

PINNACLE: PINN Adaptive ColLocation and Experimental Points Selection



Gregory Kang Ruey Lau* 1,2

Apivich Hemachandra* 1

See-Kiong Ng¹

Bryan Kian Hsiang Low¹

* Equal contribution

¹ Department of Computer Science, National University of Singapore

² CNRS@Create



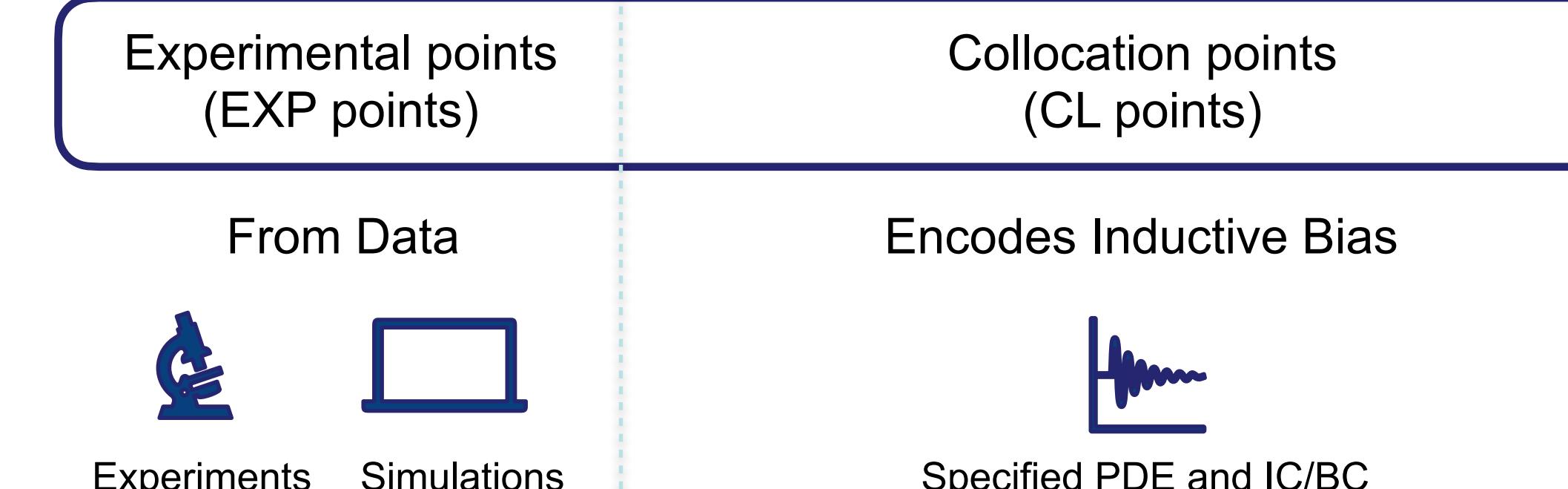
Scan me!

Training Point Selection in PINNs

- Physics-Informed Neural Networks (PINN) incorporate PDEs as soft constraints/regularization terms
- This makes PINNs hard to train, due to:
 - Different interacting training dynamics
 - A large number of training points needed

$$\mathcal{L}(\hat{u}_\theta; X) = \sum_{x \in X_s} \frac{(\hat{u}_\theta(x) - u(x))^2}{2N_s} + \lambda_p \sum_{x \in X_p} \frac{(\mathcal{N}[\hat{u}_\theta](x) - f(x))^2}{2N_p} + \lambda_b \sum_{x \in X_b} \frac{(\mathcal{B}[\hat{u}_\theta](x) - g(x))^2}{2N_b}$$

EXP points PDE CL points IC/BC CL points



Can we jointly select all types of training points in order to improve the training of PINNs?

eNTK Eigenfunctions in Augmented Space

$$\mathcal{Z} = \{(x, s) : x \in \mathcal{X}\} \cup \{(x, p) : x \in \mathcal{X}\} \cup \{(x, b) : x \in \partial\mathcal{X}\}$$

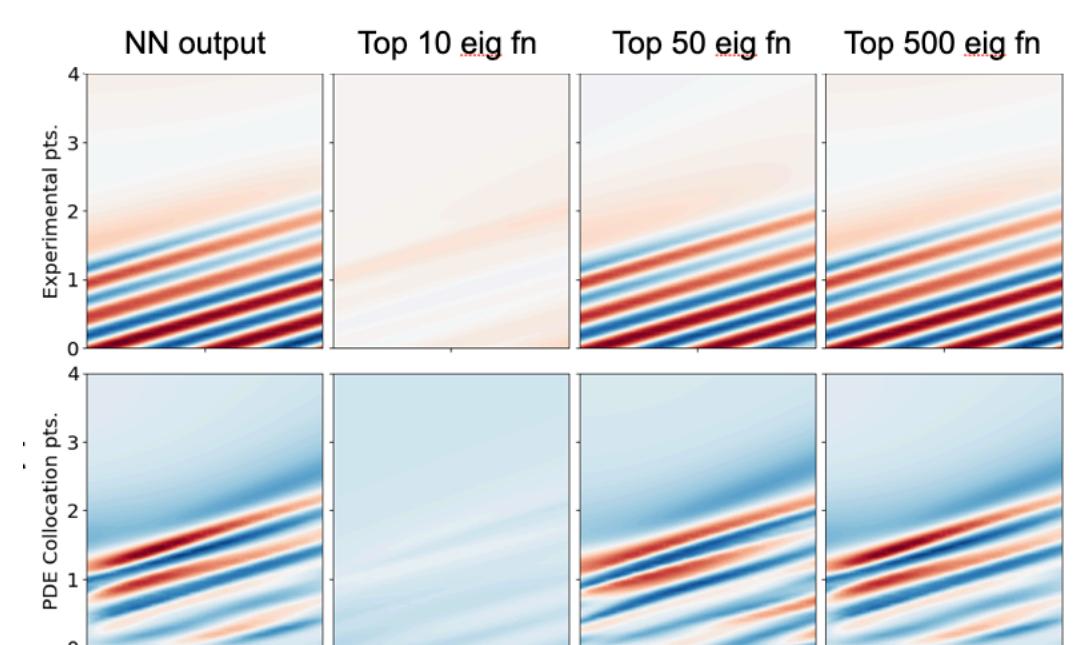
EXP points PDE CL points IC/BC CL points

$$F[h](x, s) = h(x), \quad F[h](x, p) = \mathcal{N}[h](x), \quad F[h](x, b) = \mathcal{B}[h](x)$$

Empirical NTK (eNTK)

$$\Theta_t(z, z') = \nabla_\theta F[\hat{u}_{\theta_t}](z) \nabla_\theta F[\hat{u}_{\theta_t}](z')^\top$$

Dominant eigenfunctions form basis of NN output in augmented space



Training Dynamics

- Residual component that aligns with dominant eigenfunctions will decay faster

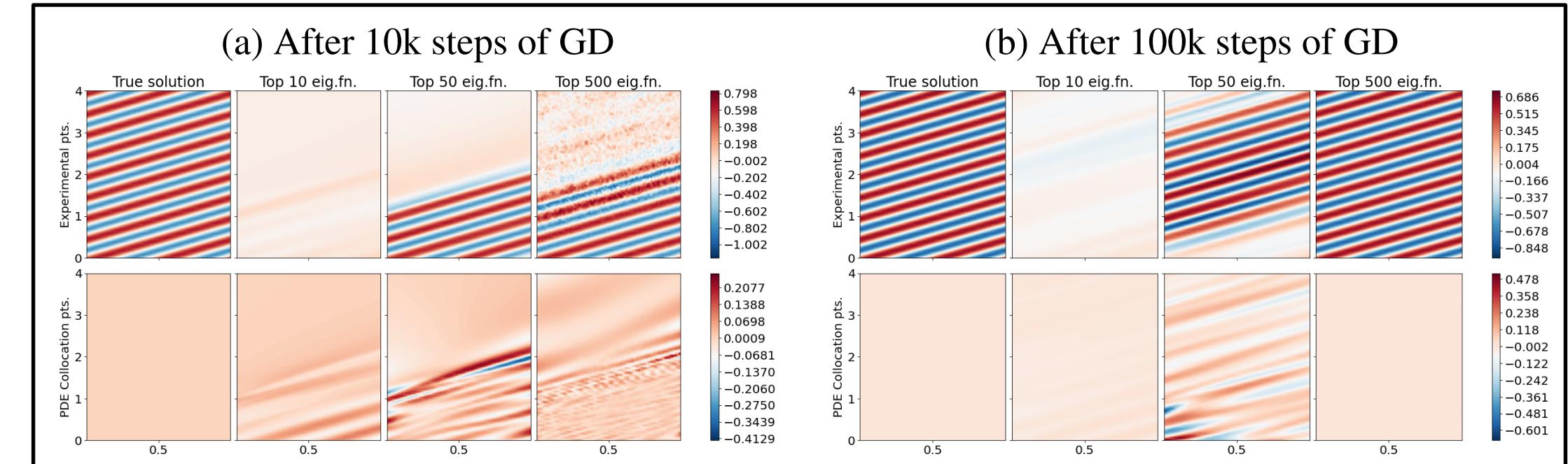
Convergence criterion

RKHS norm of residue change when trained on Z

$$\alpha(Z) \triangleq \|\Delta R_{\theta_t}(\cdot; Z)\|_{\mathcal{H}_{\Theta_t}}^2 = \sum_{i=0}^{\infty} \lambda_{t,i}^{-1} \langle \Delta R_{\theta_t}(\cdot; Z), \psi_{t,i} \rangle_{\mathcal{H}_{\Theta_t}}^2$$

Criterion related to PINN generalization error bound (Thm. 1), and can be approximated using Nystrom approximation (Prop. 1)

- The eNTK eigenspectrum evolves throughout training



Algorithm: Iterative phases of (1) point selection to maximize convergence degree and (2) PINN training

Algorithm 1 PINNACLE

```

1: Input: PINN  $\hat{u}_\theta$ , learning rate  $\eta$ , number of iterations  $T$ , eNTK approx. error  $\delta$ .
2: repeat
3:   // Point selection phase
4:   Randomly sample candidates  $Z_{pool}$  from  $\mathcal{Z}$ 
5:   Compute  $\Theta_t$  using Nystrom approximation
6:   Select subset  $Z \subset Z_{pool}$  to fit constraint using SAMPLING or K-MEANS++
7:   // Training phase
8:   Compute  $\bar{\Theta} = \Theta_t(Z_{pool})$ 
9:   for  $t' = t, \dots, t+T$  do
10:     $\theta_{t'+1} \leftarrow \theta_{t'} - \eta \nabla_\theta \mathcal{L}(\hat{u}_{\theta_{t'}}, Z)$ 
11:    Exit training if  $\|\bar{\Theta} - \Theta_{t'}(Z_{pool})\| \geq \delta \|\bar{\Theta}\|$ 
12: until training converges or budget exhausted

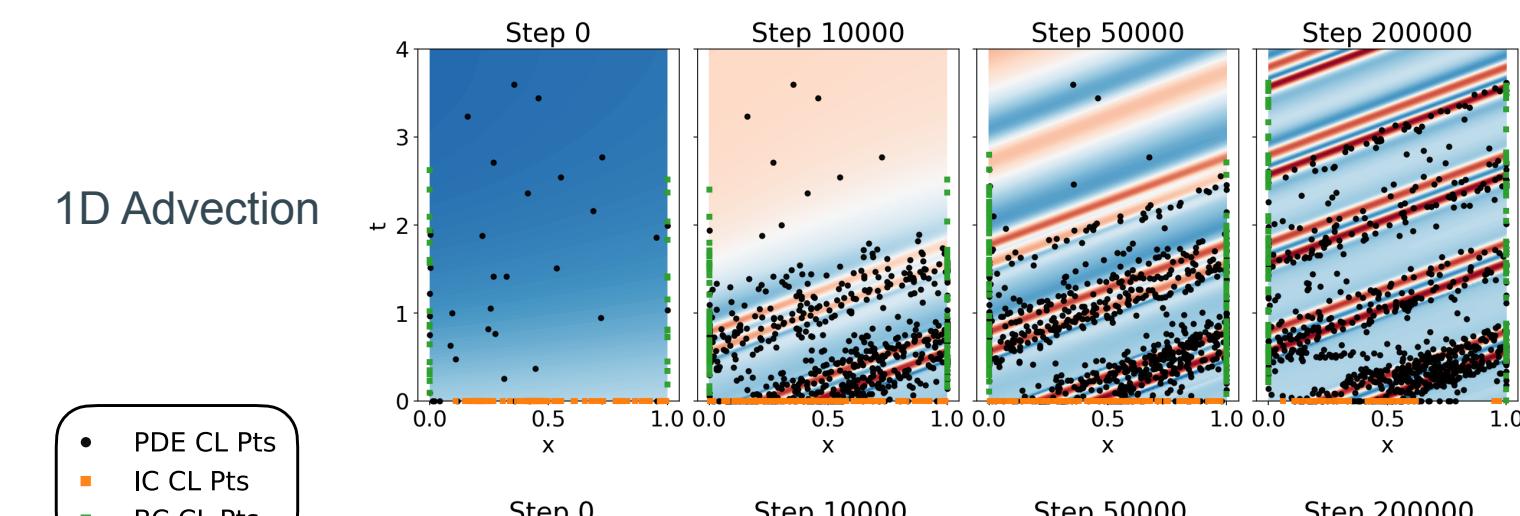
```

- PINNACLE-S: Sample z with probability proportional to $\hat{\alpha}(z)$
- PINNACLE-K: Perform K-Means++ initialization on the embedding $z \mapsto (\hat{\lambda}_{t,i}^{1/2} \hat{a}_{t,i}(z))_{i=1}^p$

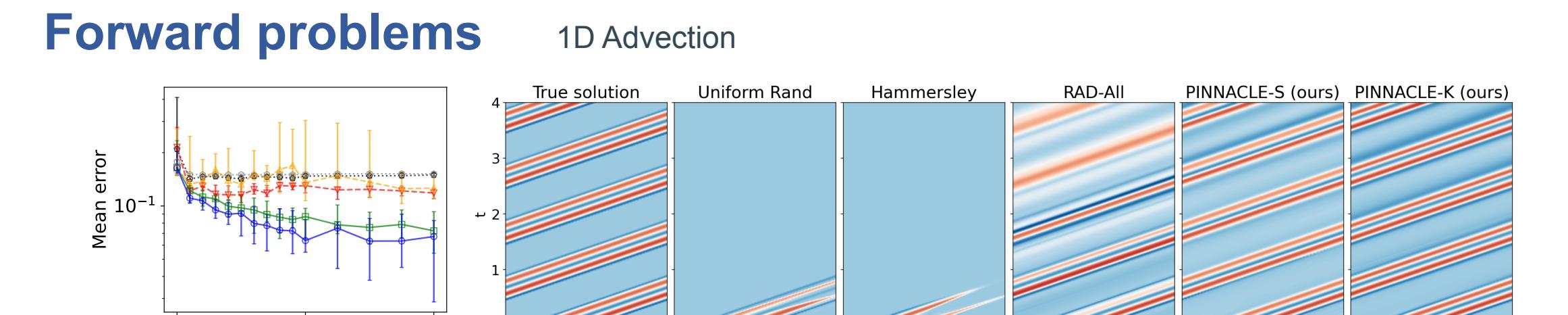
Re-compute training points when eNTK has evolved with eigenfunctions that are more aligned with labels/true solution

Empirical results: PINNACLE outperforms benchmarks for various tasks with interpretable point selection

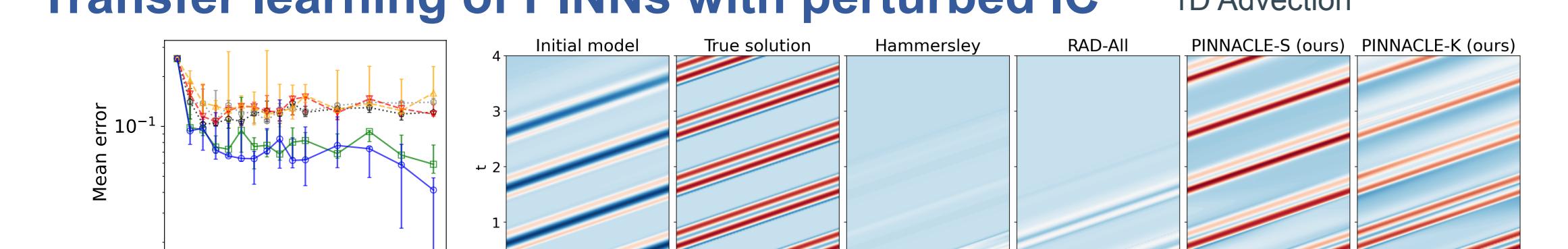
Point selection behavior of PINNACLE



Forward problems



Transfer learning of PINNs with perturbed IC



Inverse problems

