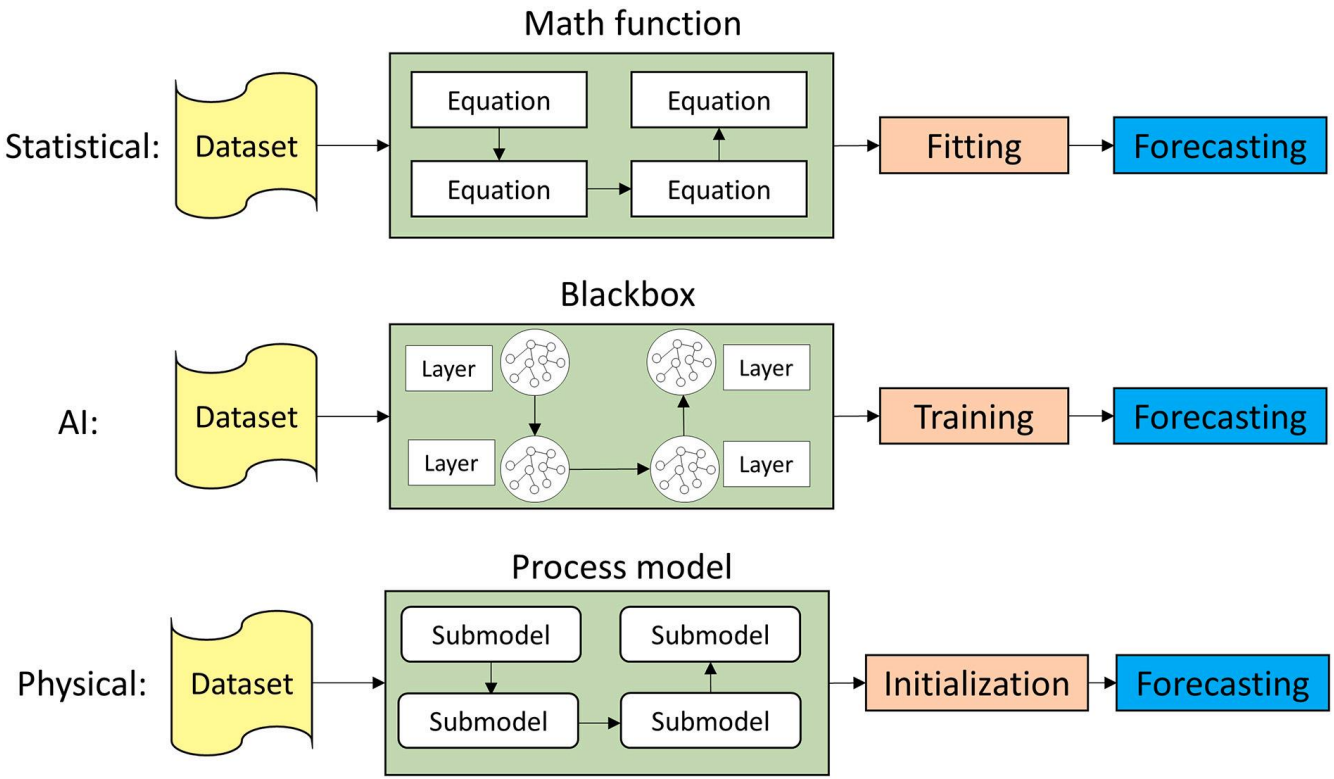
ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Physics-Coupled Spatio-Temporal Active Learning for Dynamical Systems

Yu Huang, Yufei Tang, Xingquan Zhu, Min Shi, Ali Muhamed Ali, Hanqi Zhuang, Laurent Cherubin
{yhwang2018,tangy,xzhu3,mshi2018,amuhamedali2014,zhuang,lcherubin}@fau.edu
Florida Atlantic University
Boca Raton, Florida, USA

Spatio-Temporal forecasting(STF)

extends traditional time series forecasting or spatial interpolation problem to space and time dimensions.



GWR, ST-kriging

nonlinear high-level features

lack of interpretability

Imperfect model structure

Challenges

- high cost of data acquisition

ocean data:

- need trained scientists
- cost limits the total number of sensors
- increasing costs does not necessarily mean improved model predictions

where and when to query the data within constraints

AL

Challenges

- infer the underlying causes
  - how data is generated
  - how data is propagated

homogeneity and heterogeneity

driven by the physical laws

impacted by the spatial and temporal regions

- propose an active learning algorithm for spatial-temporal dynamical systems.
- propose ST-PCNN, which captures heterogeneity from all spatial locations.
- propose embedding a physics learning module to learn the inherited homogeneity.
- conduct extensive experiments and comparative studies on both synthetic and real-world datasets.

Procedure

while *Termination Conditions NOT satisfied* **do**

Learn Physics: $[\boldsymbol{\lambda}] \leftarrow$ learn the physics, *i.e.* coefficients $\boldsymbol{\lambda}$ of PDE, from the existing training data $\mathbb{D}$ by minimizing Eq. (32);

ST-PCNN Training: train the ST-PCNN with the learned physics λ from training data $\mathbb{D}$, see Algorithm 2;

Prediction: $\left[\hat{\mathcal{S}}\right] \leftarrow$ make prediction at all locations in the network, $\forall \boldsymbol{x} \in \Omega$;

Kriging: $\Omega^n_{kriging} \leftarrow$ use Kriging to identify n query locations with the largest estimated errors for next active learning step;

Data Query: obtain consecutive observations of window size T_w from the newly selected locations $\mathbb{D}_{kriging} = \{(\boldsymbol{x}, t, \mathcal{S}) | \boldsymbol{x} \in \Omega^n_{kriging}, t \in T_w, \mathcal{S} \in \mathbb{R}\}$;

Update Dataset: add $\mathbb{D}_{kriging}$ to the existing training data $\mathbb{D}$;

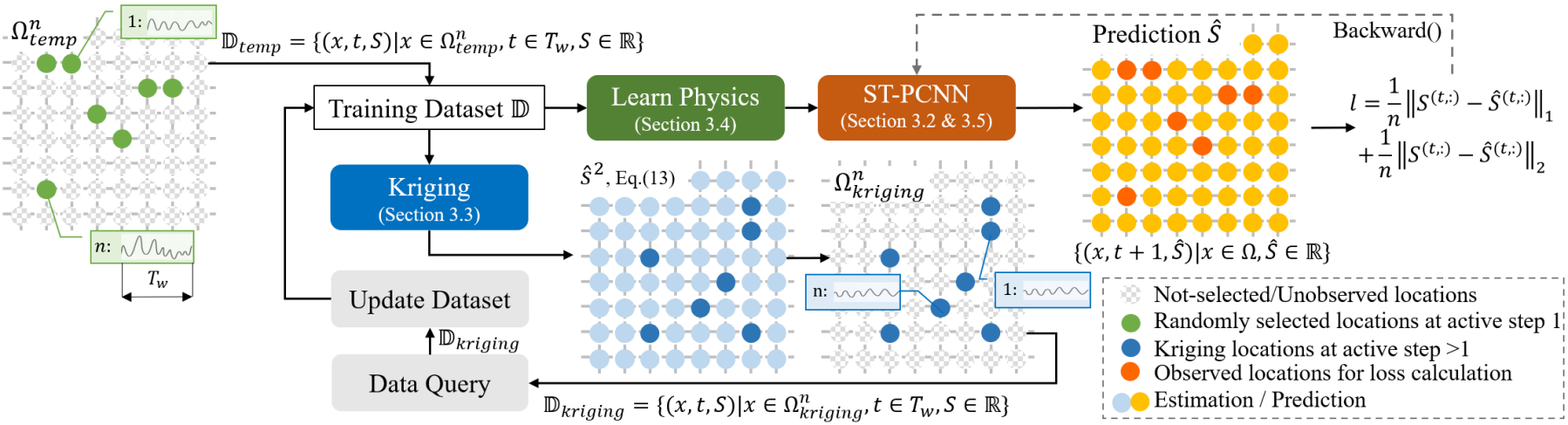
end

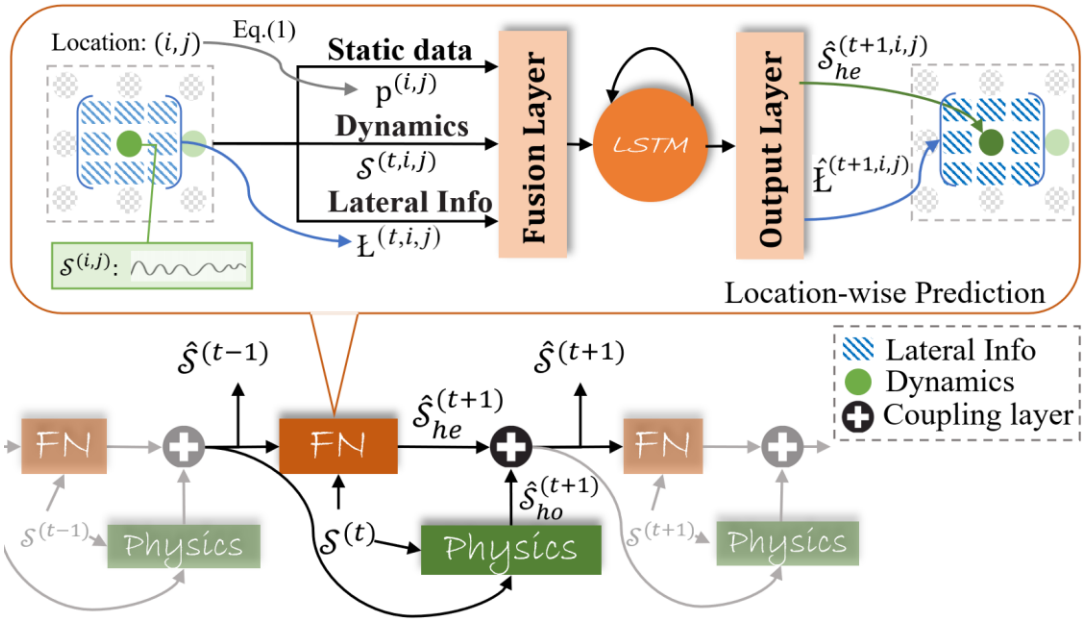
→ GP

→ Pos + LSTM

→ Kriging(GPR)

Framework


Ω^n_{temp}
1:
n:
T_w
$\mathbb{D}_{temp} = \{(x,t,S)|x \in \Omega^n_{temp}, t \in T_w, S \in \mathbb{R}\}$
Training Dataset $\mathbb{D}$
Learn Physics
(Section 3.4)
ST-PCNN
(Section 3.2 & 3.5)
Prediction $\hat{S}$
Backward()
$l = \frac{1}{n}\|S^{(t,:)} - \hat{S}^{(t,:)}\|_1 + \frac{1}{n}\|S^{(t,:)} - \hat{S}^{(t,:)}\|_2$
Kriging
(Section 3.3)
$\hat{S}^2$, Eq.(13)
$\Omega^n_{kriging}$
$\{(x,t+1,\hat{S})|x \in \Omega, \hat{S} \in \mathbb{R}\}$
Update Dataset
$\mathbb{D}_{kriging}$
Data Query
$\mathbb{D}_{kriging} = \{(x,t,S)|x \in \Omega^n_{kriging}, t \in T_w, S \in \mathbb{R}\}$
Not-selected/Unobserved locations
Randomly selected locations at active step 1
Kriging locations at active step >1
Observed locations for loss calculation
Estimation / Prediction

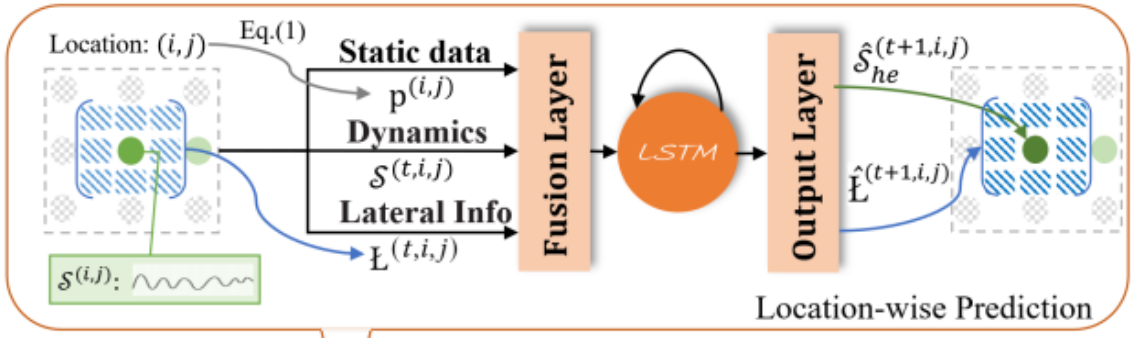


bi-network architecture:

- forecasting network (FN): produces heterogeneous prediction leveraging its own specific local attributes only
- physics network (PN): generates homogeneous solution of the dynamics regularized by the overall underlain governing physics

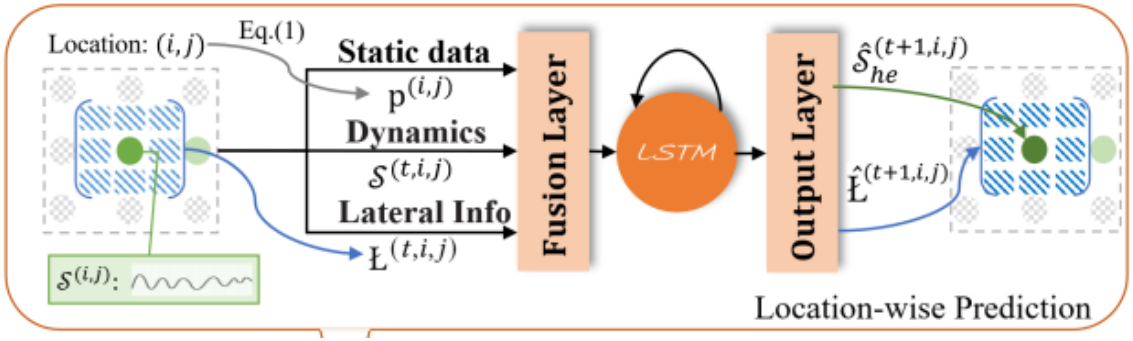
produce the final prediction by synthesizing heterogeneous prediction and homogeneous prediction

$$\hat{\mathcal{S}}^{(t+1,i,j)} = Relu([\hat{\mathcal{S}}_{he}^{(t+1,i,j)}, \hat{\mathcal{S}}_{ho}^{(t+1,i,j)}]W_C^T + b_C)$$



Static data: representing node location
Dynamics: representing dynamics of node (temperature, conductivity and velocity)
Lateral info: capturing interaction between each node and its neighbors

Predicted dynamics
Additional lateral information



FN encodes each view using a fusion layer

features are then fed into an LSTM to model the node-specific interactions

An output layer is stacked to transform the LSTM output into the expected dynamic prediction and additional lateral information

$$\mathbf{p}^{(i,j)} := \begin{cases} \sin(\omega_k, i), \sin(\omega_k, j) & \text{if } i, j = 2k \\ \cos(\omega_k, i), \cos(\omega_k, j) & \text{if } i, j = 2k+1 \end{cases} \qquad \omega_k = \frac{1}{10{,}000^{2k/d}}, k \in \mathbb{N}_{\leq \lceil \frac{d}{2} \rceil}$$

Positional Encoding

Why?

- Position and order of words define the grammar and actual semantics of a sentence.
- Transformer ditched the recurrence mechanism, word order information is lost

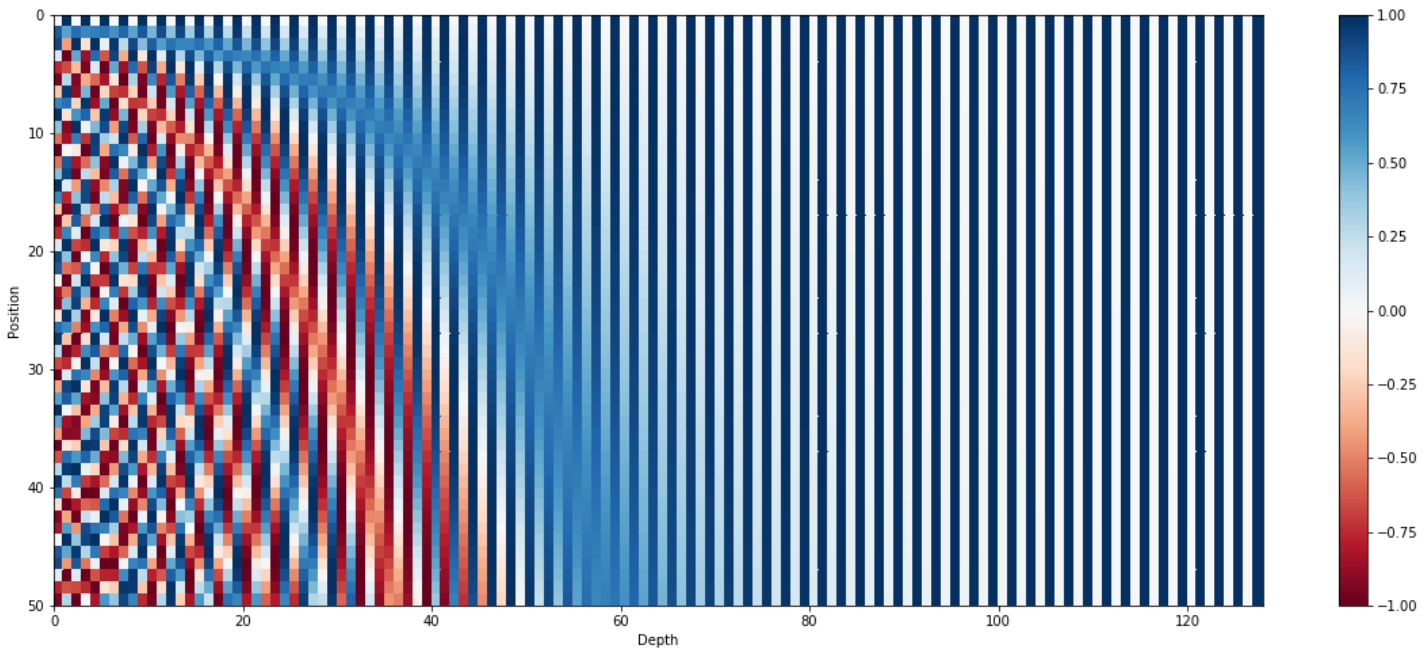
How?

- assign a number within the [0, 1] range in which 0 means the first word and 1 is the last
  - can't figure out how many words are present within a specific range
- assign a number to each time-step linearly
  - values get quite large; model may face sentences longer than the ones in training

$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases} \qquad \omega_k = \frac{1}{10000^{2k/d}}$$

output a unique encoding for each time-step (word's position in a sentence)

$$\overrightarrow{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$



allows the model to attend relative positions effortlessly

$$M.\begin{bmatrix}\sin(\omega_k.t)\\ \cos(\omega_k.t)\end{bmatrix}=\begin{bmatrix}\sin(\omega_k.(t+\phi))\\ \cos(\omega_k.(t+\phi))\end{bmatrix}$$

$$\begin{bmatrix}u_1 & v_1\\ u_2 & v_2\end{bmatrix}.\begin{bmatrix}\sin(\omega_k.t)\\ \cos(\omega_k.t)\end{bmatrix}=\begin{bmatrix}\sin(\omega_k.(t+\phi))\\ \cos(\omega_k.(t+\phi))\end{bmatrix}$$

$$M_{\phi,k}=\begin{bmatrix}\cos(\omega_k.\phi) & \sin(\omega_k.\phi)\\ -\sin(\omega_k.\phi) & \cos(\omega_k.\phi)\end{bmatrix}$$

fit GP model → get coefficients of PDE → homogeneous prediction

prior: $\boldsymbol{y} \sim \mathcal{N}(\bar{\boldsymbol{V}}, C)$

estimation: $\boldsymbol{y}^* \sim \mathcal{N}(\bar{\boldsymbol{V}}^*, C^{**})$

joint distribution: $\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}^* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \bar{\boldsymbol{V}} \\ \bar{\boldsymbol{V}}^* \end{bmatrix}, \begin{bmatrix} C + \sigma_y^2 \boldsymbol{I} & C^* \\ C^{*T} & C^{**} \end{bmatrix}\right)$

conditional predictive distribution: $p(\boldsymbol{y}^* | V^*, \boldsymbol{\phi}, \sigma_y, V, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}^* | \boldsymbol{\mu}, \Sigma)$

$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}^* | V^*, \boldsymbol{\phi}, \sigma_y, V, \boldsymbol{y}] = \bar{\boldsymbol{V}}^* + C^{*T}(C + \sigma_y^2 \boldsymbol{I})^{-1}(\boldsymbol{y} - \bar{\boldsymbol{V}})$$

$$\Sigma = cov[\boldsymbol{y}^* | V^*, \boldsymbol{\phi}, \sigma_y, V, \boldsymbol{y}] = C^{**} - C^{*T}(C + \sigma_y^2 \boldsymbol{I})^{-1} C^* + \sigma_y^2 \boldsymbol{I}$$

Estimating the PDE coefficients requires derivatives of the state variable (fitted by a GP model) with respect to independent variables.

for constant mean GP, the first order derivative:

$$p(\dot{\boldsymbol{y}}_{gp}|\boldsymbol{y},\boldsymbol{\phi}) = \mathcal{N}(\dot{\boldsymbol{\mu}}, \dot{\Sigma}_{gp}) \qquad \dot{\boldsymbol{\mu}} = \frac{\partial}{\partial v_j^*}\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{V}^*,\boldsymbol{\phi},\sigma_y,\boldsymbol{V},\boldsymbol{y}] = \dot{C}^{*T}(C+\sigma_y^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\bar{\boldsymbol{V}})$$

$$\dot{\Sigma}_{gp} = \dot{C}^{**} - \dot{C}^{*T}(C+\sigma_y^2\boldsymbol{I})^{-1}\dot{C}^{*}$$

$$cov(\boldsymbol{y}, \frac{\partial}{\partial v_j^*}\boldsymbol{y}^*) = \frac{\partial}{\partial v_j^*}k(\boldsymbol{v},\boldsymbol{v}^*) \qquad cov(\frac{\partial}{\partial v_j}\boldsymbol{y}, \frac{\partial}{\partial v_j^*}\boldsymbol{y}^*) = \frac{\partial^2}{\partial v_j \partial v_j^*}k(\boldsymbol{v},\boldsymbol{v}^*)$$

the second-derivative:

$$\ddot{\boldsymbol{y}}_{gp} = \frac{\partial^2}{\partial v_j^{*2}}\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{V}^*,\boldsymbol{\phi},\sigma_y,\boldsymbol{V},\boldsymbol{y}] = \ddot{C}^{*T}(C+\sigma_y^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\bar{\boldsymbol{V}})$$

$$cov(y, \frac{\partial^2}{\partial v_j^{*2}}\boldsymbol{y}^*) = \frac{\partial^2}{\partial v_j^{*2}}k(\boldsymbol{v},\boldsymbol{v}^*)$$

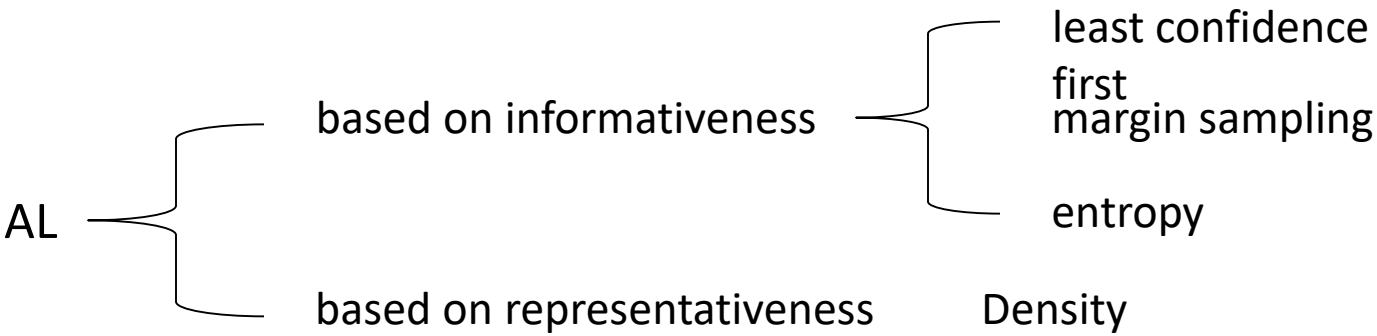
the residual error in the PDE at an observed point is given by:

$$\epsilon = f\Big(v_1, \cdots, v_m, y, \lambda \; \mathcal{GP}\{\frac{\partial y}{\partial v_1}, \cdots, \frac{\partial y}{\partial v_m}, \frac{\partial^2 y}{\partial v_1 \partial v_1}, \cdots, \frac{\partial^2 y}{\partial v_1 \partial v_m}, \cdots, \}\Big)$$

minimizing the sum of square of residual error (SSRE)

$$SSRE = \epsilon^T \epsilon = \sum_{\forall \boldsymbol{v}, \boldsymbol{y} \in \mathbb{D}} f\Big(v_1, \cdots, v_m, y, \lambda, \; \mathcal{GP}\{\frac{\partial y}{\partial v_1}, \cdots, \frac{\partial y}{\partial v_m}, \frac{\partial^2 y}{\partial v_1 \partial v_1}, \cdots, \frac{\partial^2 y}{\partial v_1 \partial v_m}, \cdots, \}\Big)^2$$

Kriging Sampling for Active Learning



$$\mathcal{Y}(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

$$\hat{y}(x^*) = \mu(x^*) + c^T C^{-1}(y - \mu)$$

mean squared error (MSE) of this prediction:

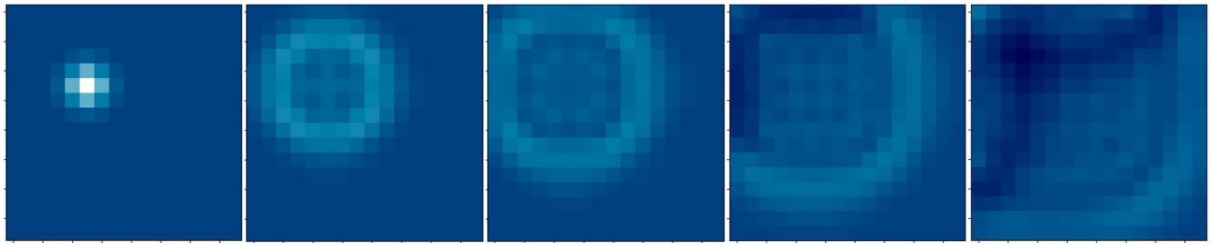
$$\hat{s}^2(x^*) = \mathbb{E}\{(\hat{y}(x^*) - \mathcal{Y}(x^*))^2\}$$

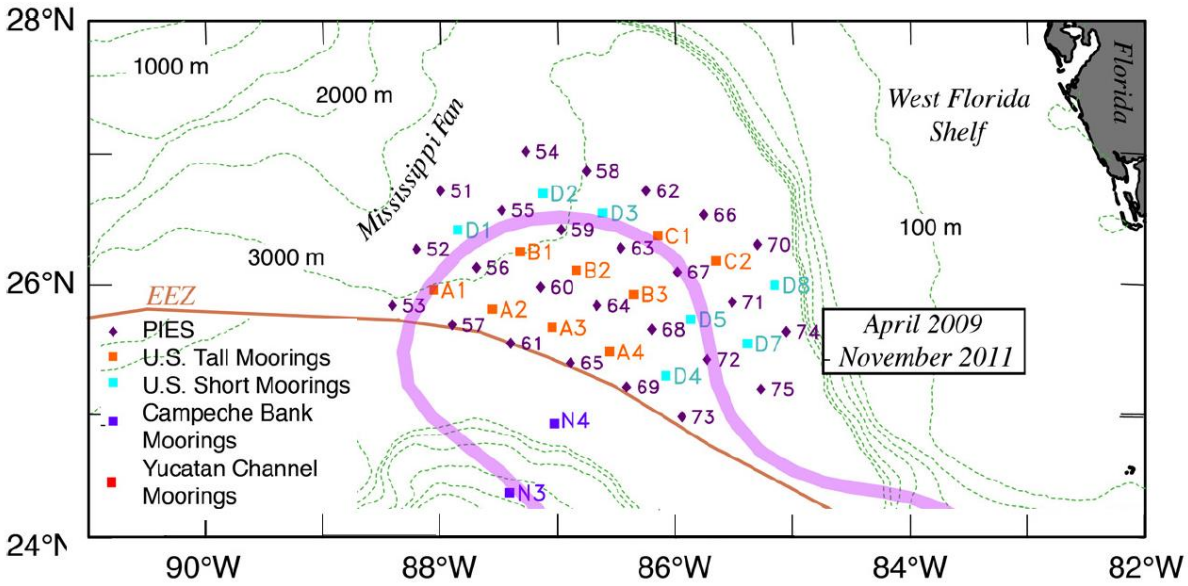
$$\hat{s}^2(x^*) = \sigma^2(x^*) - c^T C^{-1} c$$

Advantages:
Nonparametric
obtain model's uncertainty directly
fit for nonlinear data
Disdvantages:
need to calculate inverse matrix of all training data points

EXPERIMENTS

- Reflected Wave Simulation Data
- Gulf of Mexico (GoM) Loop Current Data





Ablation study:

Random sampling

Without physics learning

All data available

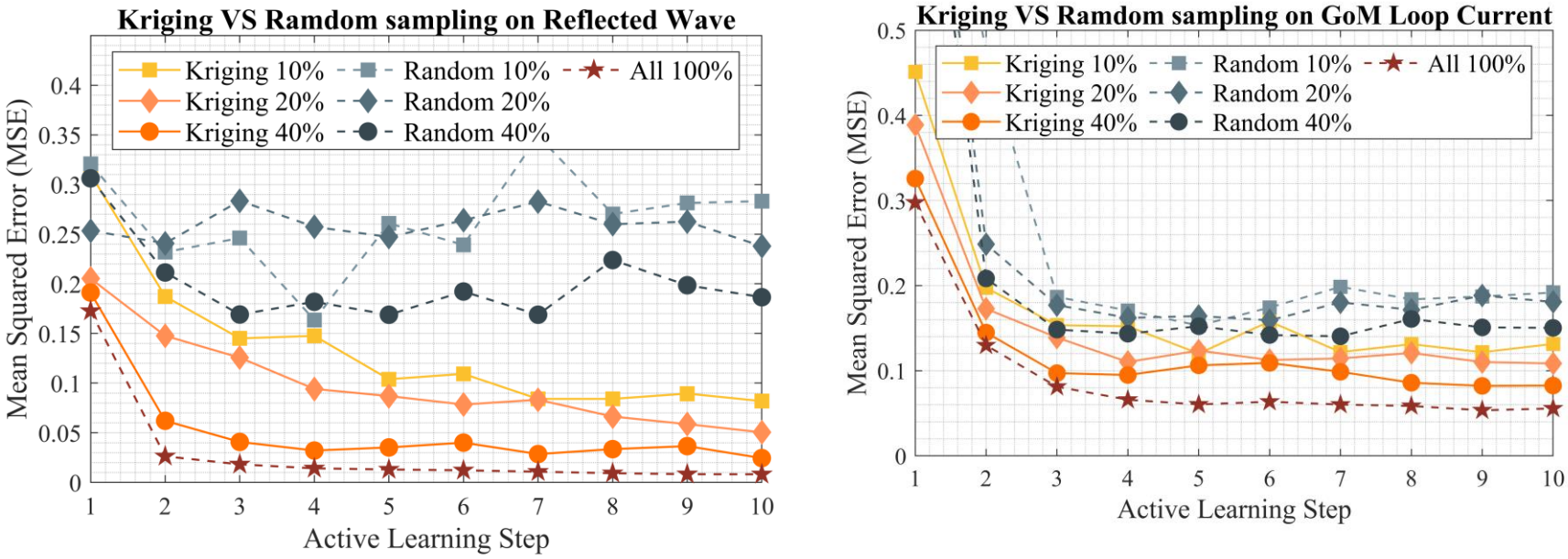


Figure 6: Explore Kriging and Random sampling effect on the performance of ST-PCNN on unseen data.

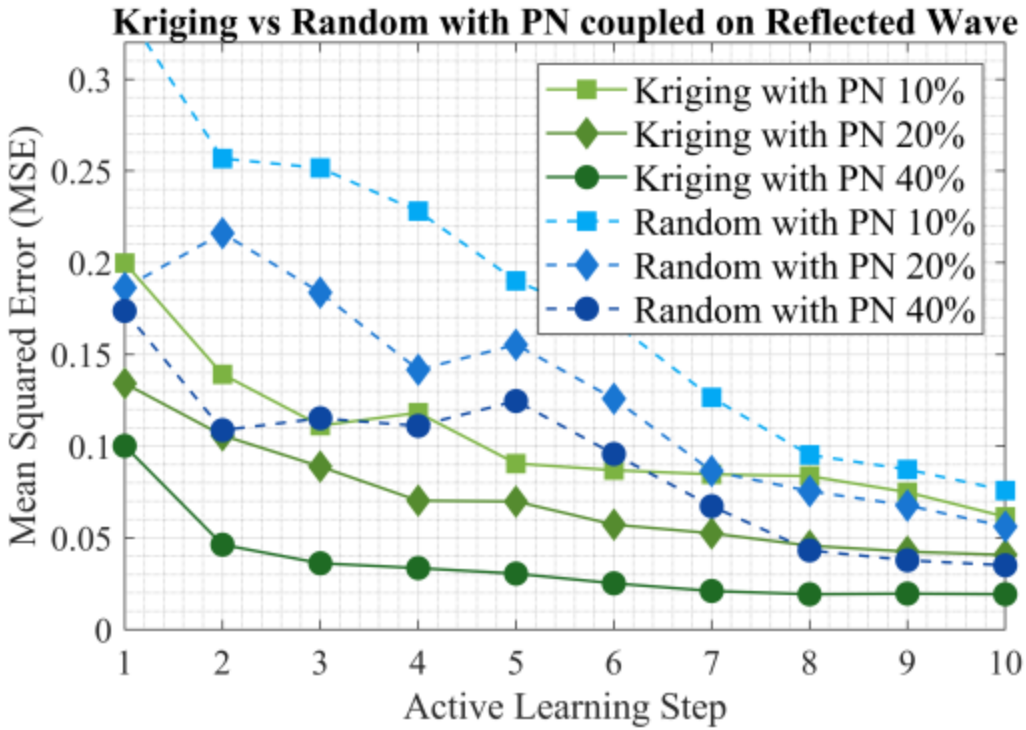


Figure 7: Explore Kriging and random sampling effect on performance of ST-PCNN on unseen data (the effect on GoM Loop Current refers to Figures 6 and 8).

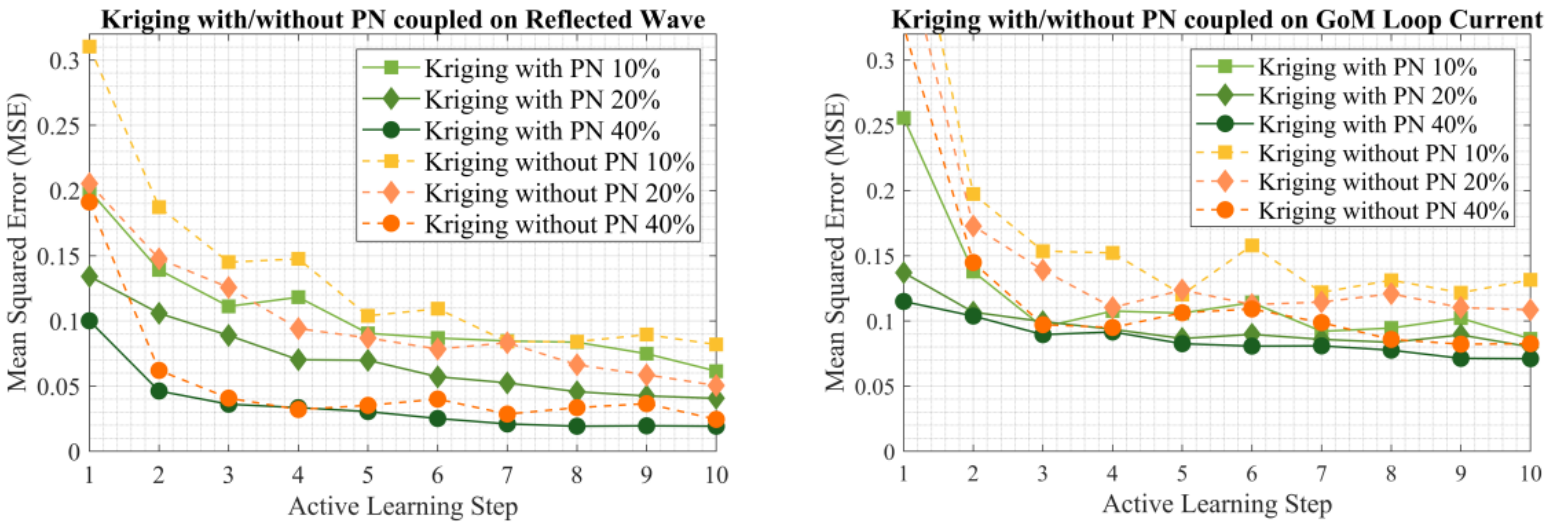


Figure 8: Explore physics effect on performance of ST-PCNN (with Kriging) on unseen data.