NEURAL INFORMATION PROCESSING SYSTEMS

Hypernetwork-based Meta-Learning for Low-Rank Physics-Informed Neural Networks

Woojin Cho, Kookjin Lee, Donsub Rim, Noseong Park

YONSEI UNIVERSITY

Arizona State University

Washington University in St.Louis

Preliminaries: Physics-informed Neural Networks

Solving PDE with coordinate-based MLP **(PINN)**

How to train?

- $L \stackrel{\text{def}}{=} \alpha L_u + \beta L_f$ **(Total loss)**
  - $L_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| F\left(x_f^i, t_f^i, \tilde{u}; \theta\right) \right|^2$ **(PDE residual loss)**
  - $L_u = \frac{1}{N_u} \sum_{i=1}^{N_u} \left| u\left(x_u^i, t_u^i\right) - \tilde{u}\left(x_u^i, t_u^i; \theta\right) \right|^2$ **(Boundary loss)**

F: PDE operator
(x_f, t_f): collocation points
(x_u, t_u): initial & boundary points
N_f: # collocation points
N_u: # initial & boundary points

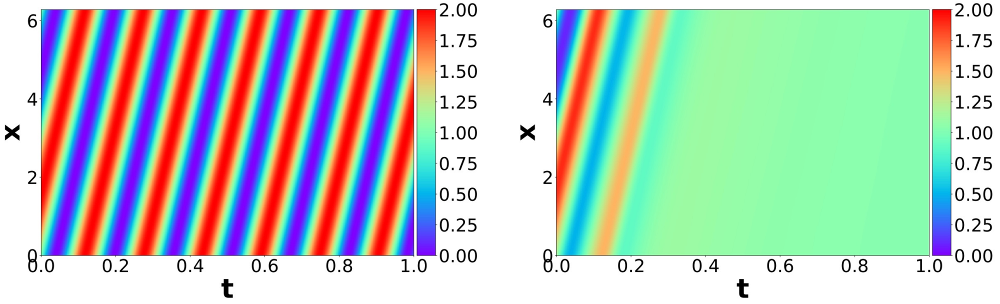
Motivation

Limitation of PINNs

- For a new data instance, training a new neural network is required.

  → Not suitable for **many-query** scenarios (especially, parameterized PDEs)

- **"Failure mode"** of PINNs



(a) Exact solution (b) PINN

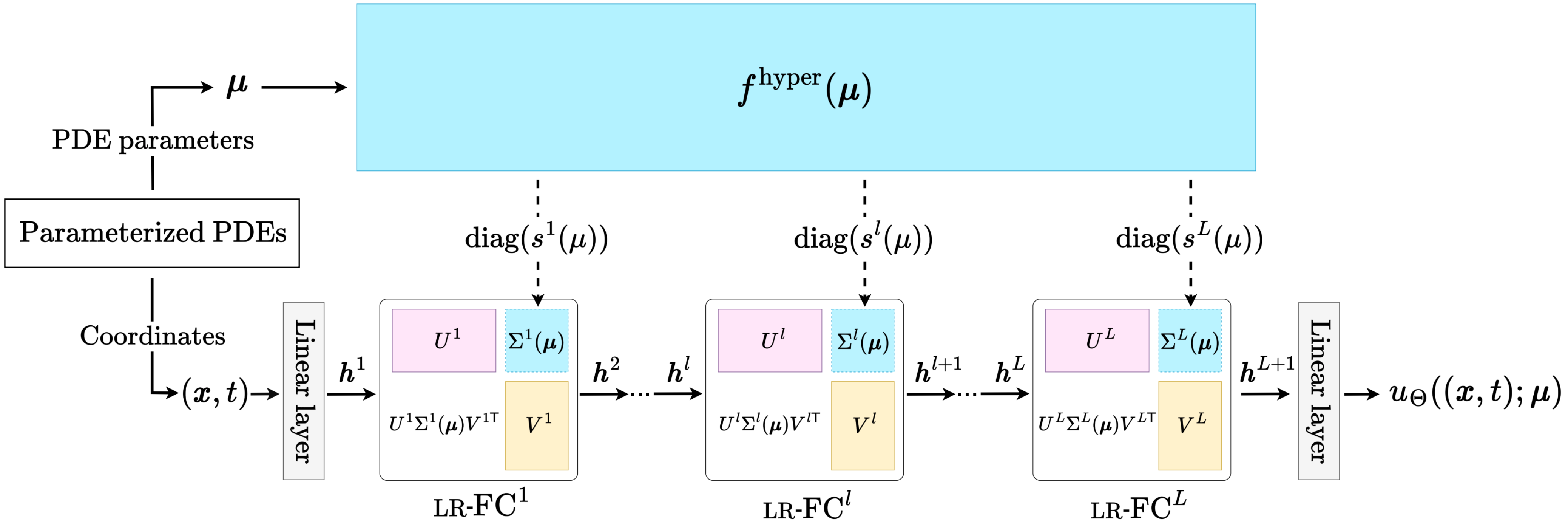
Our proposed method includes:

i) a low-rank structured neural network architecture for PINNs, i.e., LR-PINNs

ii) an efficient rank-revealing training algorithm, which adaptively adjust ranks of LR-PINNs for varying PDE inputs

iii) a two-phase procedure (offline training / online testing) for handling many-query scenarios

Proposed Method



Low-rank PINN (LR-PINN)

$$\begin{aligned} \boldsymbol{h}^1 &= \sigma(W^0\boldsymbol{h}^0 + \boldsymbol{b}^0), \\ \boldsymbol{h}^{l+1} &= \sigma(U^l(\Sigma^l(\boldsymbol{\mu})(V^{l\mathsf{T}}\boldsymbol{h}^l)) + \boldsymbol{b}^l), \\ u_\Theta((\boldsymbol{x},t);\boldsymbol{\mu}) &= \sigma(W^{L+1}\boldsymbol{h}^{L+1} + \boldsymbol{b}^{L+1}), \end{aligned}$$

where $\Sigma^l(\boldsymbol{\mu}) = \text{diag}(\boldsymbol{s}^l(\boldsymbol{\mu}))$

+ Meta learning

Hypernetwork based Low-rank PINN (Hyper-LR-PINN)

$$\begin{aligned} \boldsymbol{e}^m &= \sigma(W^{\text{emb},m}\boldsymbol{e}^{m-1} + \boldsymbol{b}^{\text{emb},m}), \quad m = 1,\dots,M, \\ \boldsymbol{s}^l(\boldsymbol{\mu}) &= \text{ReLU}(W^{\text{hyper},l}\boldsymbol{e}^M + \boldsymbol{b}^{\text{hyper},l}), \quad l = 1,\dots,L, \end{aligned}$$

Model architecture

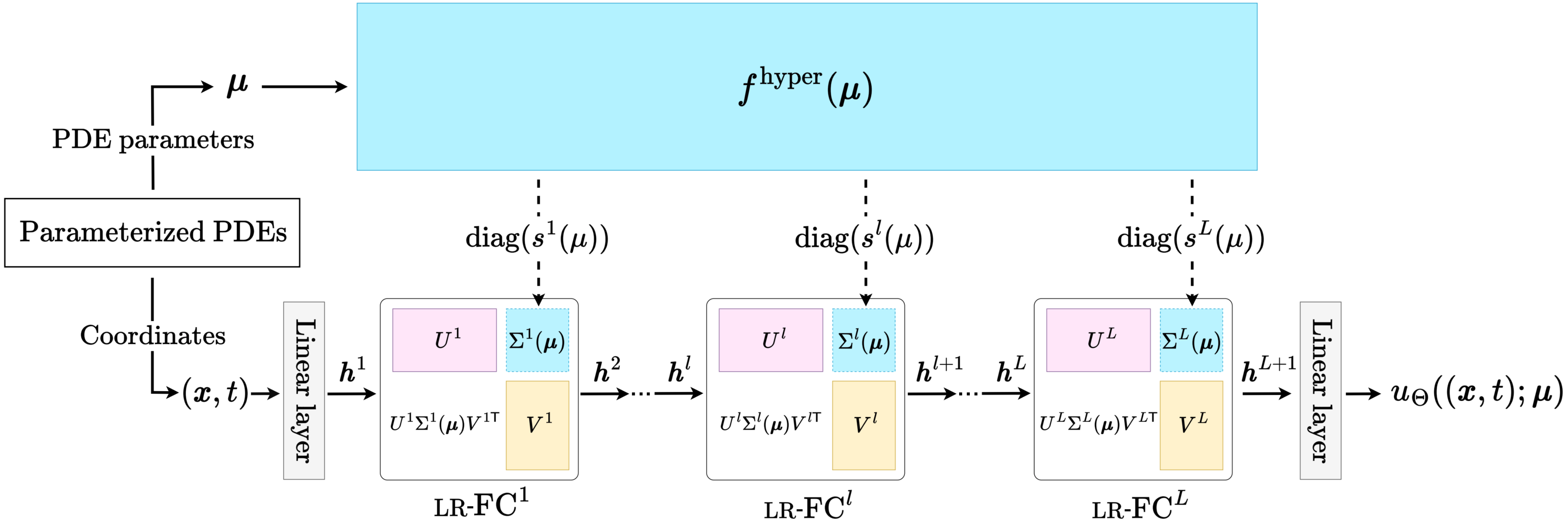


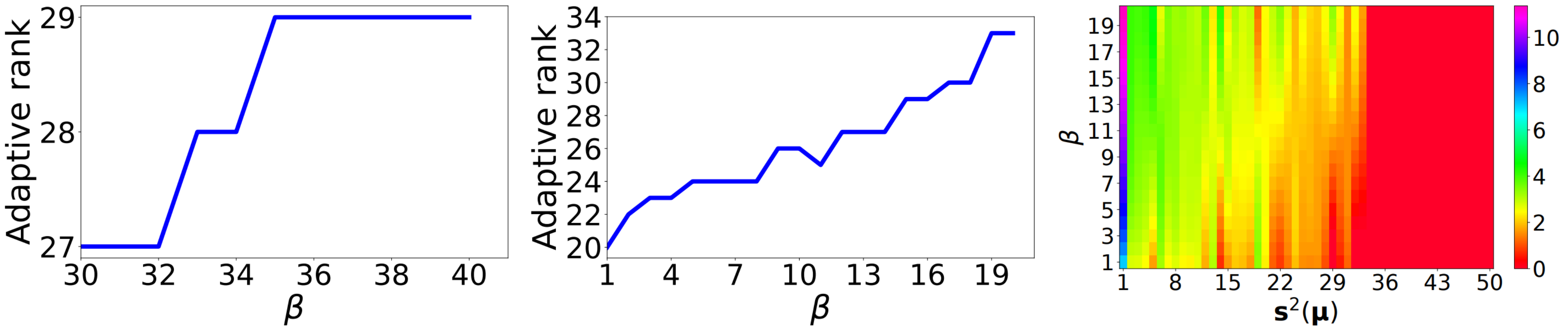
Table 1: Learnable parameters in each phase

Phase 1	$\{(U^l, V^l, \boldsymbol{b}^l)\}_{l=1}^{L}, W^0, W^{L+1}, \boldsymbol{b}^0, \boldsymbol{b}^{L+1}$ $\{(W^{\text{emb},m}, \boldsymbol{b}^{\text{emb},m})\}_{m=1}^{M}, \{(W^{\text{hyper},l}, \boldsymbol{b}^{\text{hyper},l})\}_{l=1}^{L}$	(LR-PINN), (hypernetwork),
Phase 2	$\{\boldsymbol{s}^l\}_{l=1}^{L}, W^0, W^{L+1}, \boldsymbol{b}^0, \boldsymbol{b}^{L+1}$	(LR-PINN)

Experiments

Table 3: Comparisons of model size

Model	**Naïve-LR-PINN**					**Ours**	**PINN**
Rank	10	20	30	40	50	Adaptive	-
# Parameters	381	411	441	471	501	$\sim$351	10,401



(a) $\beta \in [30,40]$ (b) $\beta \in [1,20]$ (c) $\beta \in [1,20]$

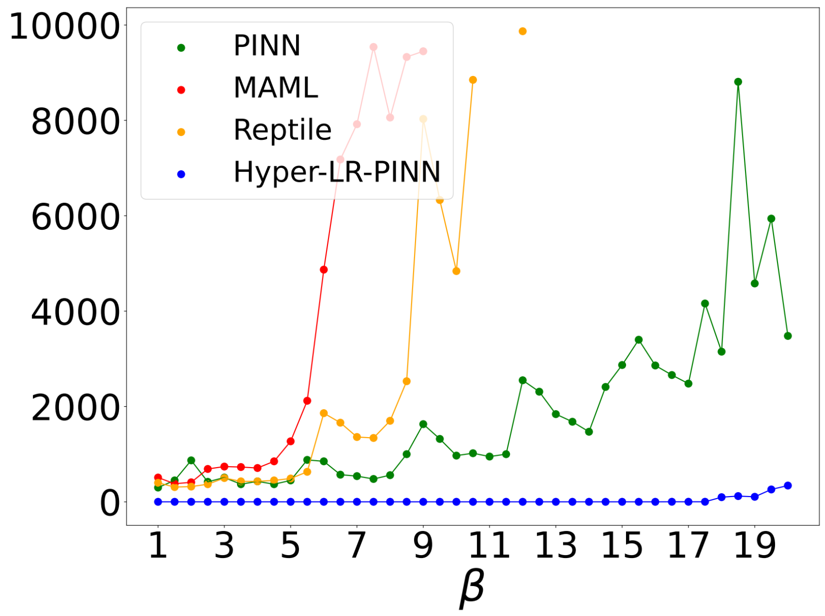
Adaptive rank on convection equation (the left and the middle panels). The magnitude of the learned diagonal elements of the second hidden layer (the right panel).

Experiments

β	Rank	[w/o] Pre-training		[w] Pre-training									
		Naïve-LR-PINN		Curriculum learning		MAML		Reptile		Hyper-LR-PINN (Full rank)		Hyper-LR-PINN (Adaptive rank)	
		Abs. err.	Rel. err.	Abs. err.	Rel. err.	Abs. err.	Rel. err.	Abs. err.	Rel. err.	Abs. err.	Rel. err.	Abs. err.	Rel. err.
	10	0.5617	0.5344	0.4117	0.4098	0.6757	0.6294	0.5893	0.5551				
	20	0.5501	0.5253	0.4023	0.4005	0.6836	0.6452	0.6144	0.5779				
30	30	0.5327	0.5126	0.4233	0.4204	0.5781	0.5451	0.6048	0.5704	0.0360	0.0379	0.0375	0.0389
	40	0.5257	0.5076	0.3746	0.3744	0.5848	0.5515	0.5757	0.5442				
	50	0.5327	0.5126	0.4152	0.4127	0.5898	0.5562	0.5817	0.5496				
	10	0.5663	0.5357	0.5825	0.5465	0.6663	0.6213	0.5786	0.5446				
	20	0.5675	0.5369	0.6120	0.5673	0.6814	0.6433	0.5971	0.5606				
35	30	0.6081	0.5670	0.5864	0.5503	0.5819	0.5466	0.5866	0.5506	0.0428	0.0443	0.0448	0.0461
	40	0.5477	0.5227	0.5954	0.5548	0.5809	0.5462	0.5773	0.5435				
	50	0.5449	0.5208	0.6010	0.5619	0.5870	0.5514	0.5731	0.5404				
	10	0.5974	0.5632	0.5978	0.5611	0.6789	0.6446	0.5992	0.5632				
	20	0.5890	0.5563	0.6274	0.5820	0.7008	0.6801	0.6189	0.5853				
40	30	0.6142	0.5724	0.6011	0.5652	0.6072	0.5700	0.6126	0.5810	0.0603	0.0655	0.0656	0.0722
	40	0.5560	0.5293	0.6126	0.5715	0.6149	0.5832	0.6004	0.5638				
	50	0.6161	0.5855	0.6130	0.5757	0.6146	0.5799	0.6007	0.5645				

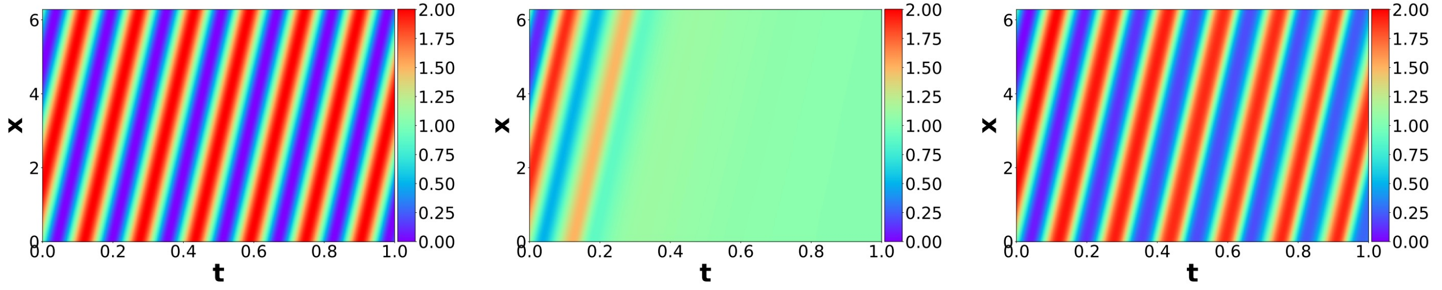
Conclusion

Lower computational cost in many-query scenario



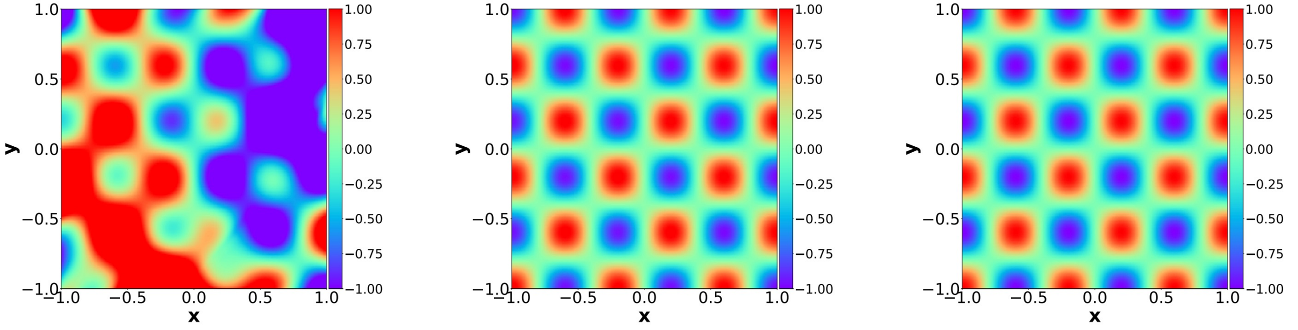
Resolving failure modes of PINNs

[Convection equation] Solution snapshots for $\beta = 40$



(a) Exact solution (b) PINN (c) Ours

[Helmholtz equation] Solution snapshots for $a = 2.5$



(a) PINN (Abs.err.=0.7403) (b) Ours (Abs.err.=0.0285) (c) Exact solution