ICML | 2023

Fortieth International Conference on Machine Learning

Solving High-Dimensional PDEs with Latent Spectral Models

Haixu Wu [1] **Tengge Hu** [1] **Huakun Luo** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Haixu Wu

Tengge Hu

Huakun Luo

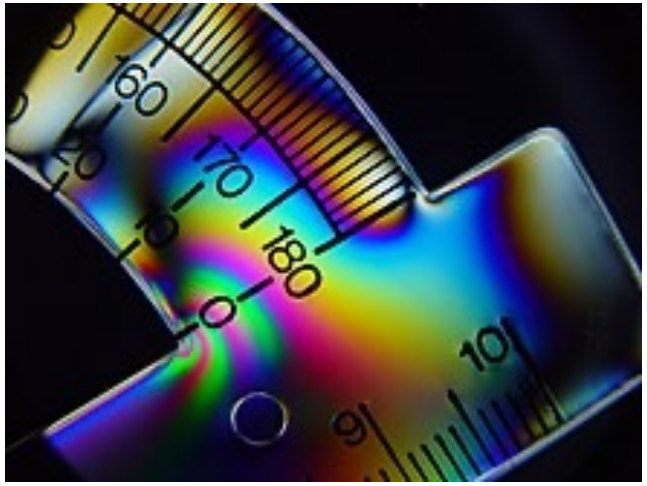
Jianmin Wang

Mingsheng Long

Real-world phenomena

Turbulence

Atmospheric circulation



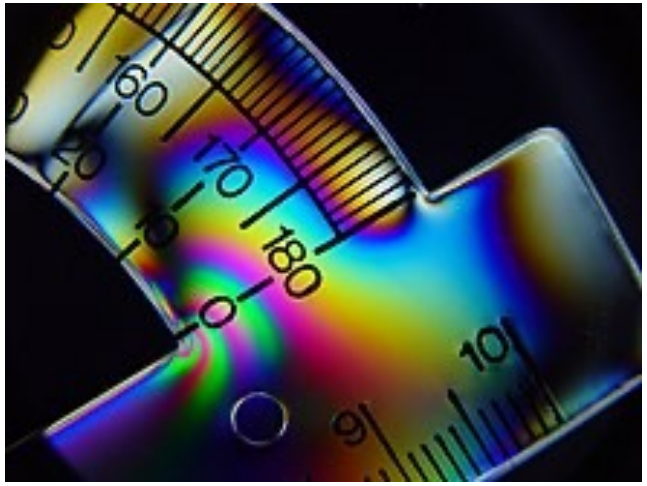
Stress

How to understand the world?

Real-world phenomena

Turbulence

Atmospheric circulation



Stress

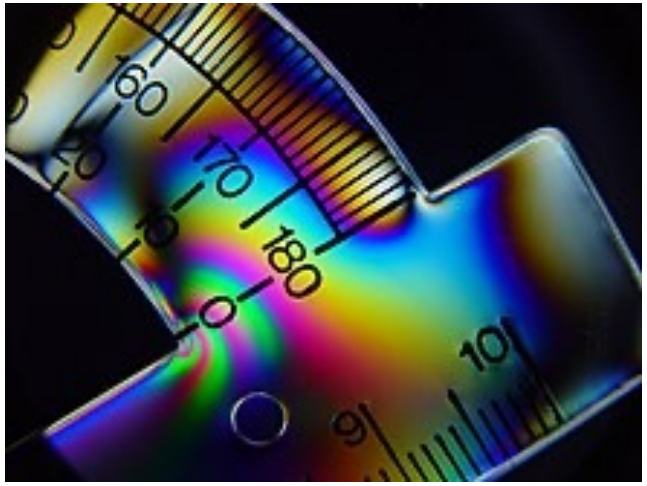
How to understand the world?

Images? Videos?

Real-world phenomena

Turbulence

Atmospheric circulation



Stress

Beyond appearances, these phenomena are governed by **scientific rules**.

Partial Differential Equations (PDEs)

- Fluid physics:

Navier-Stokes Equation for fluid dynamics

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \nabla \boldsymbol{U} = \boldsymbol{f} + \frac{1}{\rho} \nabla \cdot (\boldsymbol{T}_{ij} \boldsymbol{e}_i \boldsymbol{e}_j)$$

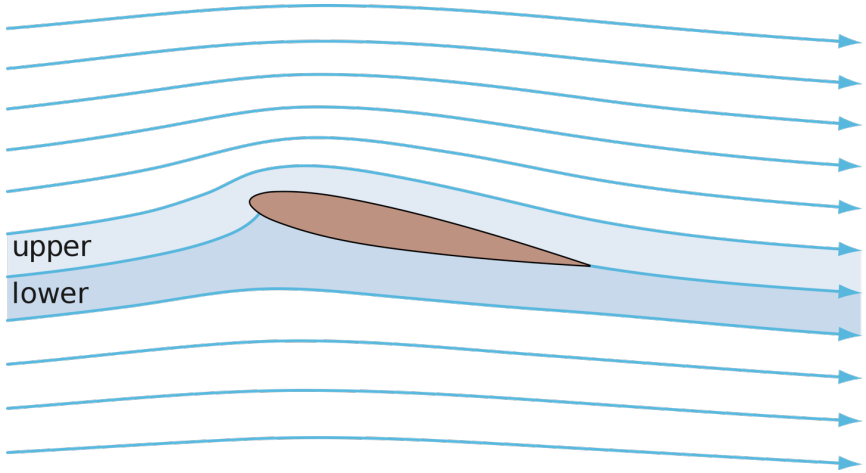
$$\frac{\partial (e + \frac{1}{2}\boldsymbol{U}^2)}{\partial t} + \boldsymbol{U} \cdot \nabla (e + \frac{1}{2}\boldsymbol{U}^2) = \boldsymbol{f} \cdot \boldsymbol{U} + \frac{1}{\rho} \nabla \cdot (\boldsymbol{U} \cdot \boldsymbol{T}_{ij} \boldsymbol{e}_i \boldsymbol{e}_j) + \frac{\lambda}{\rho} \Delta T$$

- Solid physics:

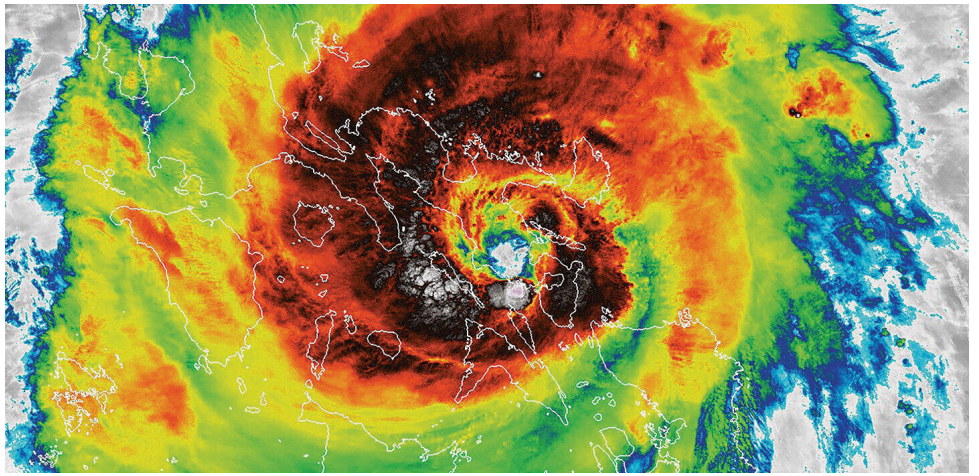
$$\rho^s \frac{\partial^2 \boldsymbol{u}}{\partial t^2} + \nabla \cdot \boldsymbol{\sigma} = 0$$

Inner stress of solid materials

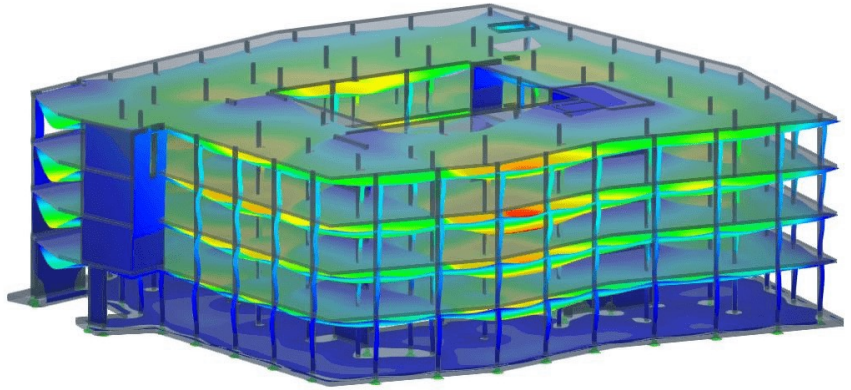
Wide Applications



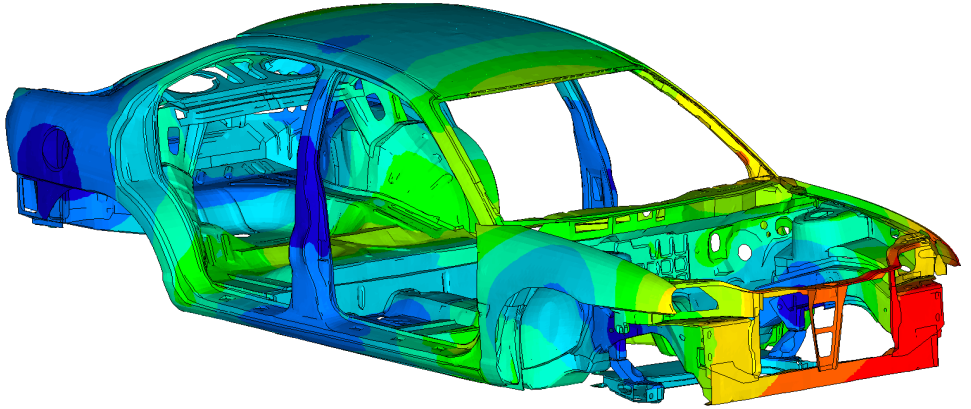
Airfoil design



Weather forecasting



Civil engineering



Vehicle manufacturing

Solving PDEs: Discretization

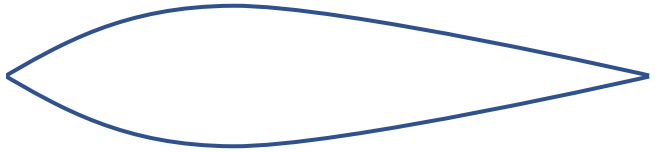
Infinite–dimensional PDE solutions

Discretization

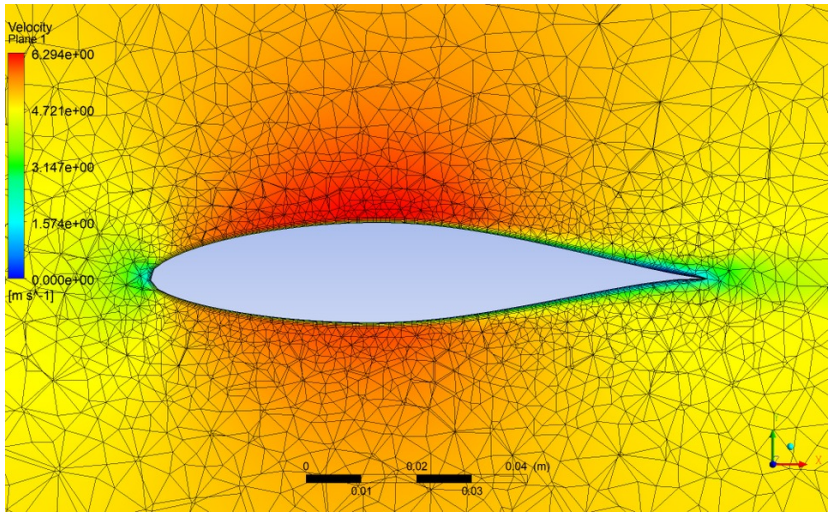
High-dimensional coordinate spaces

$\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{D}$

$\mathcal{D}$ is the mesh point set



Spatial continuous



Solving PDEs: Discretization

Infinite–dimensional PDE solutions

Discretization

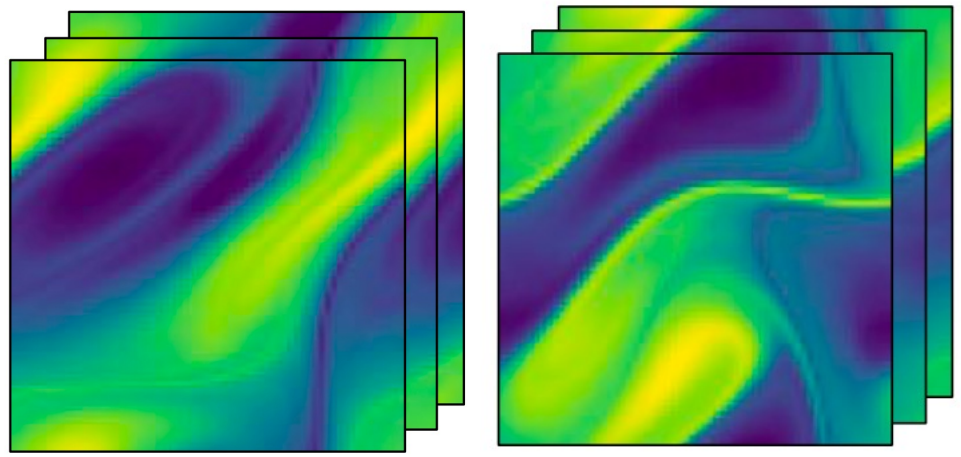
High-dimensional coordinate spaces

$\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{D}$

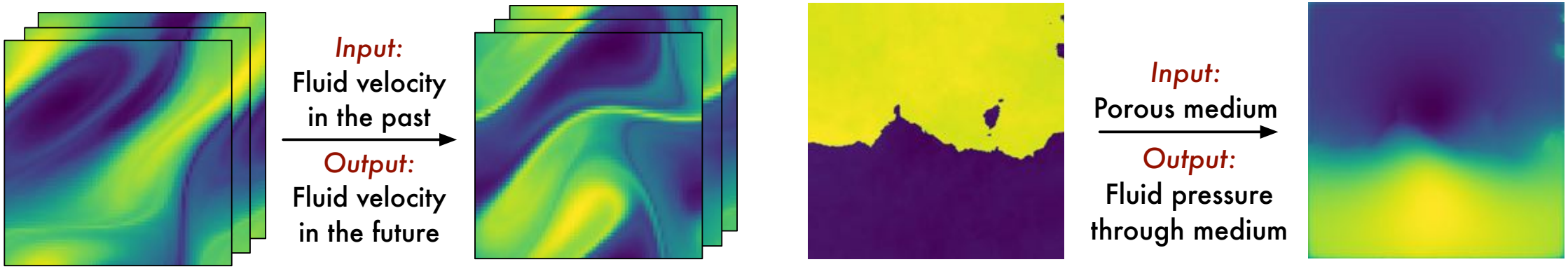
$\mathcal{D}$ is the grid point set

Spatiotemporal Continuous

Navier-Stokes Equation

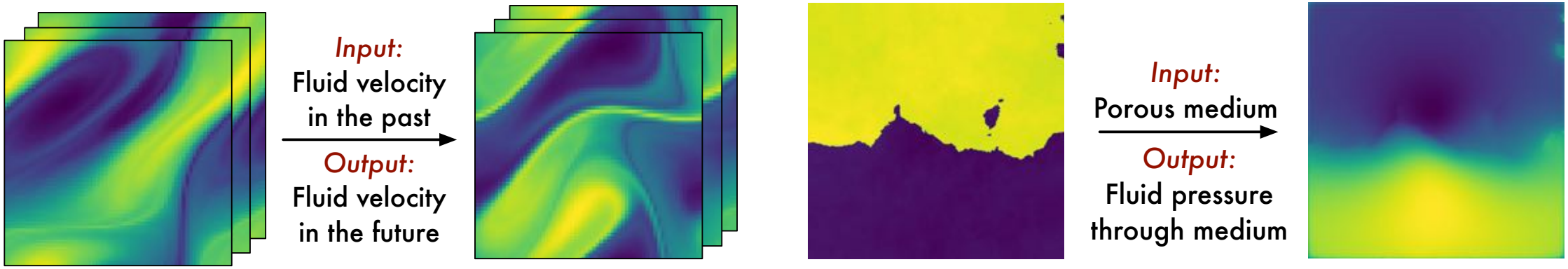


Challenges in Solving High-dimensional PDEs



- Curse of dimensionality → Huge computation cost
- Intricate interactions among physical variates of coupled equations → Complex mappings

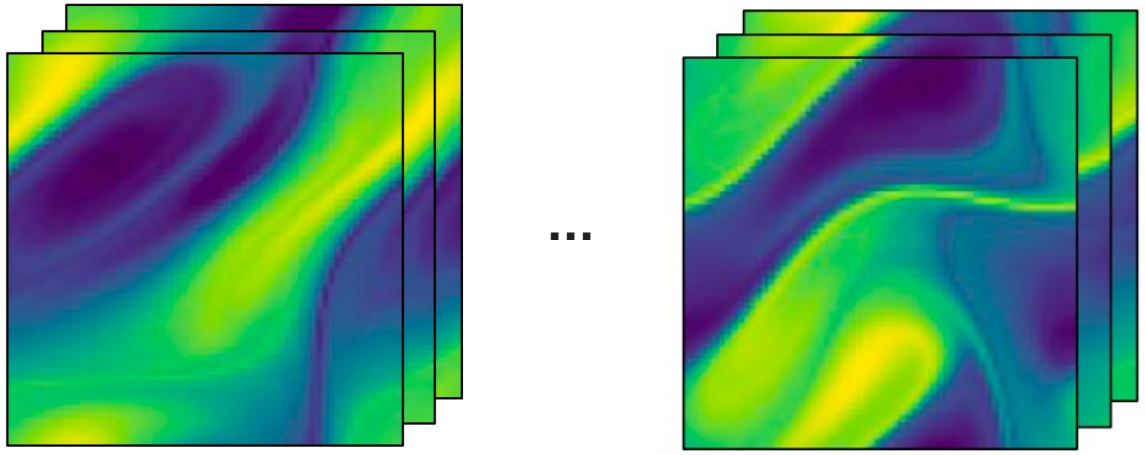
Challenges in Solving High-dimensional PDEs



- Curse of dimensionality → Huge computation cost
- Intricate interactions among physical variates of coupled equations → Complex mappings

How to efficiently and precisely approximate complex mappings between high-dimensional input-output pairs?

Motivation

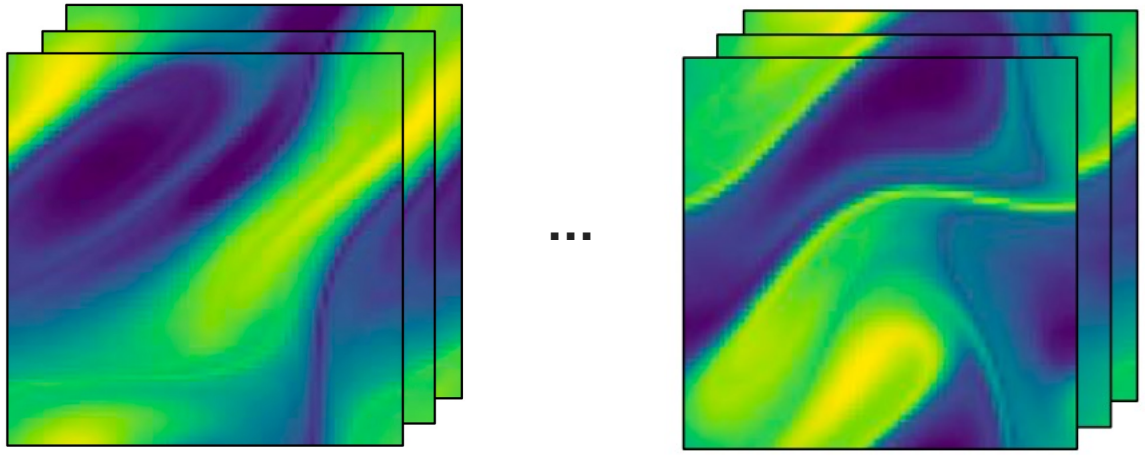


Multitudinous Data

Following the same PDE constraint

Manifold Hypothesis: *Real-world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space.*

Motivation



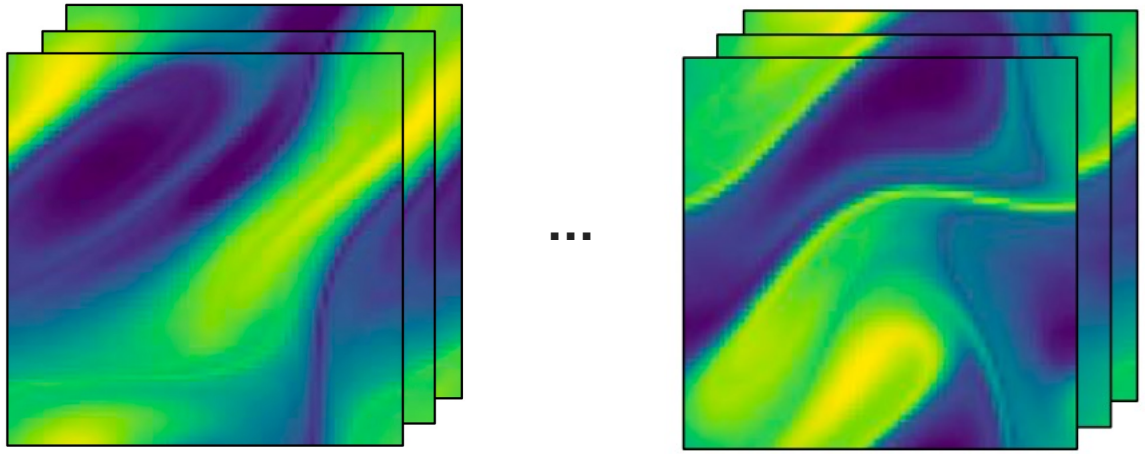
Multitudinous Data

Following the same PDE constraint

Manifold Hypothesis: *Real-world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space.*

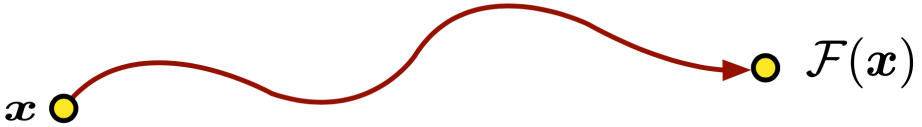
1. High-dimensional data can be projected to a more compact latent space

Motivation



How to approximate

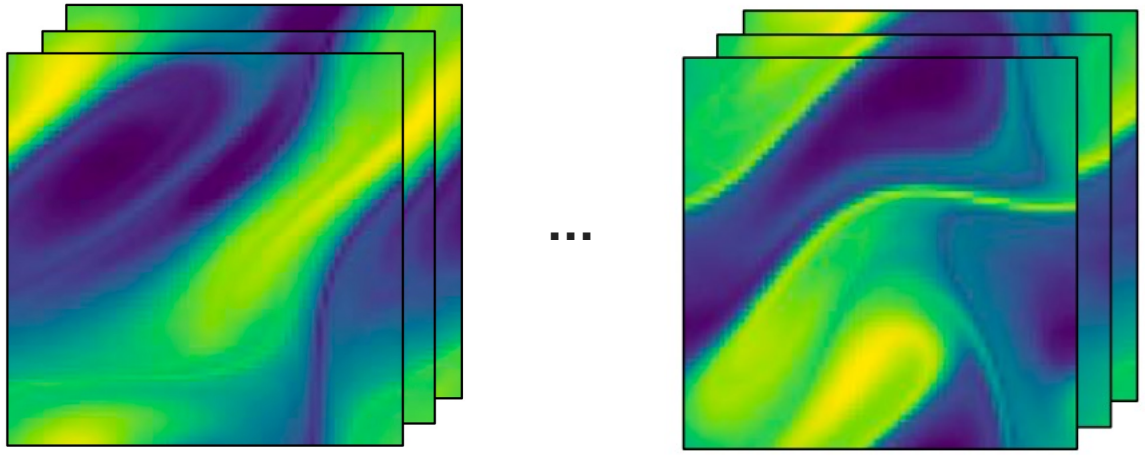
complex input-output mappings?



Previous Methods: Directly approximating with a single deep model

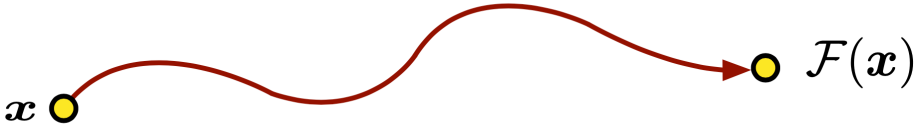
Suffer from optimization problem and limited performance

Motivation



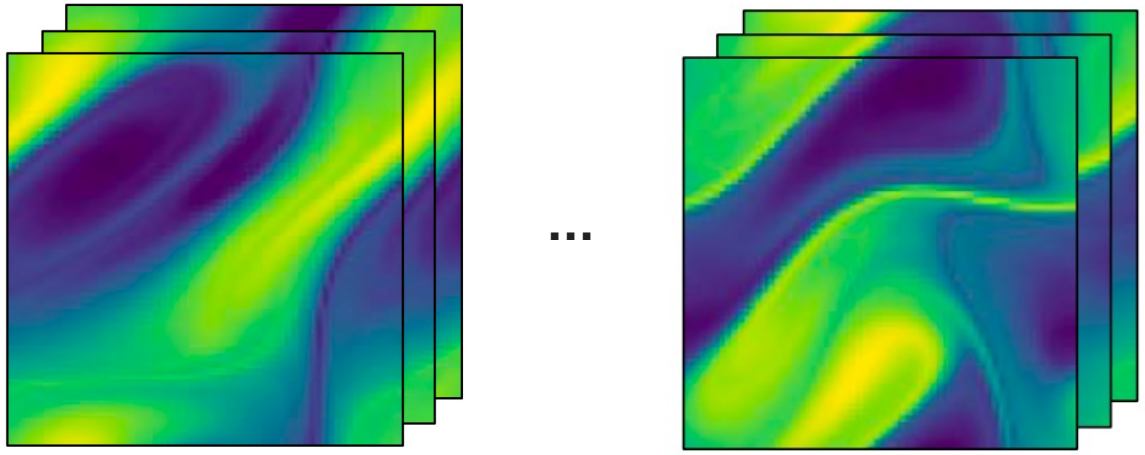
How to approximate

complex input-output mappings?



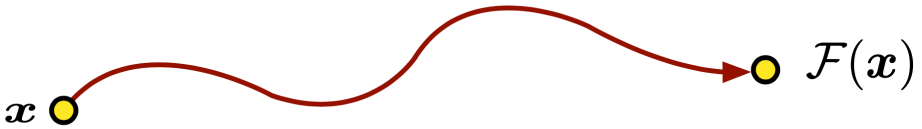
Spectral Methods: *approximate solution f of a certain PDE as a finite sum of N orthogonal basis functions $\{f_1, f_2, \cdots, f_N\}$, that is: $f \approx f^N = \sum_{i=1}^{N} w_i f_i$.*

Motivation



How to approximate

complex input-output mappings?



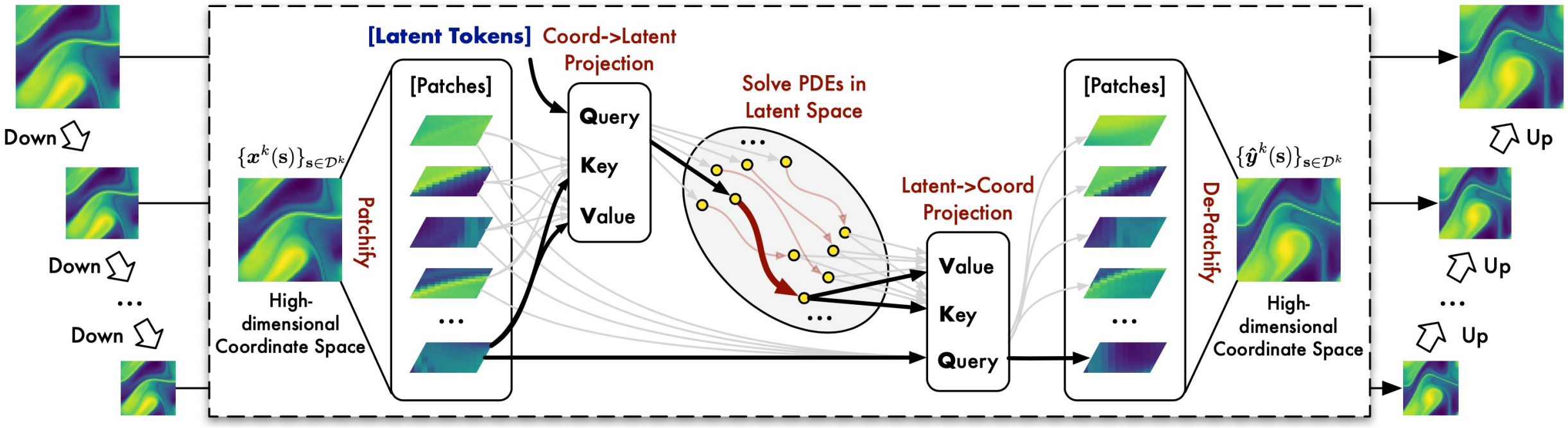
Spectral Methods: *approximate solution f of a certain PDE as a finite sum of N orthogonal basis functions $\{f_1, f_2, \cdots, f_N\}$, that is: $f \approx f^N = \sum_{i=1}^{N} w_i f_i$.*

2. Learning multiple basis operators for approximation

Latent Spectral Models (LSM)

	Previous Methods	LSM (ours)
Solving Process	Solving in the **coordinate space** • Huge computation cost • Making input-output mappings extremely complex	Solving in the **latent space** • Efficient computation • Highlight the inherent physics properties
Mapping approximation	Directly learning **a single operator** • Fail in approximating complex mappings • Lack of theoretical guarantee	Learning **multiple basis operators** • Nice approximating and convergence properties under theoretical guarantee

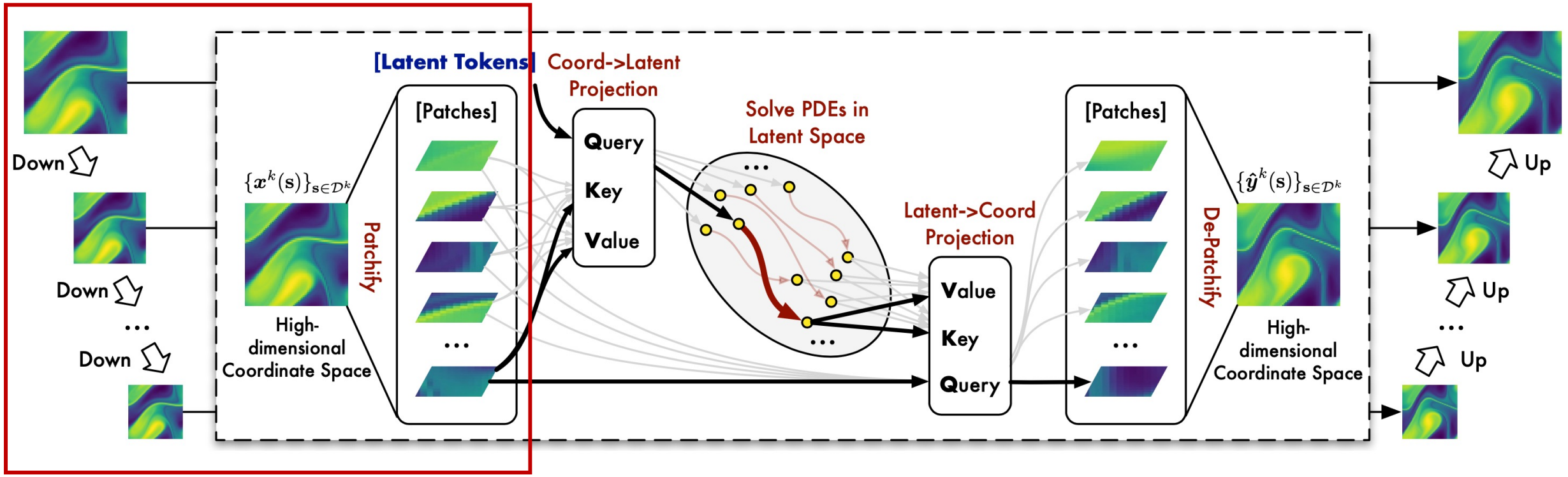
Overall design of LSM



LSM with *Hierarchical Projection Network* and *Neural Spectral Block*

① Coor → Latent ② Solving in the Latent Space ③ Latent → Coor

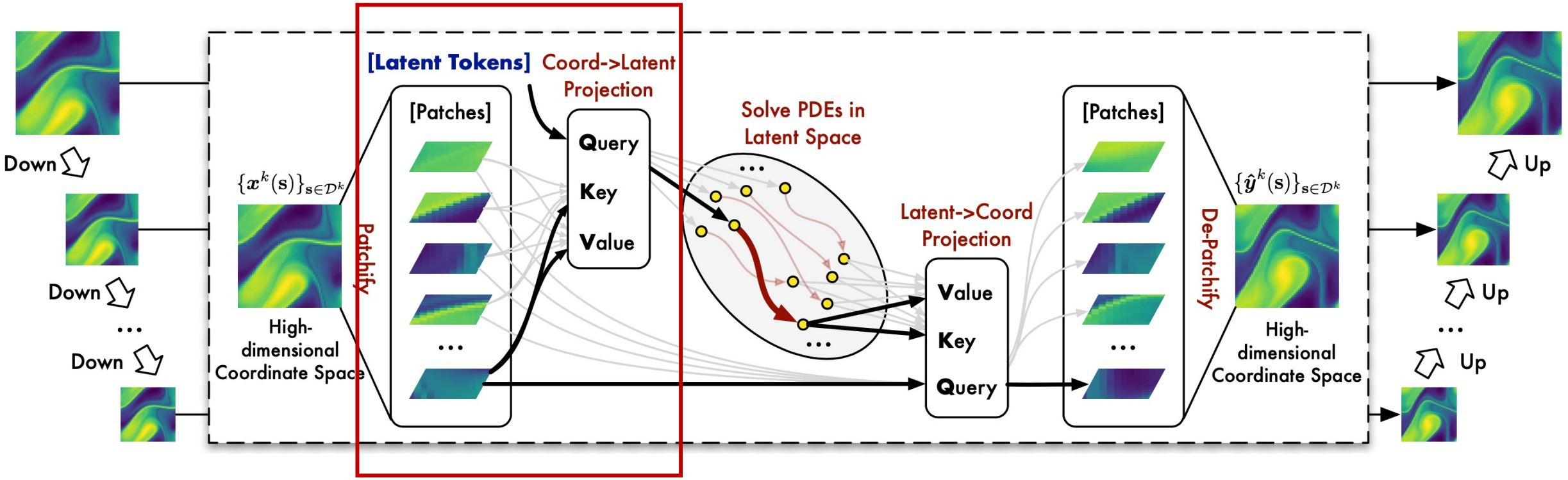
Hierarchical Projection Network



① ***Multiscale patchified architecture*** → *Solve PDEs in different regions and scales*

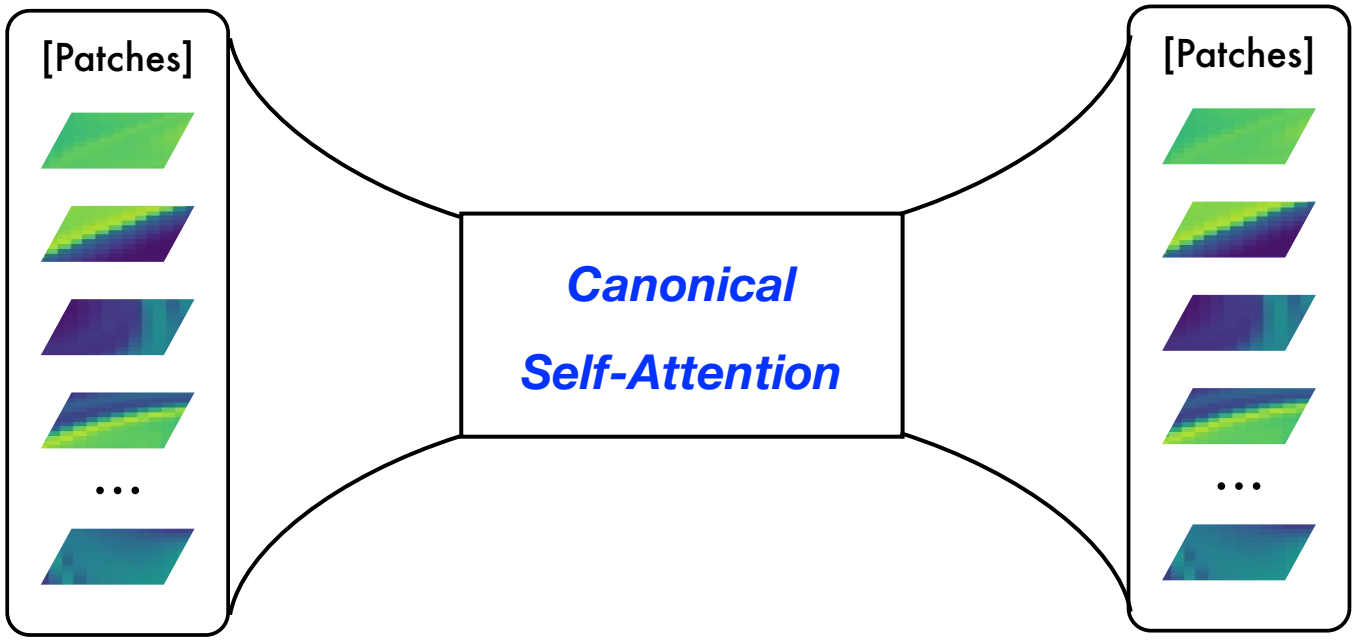
PDEs always present different physical states according to the observed scales and regions.

Hierarchical Projection Network



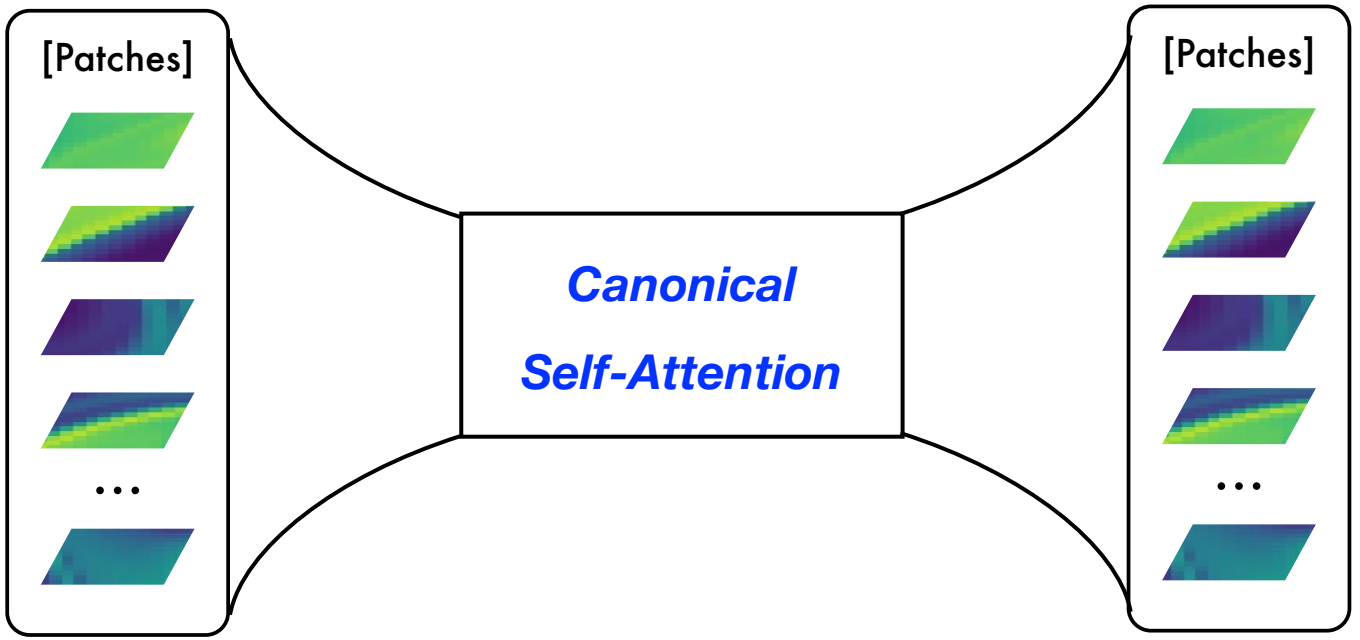
② ***Attention-based projector*** → *Remove unwieldy coordinate information*

Hierarchical Projection Network



② ***Attention-based projector*** → *Remove unwieldy coordinate information*

Hierarchical Projection Network

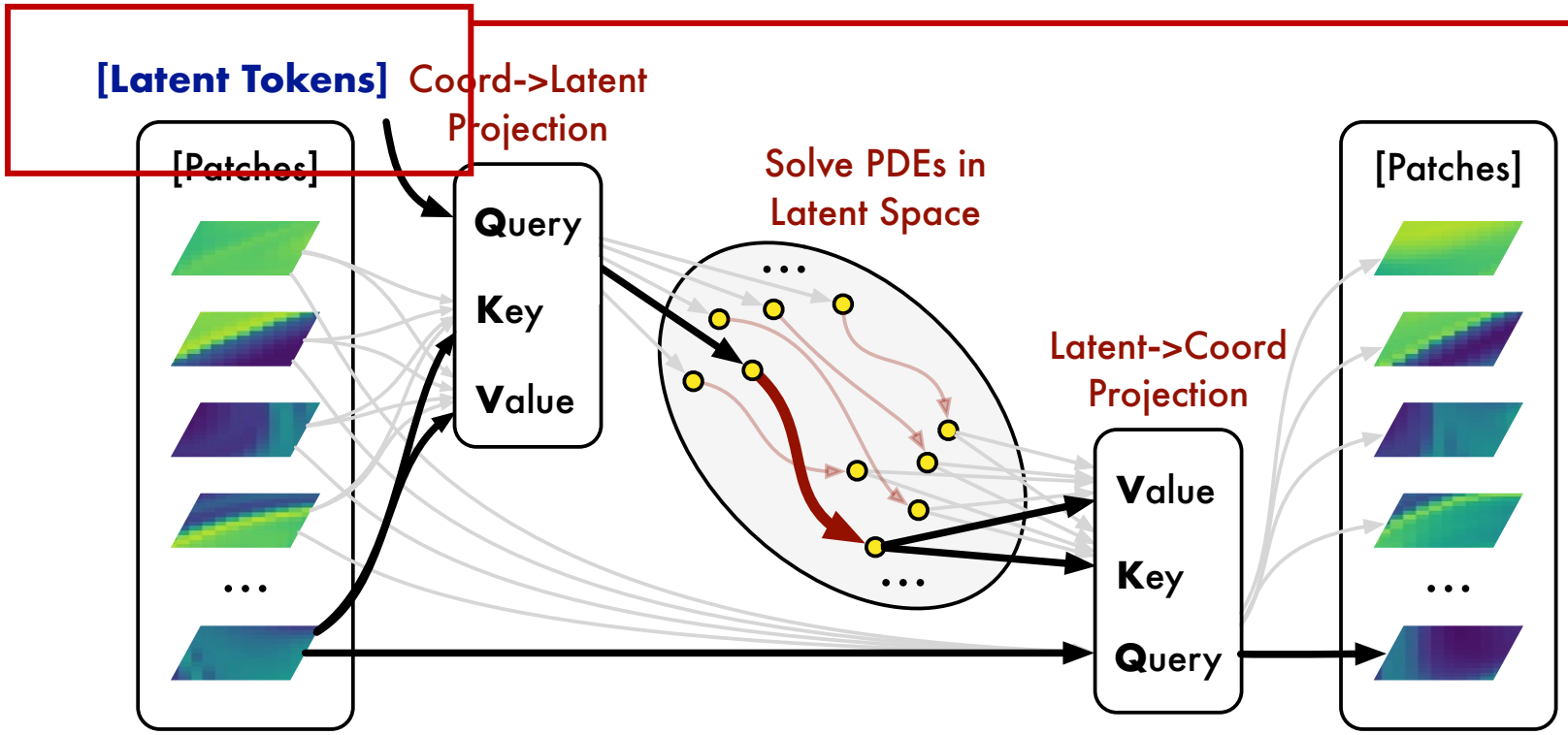


Still in the coordinate space

Undergoing the problems from high-dimensional PDEs

② ***Attention-based projector*** → *Remove unwieldy coordinate information*

Hierarchical Projection Network

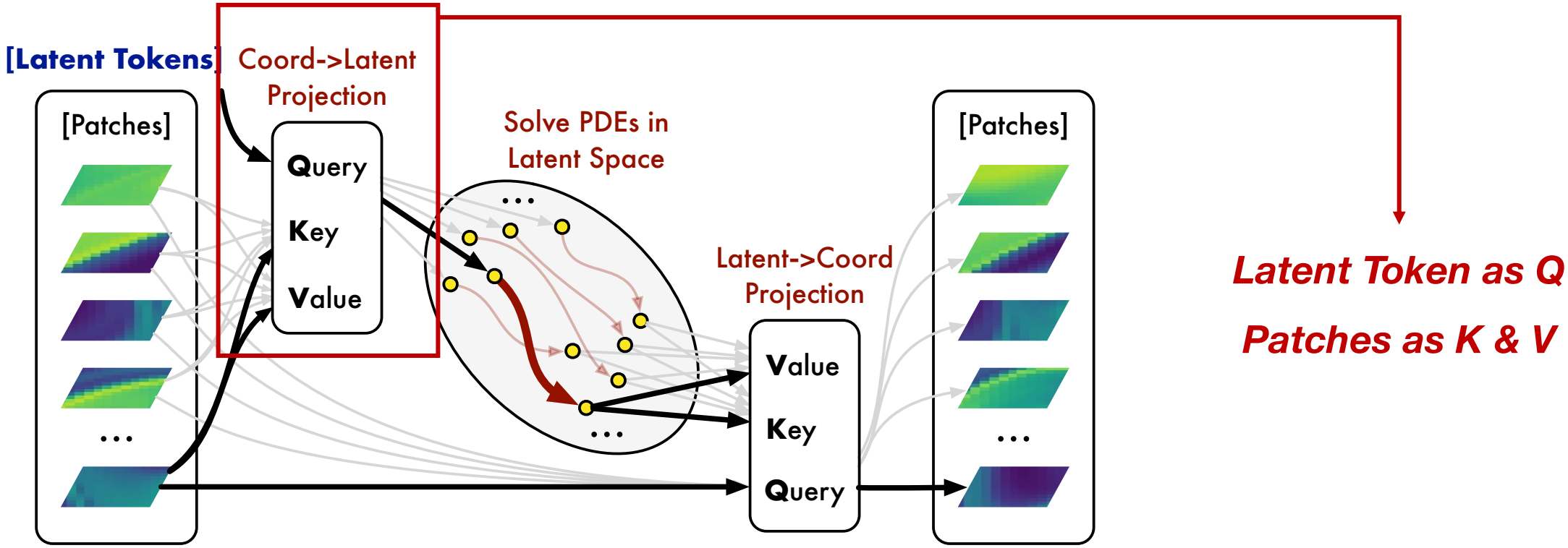


Initialize Latent tokens as model parameters

- *Share in all samples*
- *Provide physical prompt for projection*
- *Efficient computation*

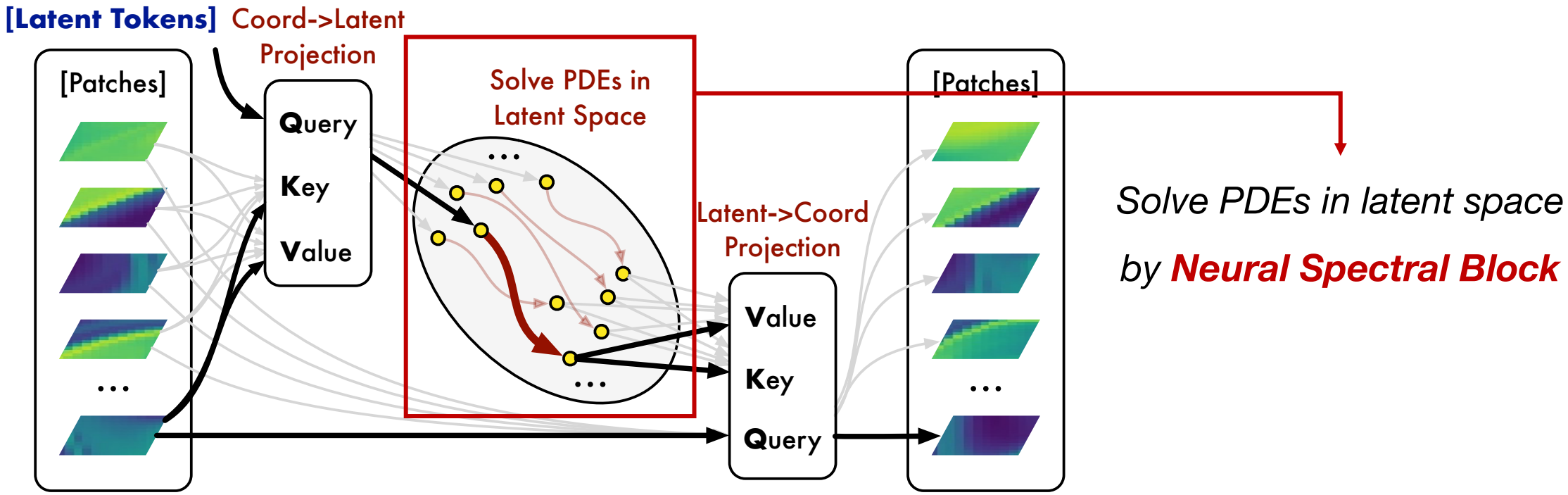
② ***Attention-based projector*** → *Remove unwieldy coordinate information*

Hierarchical Projection Network



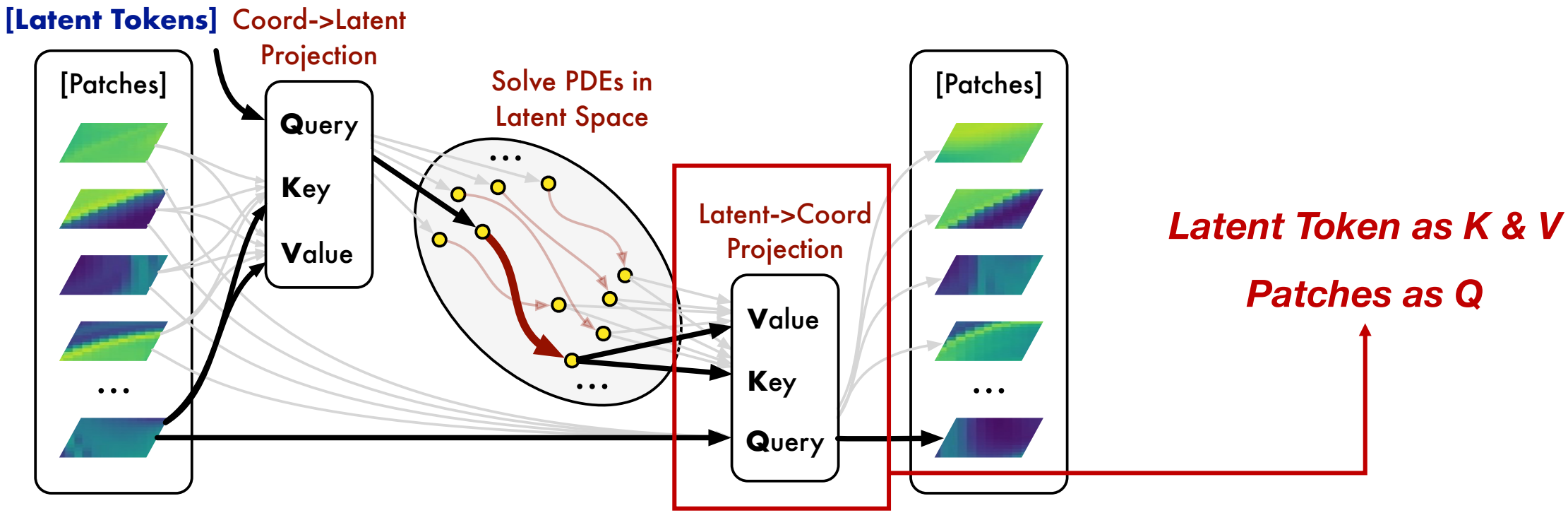
$$\mathbf{T}_{\boldsymbol{x},i} = \mathbf{T}_i + \sum_{\mathbf{s}\in\mathcal{D}} \frac{\mathrm{Sim}\left(\mathbf{T}_i, \boldsymbol{x}(\mathbf{s})\mathbf{W}_\mathrm{K}\right)}{\sum_{\mathbf{s}'\in\mathcal{D}} \mathrm{Sim}\left(\mathbf{T}_i, \boldsymbol{x}(\mathbf{s}')\mathbf{W}_\mathrm{K}\right)} \left(\boldsymbol{x}(\mathbf{s})\mathbf{W}_\mathrm{V}\right)$$

Hierarchical Projection Network



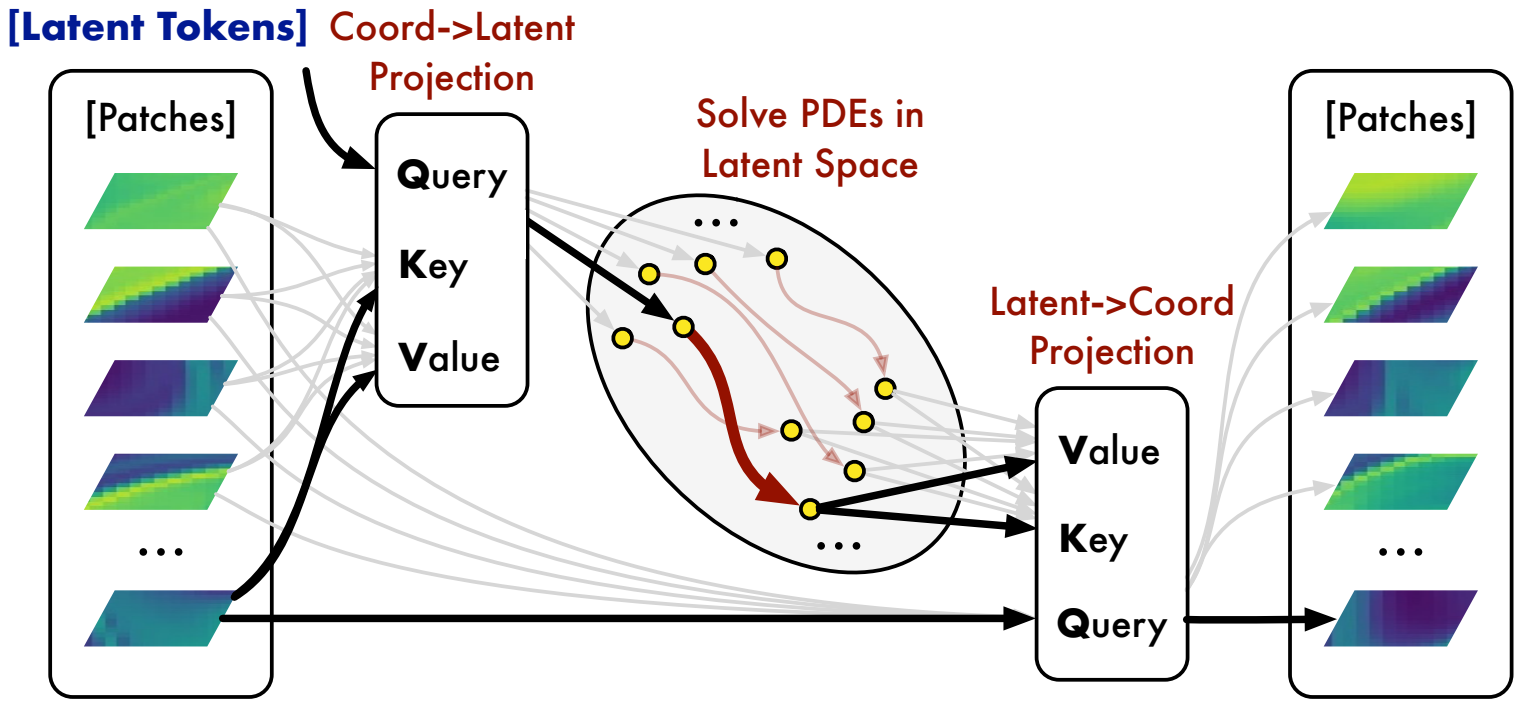
$$\{\mathbf{T}^k_{\boldsymbol{y},i,j}\}_{i=1}^{C} = \mathrm{Solve}\left(\{\mathbf{T}^k_{\boldsymbol{x},i,j}\}_{i=1}^{C}\right)$$

Hierarchical Projection Network



$$\widehat{\boldsymbol{y}}(\mathbf{s}) = \boldsymbol{x}(\mathbf{s}) + \sum_{i=1}^{C} \frac{\mathrm{Sim}\left(\boldsymbol{x}(\mathbf{s}), \mathbf{T}_{\boldsymbol{y},i}\mathbf{W}'_{\mathrm{K}}\right)}{\sum_{i'=1}^{C} \mathrm{Sim}\left(\boldsymbol{x}(\mathbf{s}), \mathbf{T}_{\boldsymbol{y},i'}\mathbf{W}'_{\mathrm{K}}\right)} \left(\mathbf{T}_{\boldsymbol{y},i}\mathbf{W}'_{\mathrm{V}}\right)$$

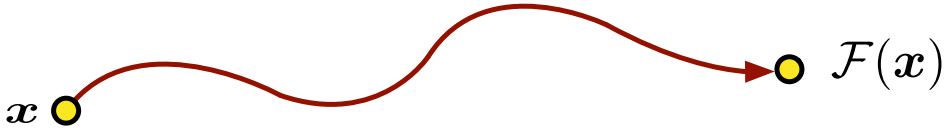
Hierarchical Projection Network



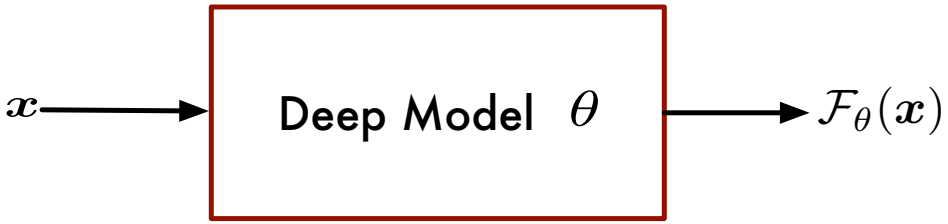
1. ***Linear complexity projection, more efficient computation***
2. ***Highlight the inherent properties of high-dimensional data***
3. ***Benefit the model convergence properties***

Neural Spectral Block

Task: Approximate Complex Nonlinear Mapping

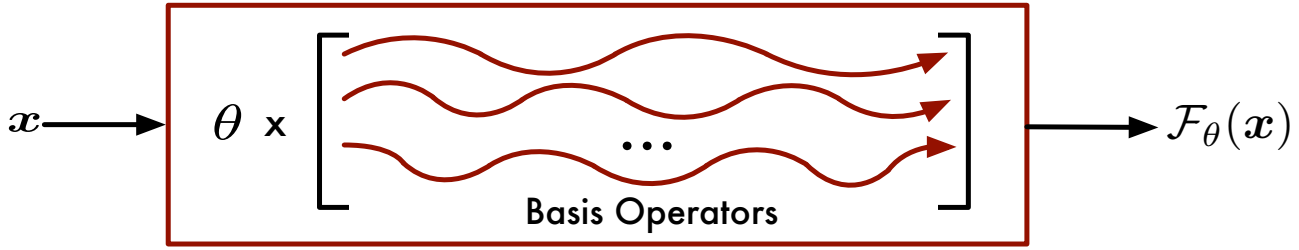


(a) U-Net: Directly Learn Mapping



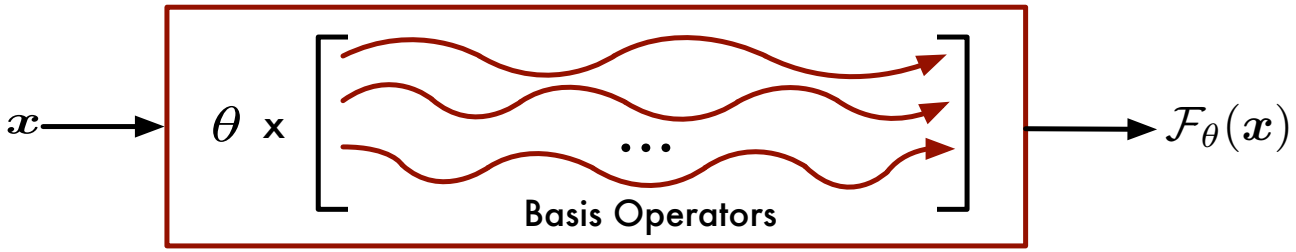
(b) FNO: Linear Transformation in Fourier Domain

(c) LSM: Decompose into Basis Operators



LSM approximates complex mappings by **learning multiple basis operators**

Neural Spectral Block

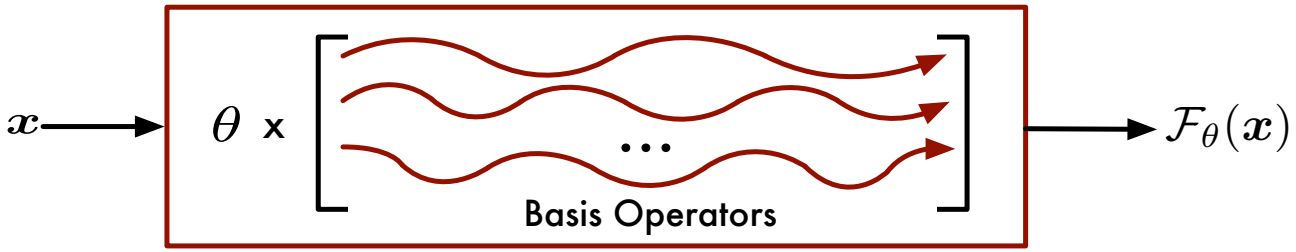


$$\mathcal{F}_{\theta_{\text{Solve}}} = \sum_{i=1}^{N} w_i \mathcal{F}_{\theta_{\text{Solve},i}}$$

We select trigonometric basis operators

$$\mathcal{F}_{\theta_{\text{Solve}},(2k-1)}\left(\boldsymbol{t}_{\boldsymbol{x}}(\mathbf{s})\right) = \sin\left(k\boldsymbol{t}_{\boldsymbol{x}}(\mathbf{s})\right)$$
$$\mathcal{F}_{\theta_{\text{Solve}},(2k)}\left(\boldsymbol{t}_{\boldsymbol{x}}(\mathbf{s})\right) = \cos\left(k\boldsymbol{t}_{\boldsymbol{x}}(\mathbf{s})\right)$$

Neural Spectral Block



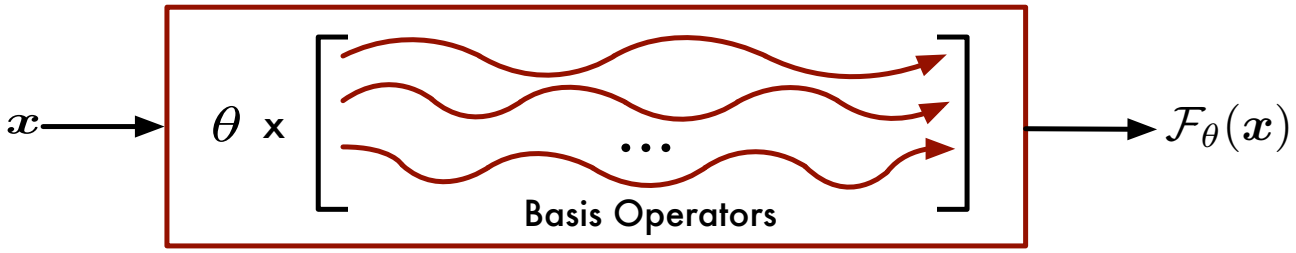
$$\mathcal{F}_{\theta_{\text{Solve}}} = \sum_{i=1}^{N} w_i \mathcal{F}_{\theta_{\text{Solve},i}}$$

$\{\omega_1, \cdots, \omega_N\}$ **are model parameters**

Will be optimized to satisfy the PDEs better,

during the training process, namely solving PDEs

Neural Spectral Block



$$\mathcal{F}_{\theta_{\text{Solve}}} = \sum_{i=1}^{N} w_i \mathcal{F}_{\theta_{\text{Solve},i}}$$

Neural spectral block is applied to

projected latent tokens of **all the patches in multiple scales**

$$\mathbf{T}_{\boldsymbol{y}} = \mathbf{T}_{\boldsymbol{x}} + \mathbf{w}_0 + \mathbf{w}_{\sin} \begin{bmatrix} \sin(\mathbf{T}_{\boldsymbol{x}}) \\ \vdots \\ \sin(\frac{N}{2}\mathbf{T}_{\boldsymbol{x}}) \end{bmatrix} + \mathbf{w}_{\cos} \begin{bmatrix} \cos(\mathbf{T}_{\boldsymbol{x}}) \\ \vdots \\ \cos(\frac{N}{2}\mathbf{T}_{\boldsymbol{x}}) \end{bmatrix}$$

Theoretical analysis

Convergence of Trigonometric Approximation in High-dimensional Space:

Let $\boldsymbol{f}: \mathbb{R}^M \to \mathbb{R}^M$ *be a* 2π*-periodic function w.r.t. the variable on each dimension, where* $\boldsymbol{f} \in L_p([-\pi, \pi)^M), M \geq 2, 1 \leq p \leq \infty$ *and* $p \neq 2$*. For* $\boldsymbol{f}$ *defined on the* ***M-dimension space****, its trigonometric approximation* $\boldsymbol{f}^N$ *is defined as:*

$$\boldsymbol{f}^N(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^M, |\mathbf{k}| \leq N} \left(\frac{1}{2\pi} \int_{[-\pi,\pi)^M} \boldsymbol{f}(\mathbf{t}) e^{-i\mathbf{k}\mathbf{t}} \mathrm{d}\mathbf{t} \right) e^{i\mathbf{k}\mathbf{x}},$$

if $\boldsymbol{f}$ *satisfies the Lipschitz condition, then there exists a constant* K*, such that*

$$\left|\left|\boldsymbol{f} - \boldsymbol{f}^N\right|\right| \leq K N^{(M-1)\left|\frac{1}{2} - \frac{1}{p}\right| - 1}.$$

Slow Convergence Rate in High-dimensional Space

Theoretical analysis

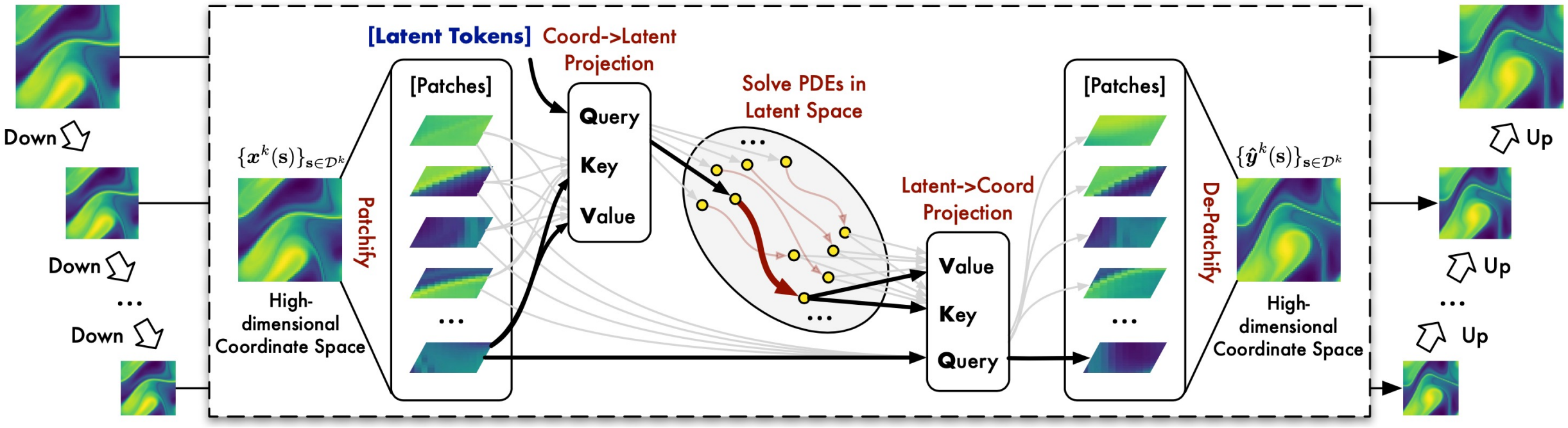
Approximation and Convergence Properties of Neural Spectral Block

(trigonometric approximation with residual): Given $f:[0,\pi]\to\mathbb{R}$*, if* f *satisfies the Lipschitz condition, there is a choice of model parameters such that the approximation* f^N *defined in neural spectral block can uniformly converge to f with the speed as follows:*

$$\left||f-f^N(\boldsymbol{x})|\right|\leq K\frac{\ln N}{N},\forall x\in[0,\pi].$$

Projecting M-dimension data into **independent latent tokens** brings favorable convergence speed of Neural Spectral Block.

Overall design of LSM



LSM with *Hierarchical Projection Network* and *Neural Spectral Block*

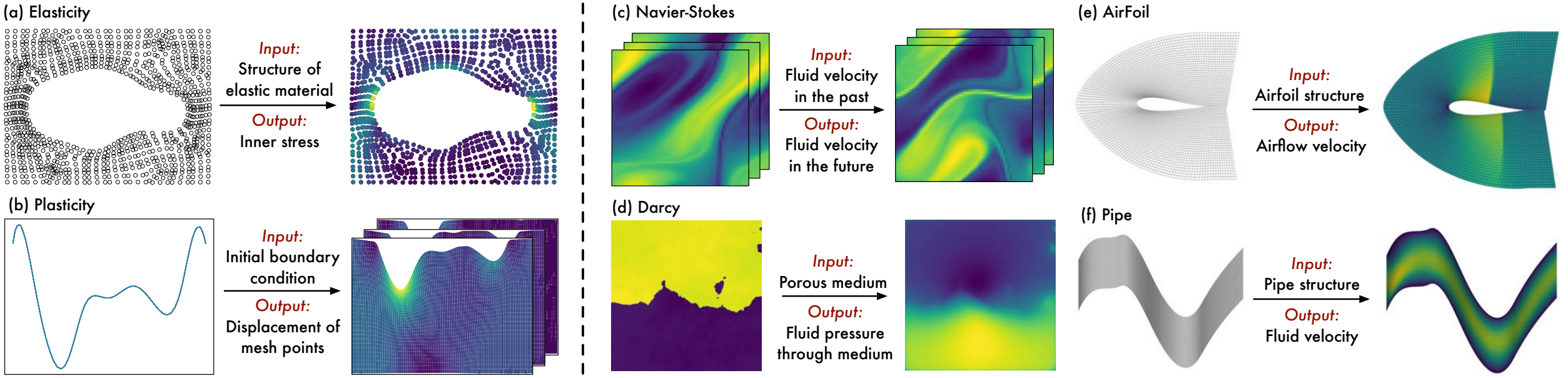
① Coor → Latent ② Solving in the Latent Space ③ Latent → Coor

Experiments

Physics	Benchmarks	Geometry	#Dim
Solid	Elasticity-P	Point Cloud	2D
	Elasticity-G	Regular Grid	2D
	Plasticity	Structured Mesh	3D
Fluid	Navier–Stokes	Regular Grid	3D
	Darcy	Regular Grid	2D
	Airfoil	Structured mesh	2D
	Pipe	Structured mesh	2D

Seven typical PDE solving tasks, covering both fluid and solid physis, various geometrics.

PDE-governed Tasks



Approximate complex input-output mappings with deep models

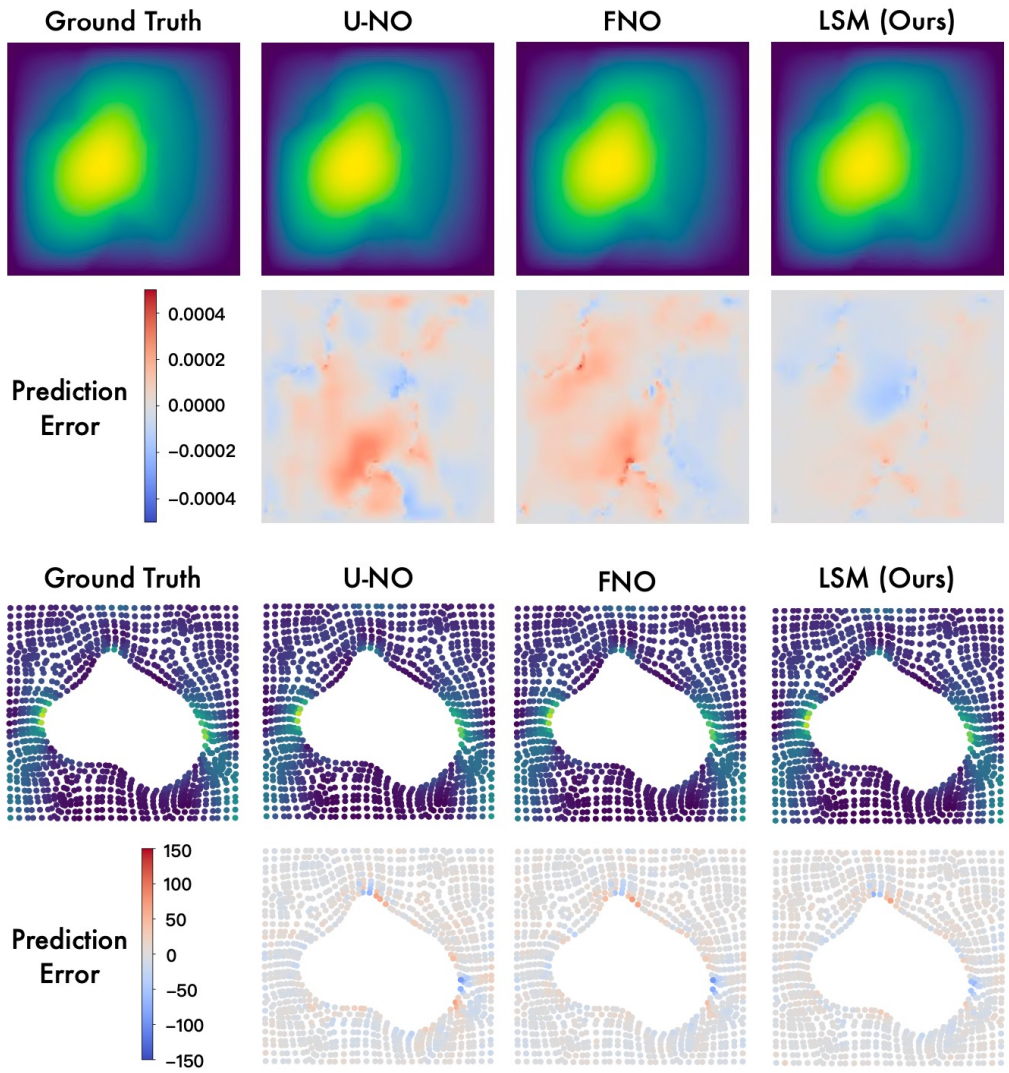
Main Results

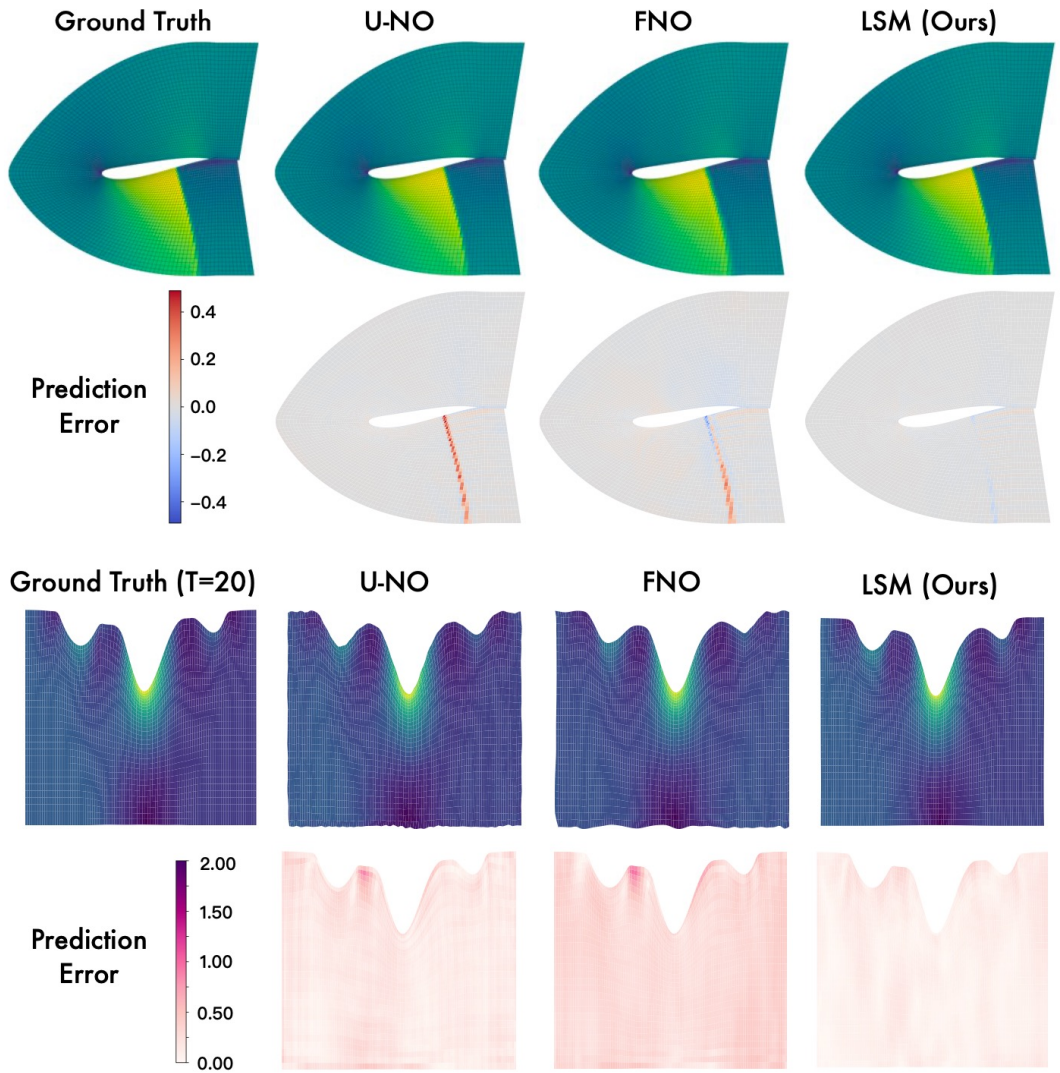
Model	Solid Physics*			Fluid Physics†			
	Elasticity-P ‡	Elasticity-G	Plasticity	Navier–Stokes	Darcy	Airfoil	Pipe
U-Net (2015)	0.0235	0.0531	0.0051	0.1982	0.0080	0.0079	0.0065
ResNet (2016)	0.0262	0.0843	0.0233	0.2753	0.0587	0.0391	0.0120
TF-Net (2019)	/	/	/	0.1801	/	/	/
Swin (2021)	0.0283	0.0819	0.0170	0.2248	0.0397	0.0270	0.0109
DeepONet (2021)	0.0965	0.0900	0.0135	0.2972	0.0588	0.0385	0.0097
FNO (2021)	0.0229	0.0508	0.0074	0.1556	0.0108	0.0138	0.0067
U-FNO (2021)	0.0239	0.0480	0.0039	0.2231	0.0183	0.0269	0.0056
WMT (2021)	0.0359	0.0520	0.0076	0.1541	0.0082	0.0075	0.0077
Galerkin (2021)	0.0240	0.1681	0.0120	0.2684	0.0170	0.0118	0.0098
SNO (2022)	0.0390	0.0987	0.0070	0.2568	0.0495	0.0893	0.0294
U-NO (2022)	0.0258	0.0469	0.0034	0.1713	0.0113	0.0078	0.0100
HT-Net (2022)	0.0372	0.0472	0.0333	0.1847	0.0079	0.0065	0.0059
F-FNO (2023)	0.0263	0.0475	0.0047	0.2322	0.0077	0.0078	0.0070
KNO (2023a)	/	/	/	0.2023	/	/	/
LSM	**0.0218**	**0.0408**	**0.0025**	**0.1535**	**0.0065**	**0.0059**	**0.0050**
Promotion	4.8%	13.0%	26.5%	0.4%	15.6%	9.2%	10.7%

LSM achieves consistent SOTA and surpasses previous

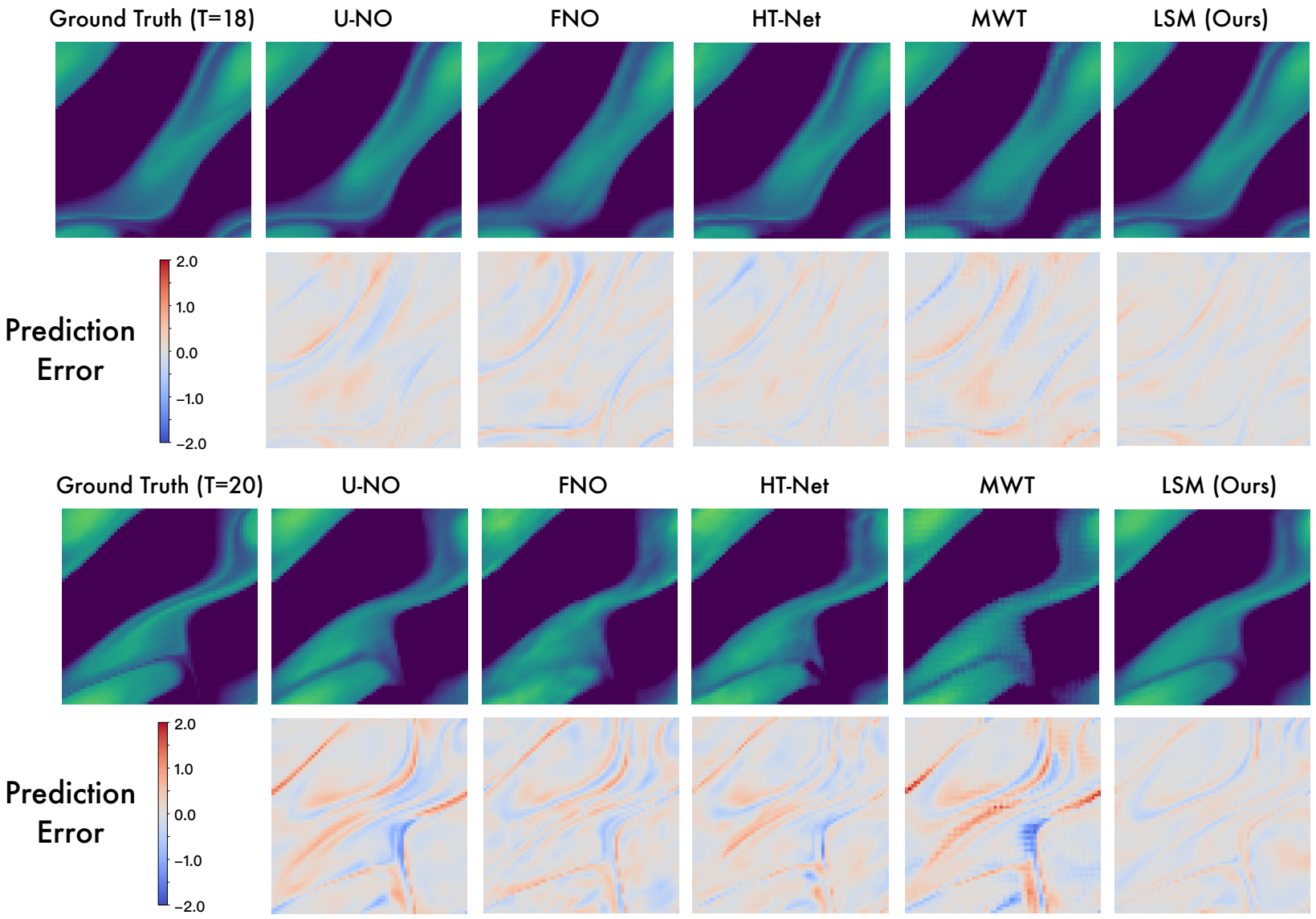
14 baselines with 11.5% error reduction.

Showcases

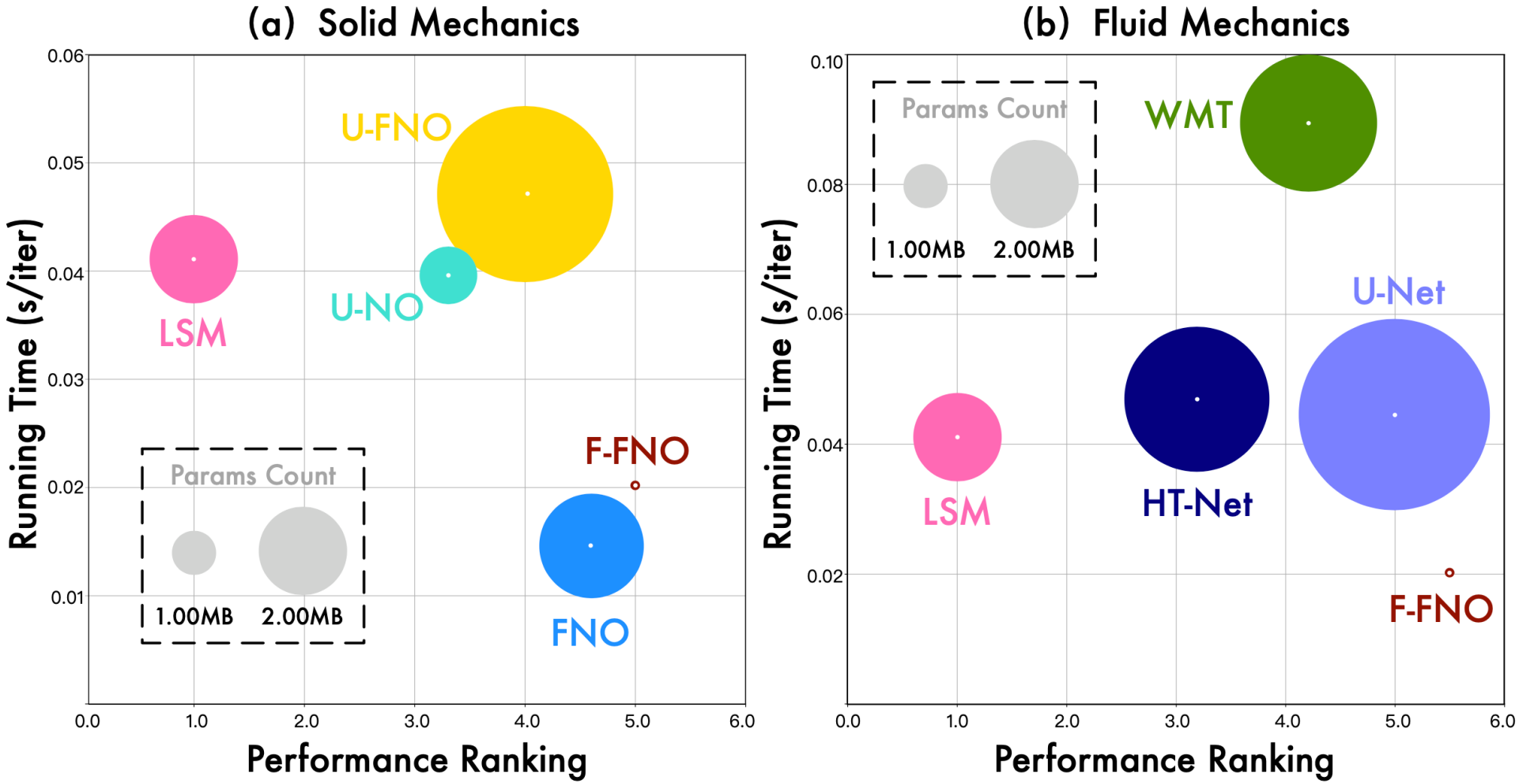

Ground Truth
U-NO
FNO
LSM (Ours)
Prediction Error
0.0004
0.0002
0.0000
−0.0002
−0.0004
Ground Truth
U-NO
FNO
LSM (Ours)
Prediction Error
150
100
50
0
−50
−100
−150


Ground Truth
U-NO
FNO
LSM (Ours)
Prediction Error
0.4
0.2
0.0
−0.2
−0.4
Ground Truth (T=20)
U-NO
FNO
LSM (Ours)
Prediction Error
2.00
1.50
1.00
0.50
0.00

Showcases

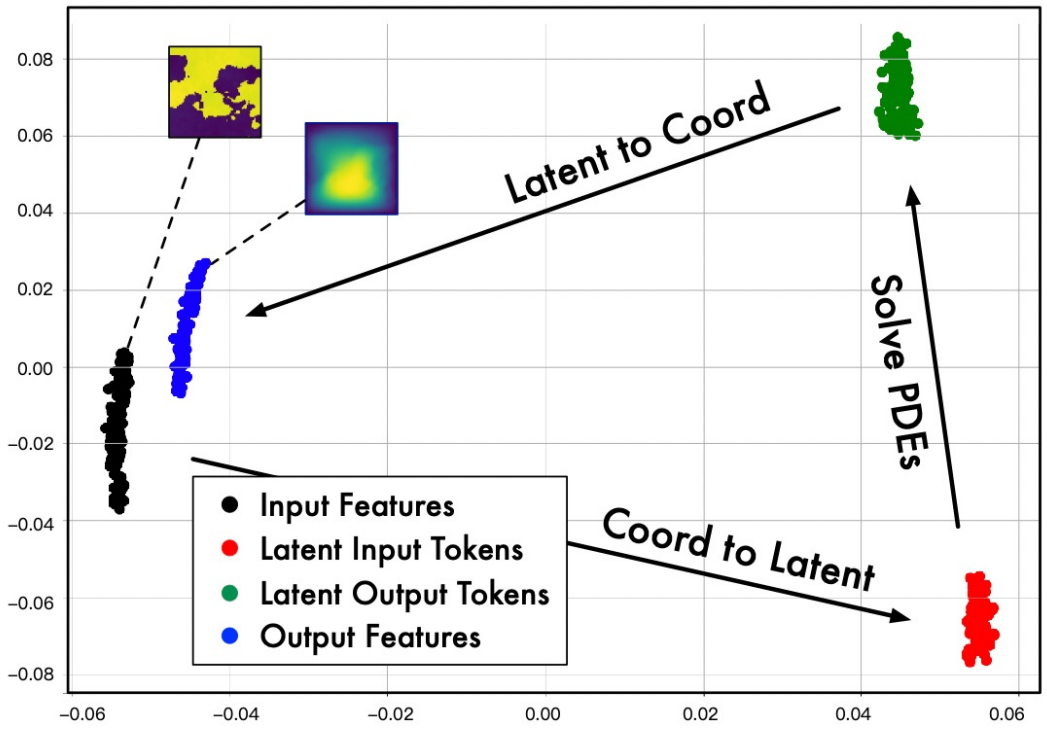

Ground Truth (T=18)
U-NO
FNO
HT-Net
MWT
LSM (Ours)
Prediction Error
2.0
1.0
0.0
−1.0
−2.0
Ground Truth (T=20)
U-NO
FNO
HT-Net
MWT
LSM (Ours)
Prediction Error
2.0
1.0
0.0
−1.0
−2.0

Efficiency

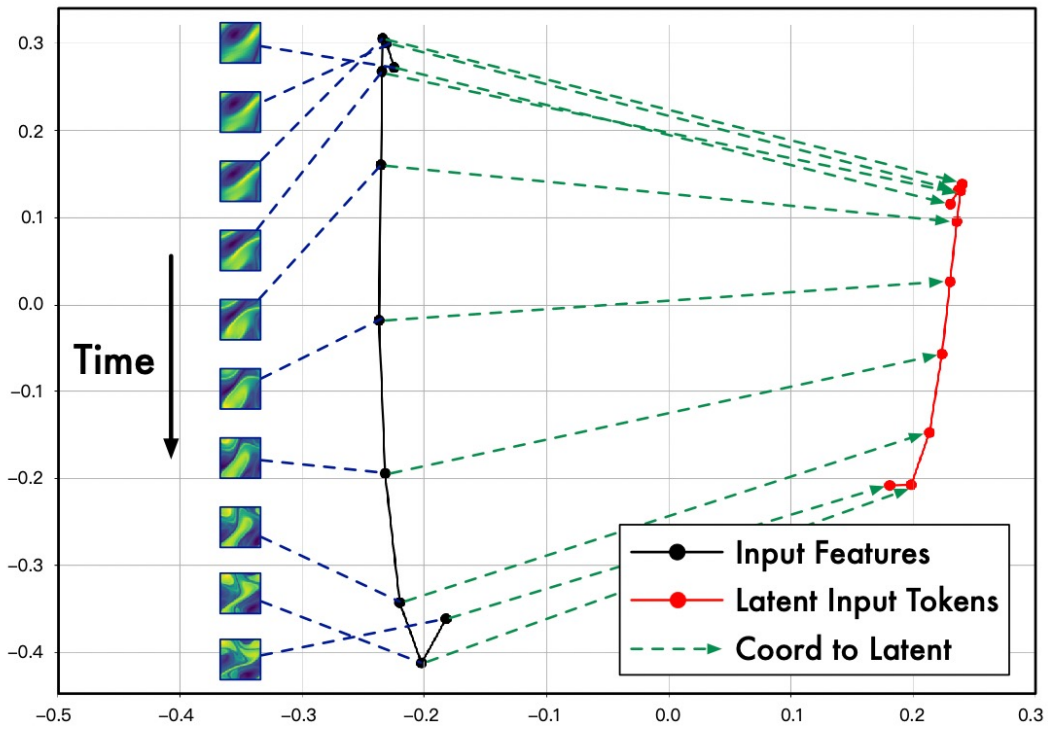


Favorable trade-off between performance and efficiency.

Solving Process Visualization



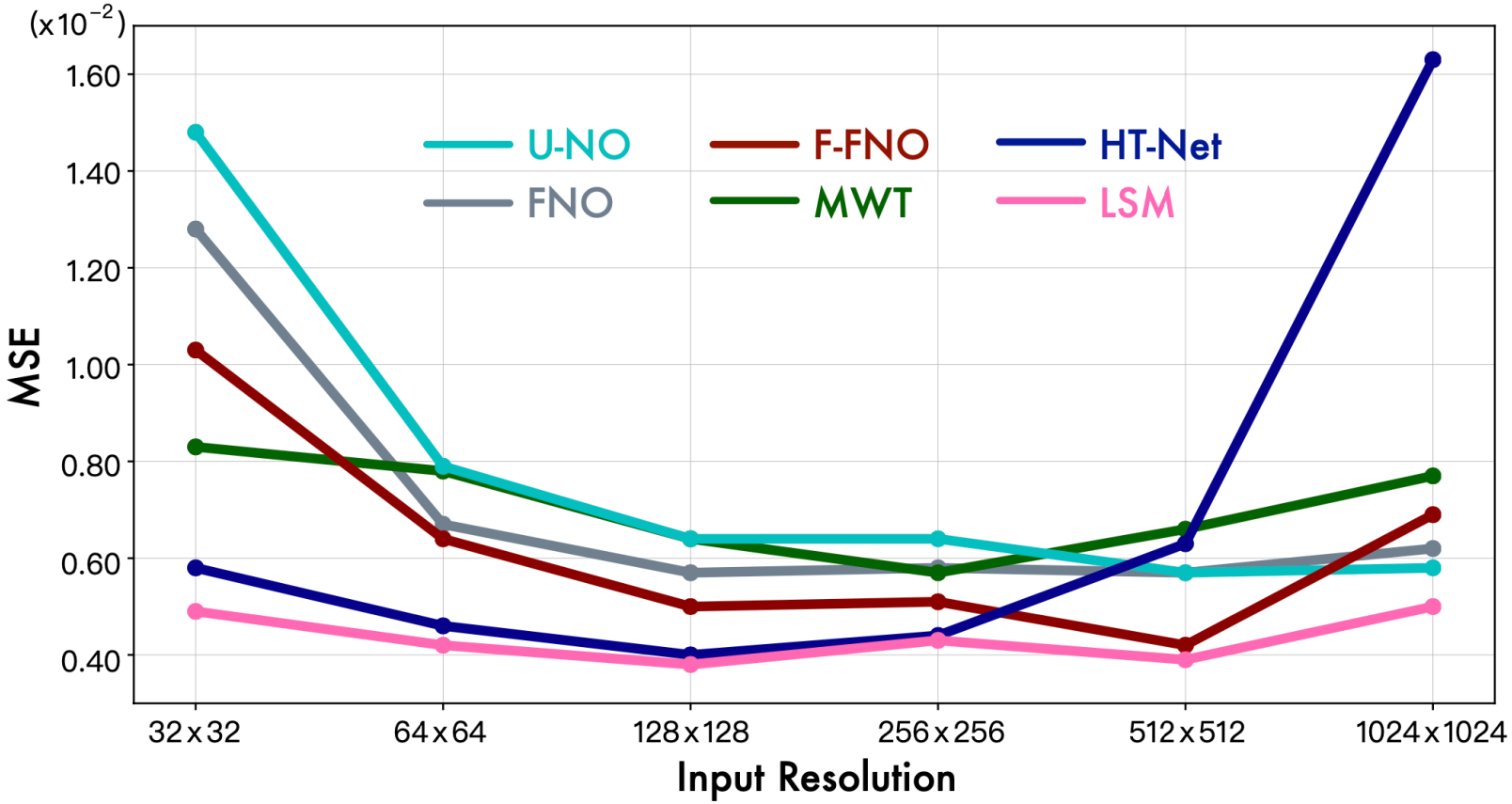
(a) Darcy Benchmark



(b) Latent Process on Navier–Stokes

LSM can precisely capture the complex mapping and latent process from high-dimensional coordinate space.

Performance under various resolutions in Darcy benchmark

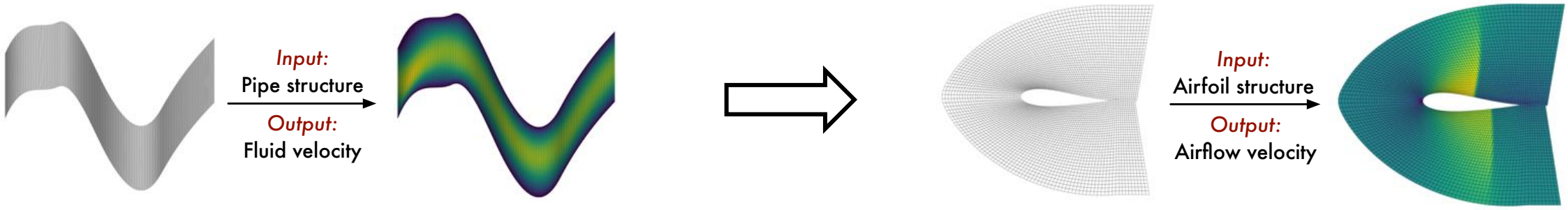


LSM presents a stable performance w.r.t. different inputs and consistently surpasses other baselines.

Transferability

Finetune the pipe-pretrained model into airfoil with limited data

(same PDE equation but different boundary conditions)



MSE ($\times 10^{-2}$)	20% Airfoil Data	40% Airfoil Data	60% Airfoil Data	80% Airfoil Data	100% Airfoil Data
U-Net (2015)	1.88→1.93 (-2.7%)	1.38→1.14 (+17.3%)	0.96→0.90 (+6.3%)	0.85→0.81 (+4.7%)	0.79→0.77 (+2.5%)
U-NO (2022)	6.30→1.72	2.39→1.73	1.10→1.00 (+9.1%)	0.86→0.82 (+4.7%)	0.78→0.82 (-5.1%)
HT-Net (2022)	1.73→1.43 (+17.3%)	1.08→0.82 (**+24.1%**)	0.75→0.69 (+8.0%)	0.70→0.65 (+7.1%)	0.65→0.61 (+6.2%)
LSM	**1.66→1.31 (+21.1%)**	**0.91→0.75** (+17.6%)	**0.69→0.61 (+11.6%)**	**0.63→0.58 (+7.9%)**	**0.59→0.55 (+6.8%)**

LSM shows good Transferability between different conditions.

Leaderboard for solving PDEs

Input Size ($S \times S$)		Parameter	GPU Memory	Running Time	Ranking			
Model	S	(MB)	(MB)	(s / iter)	Seven tasks*	Solid*	Fluid†	Averaged‡
LSM (ours)	64	2.002	1409	0.0353	(1, 1, 1, 1, 1, 1, 1)	1.0	1.0	1.0
	128	2.002	1679	0.0359				
	256	2.002	1959	0.0411				
	512	2.002	3019	0.0602				
	1024	2.002	7859	0.2002				
U-NO	64	1.307	1345	0.0347	(6, 2, 2, 4, 7, 4, 9)	3.3	8.0	4.9
	128	1.307	1381	0.0354				
	256	1.307	1603	0.0397				
	512	1.307	2473	0.0989				
	1024	1.307	6833	0.3335				
U-Net	64	4.332	1171	0.0321	(3, 8, 5, 6, 4, 6, 4)	5.3	5.0	5.1
	128	4.332	1243	0.0307				
	256	4.332	1515	0.0450				
	512	4.332	2429	0.1589				
	1024	4.332	6235	0.8100				
FNO	64	2.368	1137	0.0202	(2, 6, 6, 3, 6, 8, 5)	4.6	7.3	5.1
	128	2.368	1179	0.0203				
	256	2.368	1349	0.0147				
	512	2.368	1975	0.0401				
	1024	2.368	4591	0.1270				
HT-Net	64	3.285	1175	0.0406	(10, 3, 10, 5, 3, 2, 3)	7.6	3.2	5.1
	128	3.285	1283	0.0415				
	256	3.285	1749	0.0469				
	512	3.285	3267	0.1300				
	1024	3.285	9259	0.4581				

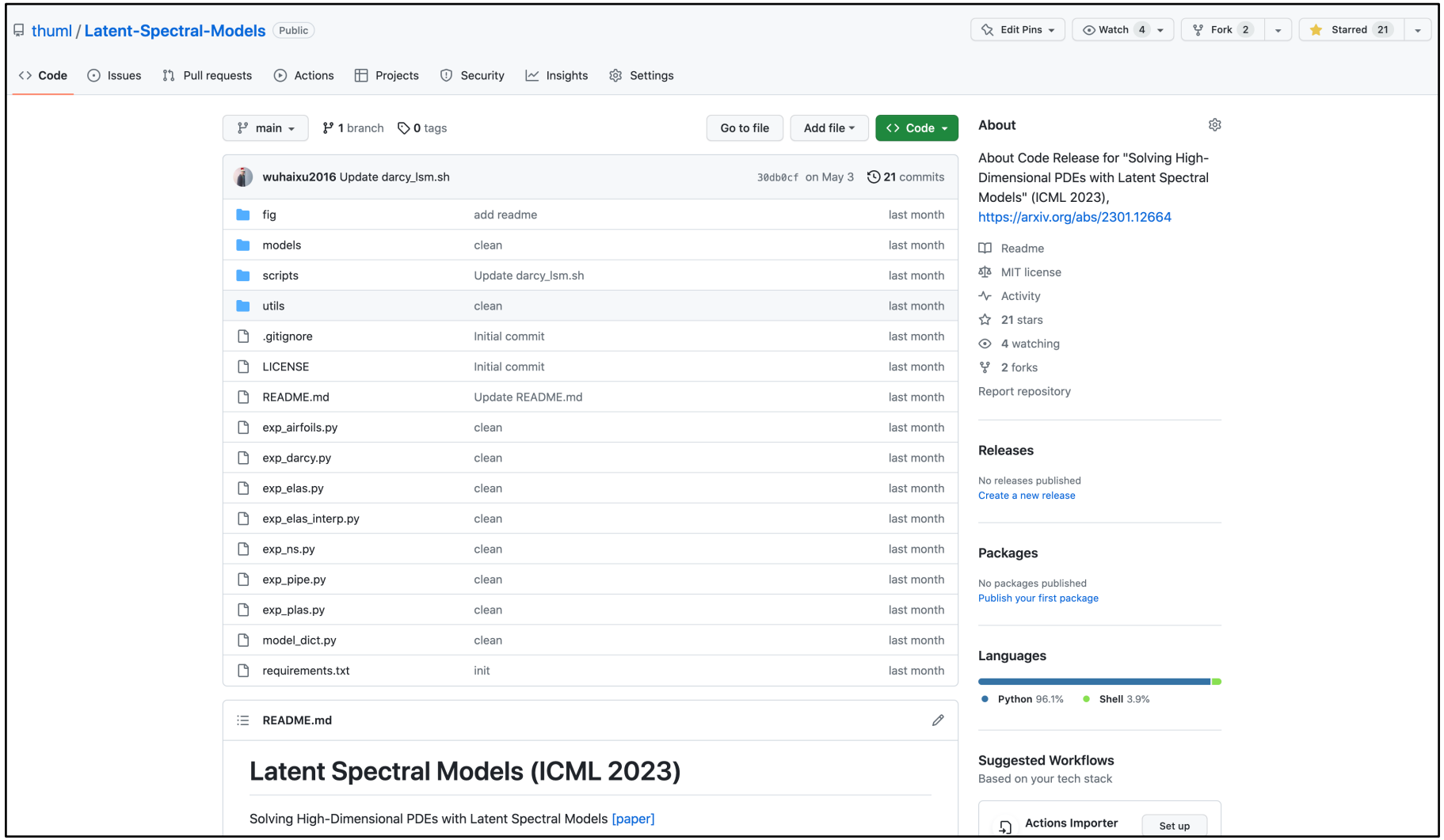
Top 5 in fluid physics

LSM (ours), HT-Net (2022), WMT (2021), U-Net (2015), F-FNO (2023)

Top 5 in solid physics

LSM (ours), U-NO (2022), U-FNO (2021), FNO (2021), F-FNO (2023)

Open Source



Code is available at https://github.com/thuml/Latent-Spectral-Models

Thank You!

whx20@mails.tsinghua.edu.cn