/NESTOR.

Table 9. Comparison with MoE-POT in pre-training across six datasets. The evaluation metric is L2RE. The best result within each part is highlighted in **bold**.

L2RE	Activated	FNO-ν		PDEBench			CFDBench
Model	Params	1e-5	1e-3	0.1,0.01	SWE	DR	-
MoE-POT	17M	0.0682	0.00768	**0.0105**	0.00640	0.0411	**0.00529**
Ours	13M	**0.0674**	**0.00763**	0.0159	**0.00449**	**0.0184**	0.00911

A.7. Supplementary Experiments

Comparison with Recent Methods and Transfer to Additional Downstream Tasks. First, under a mixed pre-training setting that includes six datasets, we compare the proposed model with MoE-POT. As shown in Table 9, our method achieves new best results on four out of the six datasets.

Next, we fine-tune the pre-trained models of Poseidon and MoE-POT, and compare them with our method on two unseen downstream tasks (Wave-Layer and Wave-Gauss). As shown in Table 10, our method demonstrates strong generalization and transfer capability on these tasks.

Table 10. Comparison of generalization and transfer ability.

Model	Activated Params	Wave-Layer	Wave-Gauss
Poseidon-T	21M	0.29	0.29
Poseidon-B	158M	0.21	0.24
MoE-POT-T	17M	0.07	0.07
MoE-POT-S	90M	0.05	0.06
Nestor	13M	**0.0092**	**0.0067**

Finally, we compare the pre-training performance on the mixed dataset with OmniArch and Unisolver. The results in Table 11 show that our method achieves superior overall performance. It should be noted that since OmniArch and Unisolver are large-scale models, forcing them to use a smaller parameter budget to match our experimental setting may lead to some performance degradation.

Table 11. Performance comparison with other SOTA models.

Model	**Params**	**1e-5**	**1e-3**	**0.1, 0.01**	**swe**	**dr**	**cfdbench**
OmniArch	11M	0.3135	0.2434	0.3495	0.0984	0.2103	0.1236
Unisolver	12M	0.0680	0.0112	0.4785	0.1167	0.1390	0.0163
Nestor	13M	**0.0674**	**0.0076**	**0.0159**	**0.0045**	**0.0184**	**0.0091**

Comparison with FreqMoE and Analysis of the Routing Design. Since the code of FreqMoE is not publicly available, we implement a frequency-domain routing variant following its core idea. Specifically, we first decompose the input data into the frequency domain and then feed the frequency-domain features into the routing module for expert selection. The comparison results are shown in Table 12. Our proposed image-level routing consistently outperforms frequency-domain routing across all tasks.Furthermore, Table 13 presents the activation distribution of different experts. It can be observed that image-level routing leads to a more concentrated expert specialization pattern, enabling clearer differentiation between different types of PDEs (e.g., DR is mainly handled by experts 0 and 5). In contrast, the frequency-domain routing variant exhibits more dispersed expert activations, indicating a lower degree of expert specialization.

Under the setting of large-scale mixed PDE datasets, effective identification of equation types can transform the original complex unified modeling problem into a relatively simpler task conditioned on known types. However, the dynamics of complex PDEs are often difficult to characterize clearly through simple frequency-domain decomposition. Noise and spectral interference may blur the boundaries between experts, which is also reflected in the more scattered activation ratios shown in Table 13. In contrast, our proposed image-level routing leverages global visual features for expert selection, resulting in clearer expert specialization. By further integrating token-level experts for local modeling, the overall architecture forms a global-to-local modeling strategy. Such a strategy has been widely validated in the computer vision community as a robust and effective framework for capturing complex physical evolution processes.

Table 12. Comparison on different routing domain.

Routing Domain	**1e-5**	**1e-3**	**0.1, 0.01**	**swe**	**dr**	**cfdbench**
Frequency	0.2373	0.0234	0.1045	0.0700	0.9560	0.0235
Spatial	**0.0674**	**0.0076**	**0.0159**	**0.0045**	**0.0184**	**0.0091**

Table 13. Expert routing distribution on different datasets.(%)

Dataset	**Stage**	**Global image features**						**Frequency-based features**					
		Exp 0	**Exp 1**	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**	**Exp 0**	**Exp 1**	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**
M1(-1,-1)	FT-300	0.00	**50.00**	0.00	**50.00**	0.00	0.00	0.00	0.00	7.81	**42.19**	**46.88**	3.12
	FT-500	0.00	**50.00**	0.00	0.00	**49.75**	0.25	0.00	0.00	7.81	**42.19**	**46.88**	3.12
M-1(-1,-1)	FT-300	**50.00**	**50.00**	0.00	0.00	0.00	0.00	0.00	0.00	**43.16**	0.00	15.79	**41.05**
	FT-500	**50.00**	**50.00**	0.00	0.00	0.00	0.00	0.00	0.00	**47.02**	0.00	7.05	**45.93**
SWE	FT-300	0.00	0.00	**50.00**	**50.00**	0.00	0.00	0.00	**50.00**	0.00	**50.00**	0.00	0.00
	FT-500	0.00	0.00	**50.00**	**50.00**	0.00	0.00	0.00	**50.00**	0.00	**50.00**	0.00	0.00
DR	FT-300	**50.00**	0.00	0.00	0.00	0.00	**50.00**	19.12	0.00	**49.97**	0.00	0.00	**30.91**
	FT-500	**50.00**	0.00	0.00	0.00	0.00	**50.00**	17.42	0.00	**49.28**	0.00	0.00	**33.30**

Module Contribution Analysis. To analyze the contribution of each module to the overall performance, we further conduct ablation studies. As shown in Table 14, the image-level MoE routing is the key driver of model performance; removing this component leads to a significant degradation in performance. This is because, without image-level routing to distinguish different PDE systems, the experts are forced to share representations across heterogeneous dynamics, which may lead to negative transfer. Therefore, this gating mechanism is crucial for effectively modeling multiple types of PDE systems.

Table 14. Ablation study of different components.

Method	**1,0.1**	**1,0.01**	**0.1,0.1**	**0.1,0.01**	**dr**	**swe**	**Avg L2**	**Promotion**
Ours	0.0144	0.0355	0.0135	0.0178	0.0282	0.0045	**0.0173**	–
w/o Image-level MoE	0.0153	0.0375	0.0134	0.0200	0.0084	0.0309	0.0209	0.0036
w/o Sub-MoE	0.0157	0.0393	0.0130	0.0209	0.0245	0.0049	0.0197	0.0024
w/o Load Balance Loss	0.0135	0.0335	0.0109	0.0159	0.0265	0.0062	0.0178	0.0005

A.8. Visualization

For each specific subtask, we first load the model weights pretrained on large-scale PDE datasets, and then fine-tune the model for the subtask. During fine-tuning, the model can adapt to the data distribution and equation characteristics of each subtask. The visualization of the prediction results is shown in the figure. For each data series, we select a representative equation to illustrate the model's performance across different tasks. These visualizations allow us to observe the model's ability to capture spatiotemporal trends, local details, and global patterns, thereby demonstrating the effectiveness and advantages of the pretrained weights in downstream tasks.

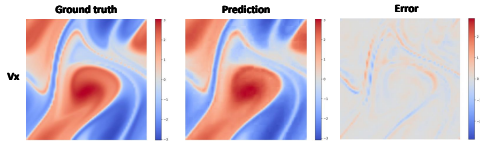

Figure 6. FNO series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.

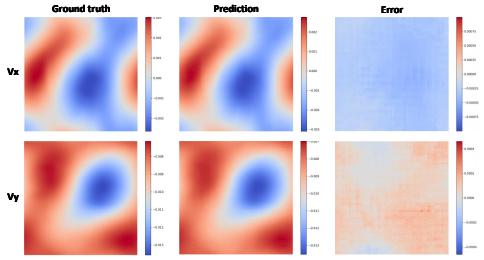

Figure 7. PDEBench series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.

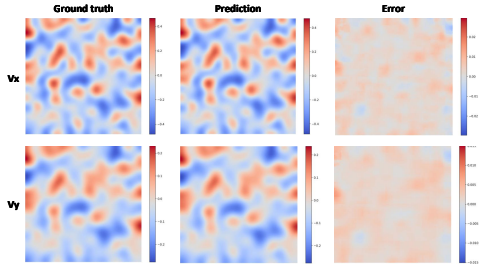

Figure 8. DR series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.

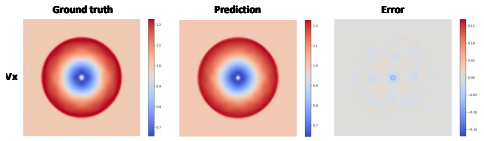

Figure 9. SWE series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.

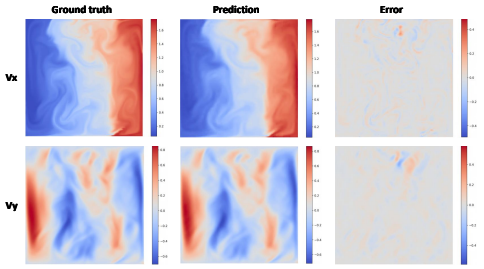

Figure 10. PDEArena series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.

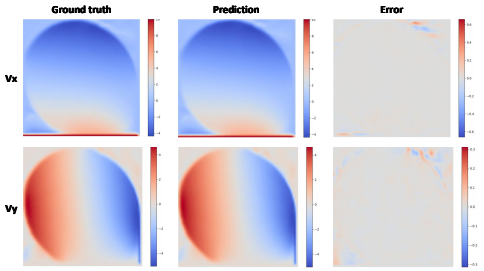

Figure 11. CFDBench series of result visualizations. (1) The first column shows the true value, the second column shows the model prediction value, and the third column shows the corresponding error. (2) Each row is the predicted physical quantity.