ICML
International Conference
On Machine Learning

Parameterized Physics-informed Neural Networks for Parameterized PDEs

Woojin Cho[1,2], Minju Jo[3], Haksoo Lim[1], Kookjin Lee[2], Dongeun Lee[4], Sanghyun Hong[5], Noseong Park[6]

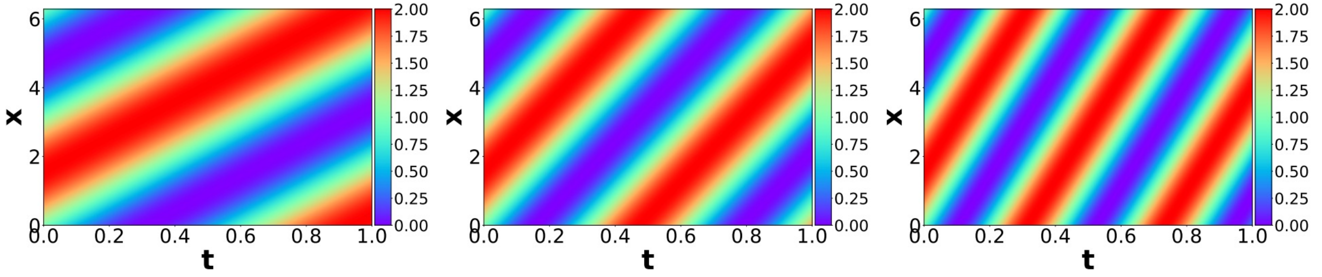
[1]Yonsei University, [2]Arizona State University, [3]LG CNS, [4]Texas A&M University-Commerce, [5]Oregon State University, [6]KAIST

Preliminaries: Parameterized PDEs

Example: Convection equation

Governing Equation: $$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} = 0$$

PDE parameter: β

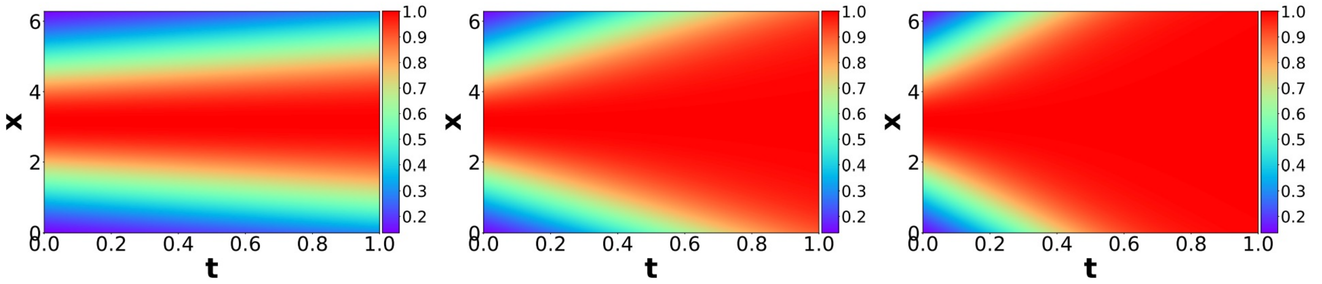


(a) Conv. ($\beta = 5$) (b) Conv. ($\beta = 10$) (c) Conv. ($\beta = 15$)

Example: Reaction equation

Governing Equation: $$\frac{\partial u}{\partial t} - \rho u(1-u) = 0$$

PDE parameter: ρ



(d) Reac. ($\rho = 1$) (e) Reac. ($\rho = 4$) (f) Reac. ($\rho = 7$)

Parameterized PDEs are a type of partial differential equation that include specific parameters within the equation. These PDE parameters can reflect the physical properties of the system.

Preliminaries: Physics-informed Neural Networks

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

$$(\boldsymbol{x}, t) \longrightarrow \boxed{Neural\ Network: \theta} \longrightarrow \tilde{u}$$

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

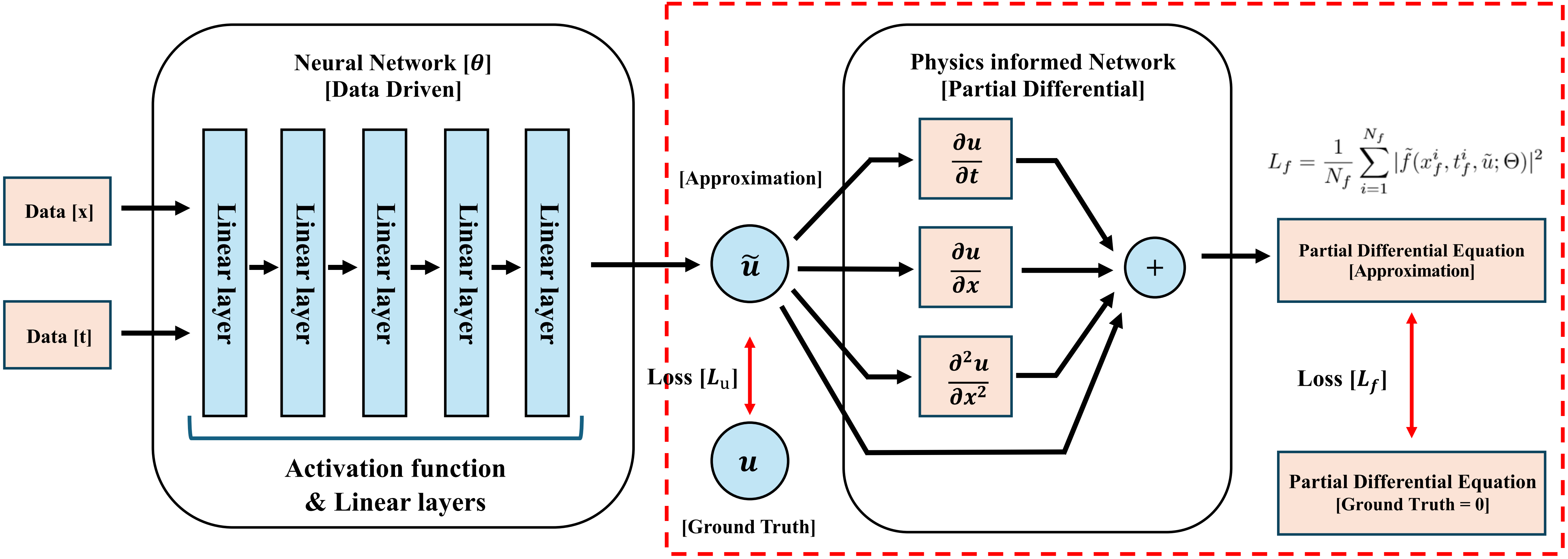
(x_u, t_u): initial & boundary points

N_f: # collocation points

N_u: # initial & boundary points

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equation, **Journal of Computational Physics 2019**

Physics informed Neural Network



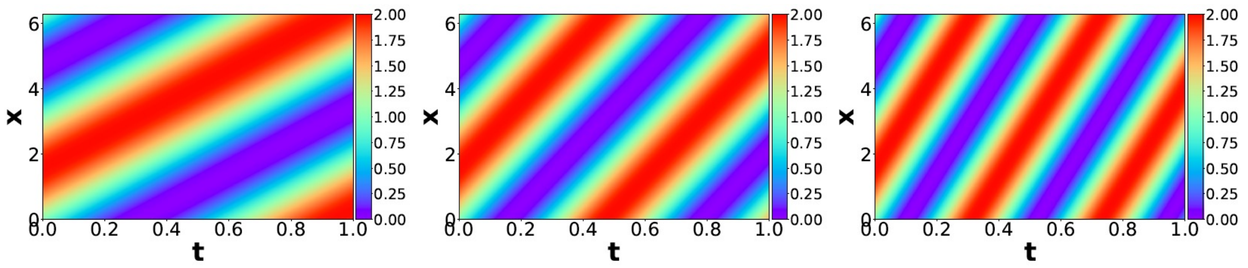
Weaknesses of PINNs

W1. PINNs rely on the PDE loss, and this loss function is a non-convex function.

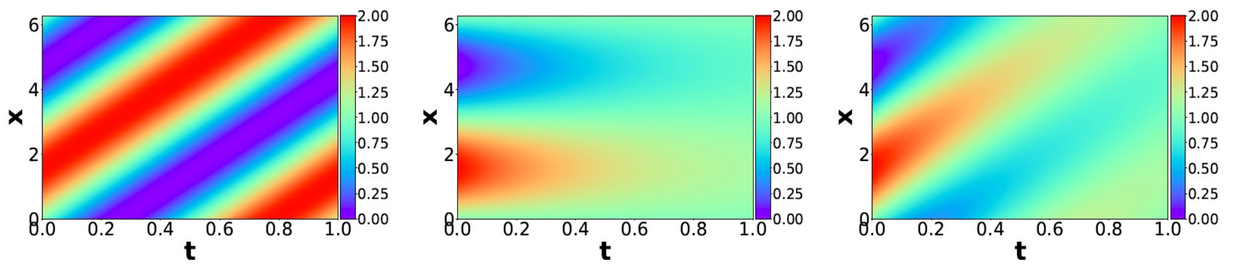
W2. A single PINN can learn only one governing equation.

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equation, **Journal of Computational Physics 2019**

Motivations



(a) Conv. ($\beta = 5$) (b) Conv. ($\beta = 10$) (c) Conv. ($\beta = 15$)



(a) Conv. (b) Diff. (c) Conv.-Diff.

1. A latent space of parameterized PDEs may exist.

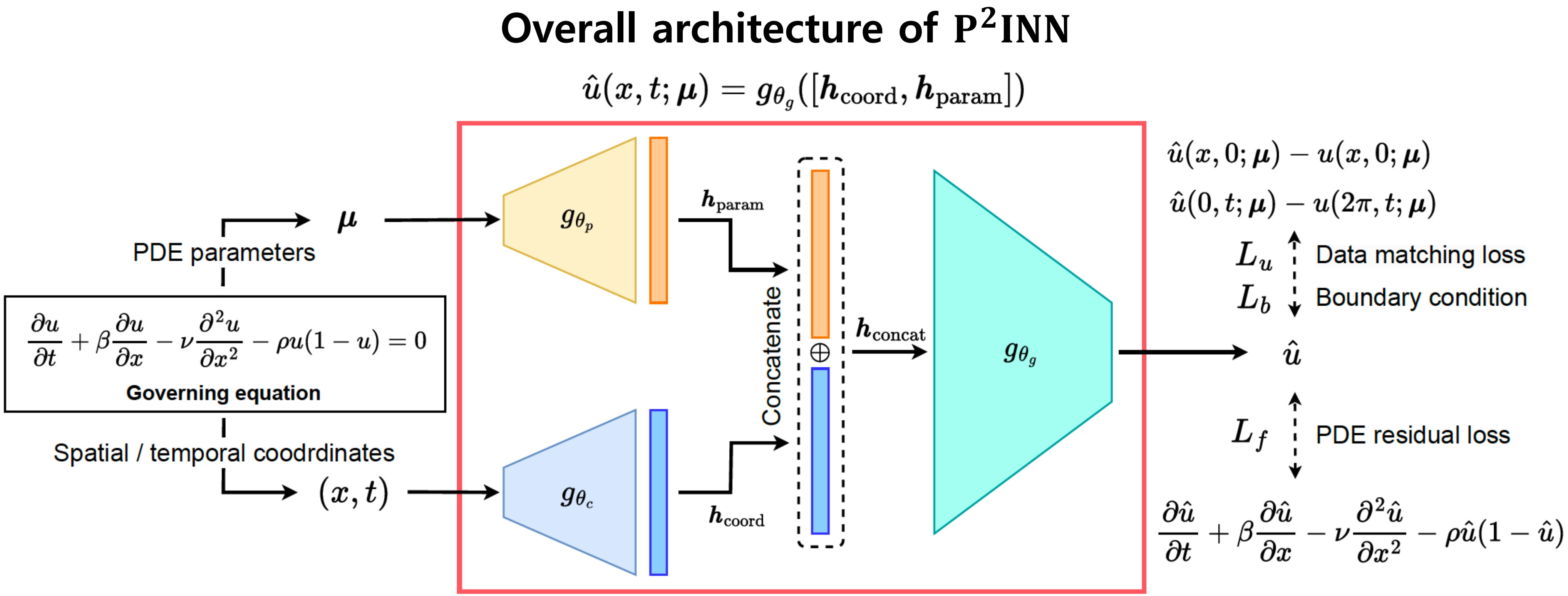
2. It will be more effective to solve similar problems simultaneously.

Contributions

1) We show that the proposed method **addresses the failure modes** of PINNs.
2) We present a new efficient PINN framework for **multi-query** scenarios.
3) We develop a **SVD modulation** for solving multiple PDEs.

Parameterized Physics-informed neural networks for Parameterized PDEs, **ICML 2024 (Oral)**

Proposed Method



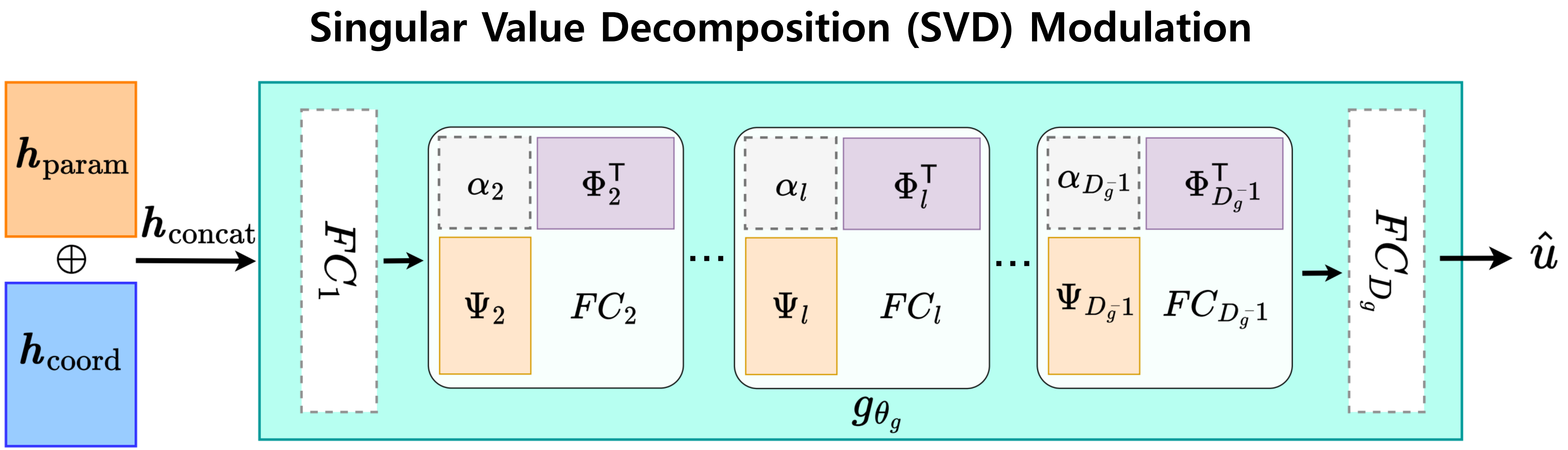
$$u_\theta(x,t;\boldsymbol{\mu}) = g_{\theta_g}\left(\left[g_{\theta_c}(x,t); g_{\theta_p}(\mu)\right]\right),$$

$$\boldsymbol{h}_{param} = g_{\theta_p}(\boldsymbol{\mu}) \text{ and } \boldsymbol{h}_{coord} = g_{\theta_c}(x,t)$$

$$\left\{ \begin{aligned} \boldsymbol{h}_{param} &= \sigma(FC_{D_p} \cdots (\sigma(FC_2(\sigma(FC_1(\boldsymbol{\mu})))))) \\ \boldsymbol{h}_{coord} &= \sigma(FC_{D_c} \cdots (\sigma(FC_2(\sigma(FC_1(x,t)))))) \end{aligned} \right\}$$

The manifold network g_{θ_g} reads the two hidden re presentations, $\boldsymbol{h}_{param}$ and $\boldsymbol{h}_{coord}$, and infer the inp ut equation's solution at (x,t).

Proposed Method



$$FC_l = \psi_l \, \alpha_l \, \phi_l^{\mathsf{T}}, \quad l = 2, 3, \ldots, D_g - 1$$

From the pre-trained decoder layer of $\mathrm{P^2INN}$, we obtain the bases ψ_l, ϕ_l for parameterized PDEs through SVD. During fine-tuning, we set $\{\alpha_l\}_{l=2}^{D_g-1}$ to be learnable, while keeping all other paramet ers in the network fixed. It is an option to fix the parameters of FC_1 and FC_{D_g}

Experimental Results

Addressing W1: PINNs rely on the PDE loss, and this loss function is a non-convex function.

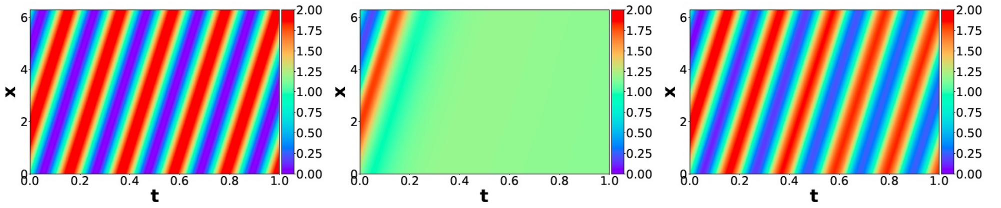
<1D CDR equations>

The relative and absolute L_2errors

	PDE type	Metric	PINN	LargePINN	PINN-P	P2INN
Class 1	Convection	Abs. err.	0.1140	0.1191	0.0209	**0.0198**
		Rel. err.	0.1978	0.2084	**0.0410**	0.0464
	Diffusion	Abs. err.	0.6782	0.5868	0.3800	**0.1916**
		Rel. err.	1.2825	1.0994	0.7912	**0.3745**
	Reaction	Abs. err.	0.7902	0.7910	0.8975	**0.0042**
		Rel. err.	0.8460	0.8469	0.9908	**0.0092**
Class 2	Conv.-Diff.	Abs. err.	0.2735	0.1626	0.1253	**0.0622**
		Rel. err.	0.5106	0.3189	0.3009	**0.1495**
	Reac.-Diff.	Abs. err.	0.7167	0.7399	0.1756	**0.0898**
		Rel. err.	0.7998	0.8186	0.2632	**0.1411**
Class 3	Conv.-Diff.-Reac.	Abs. err.	0.7450	0.7415	0.8590	**0.0353**
		Rel. err.	0.7960	0.7915	0.9532	**0.0812**

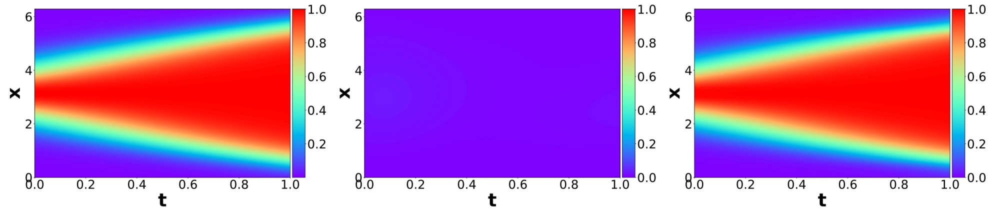
Failure modes

Convection equation $(\boldsymbol{\beta} = \mathbf{30})$



(a) Exact (b) PINN (c) P^2INN

Reaction equation $(\boldsymbol{\rho} = \mathbf{5})$



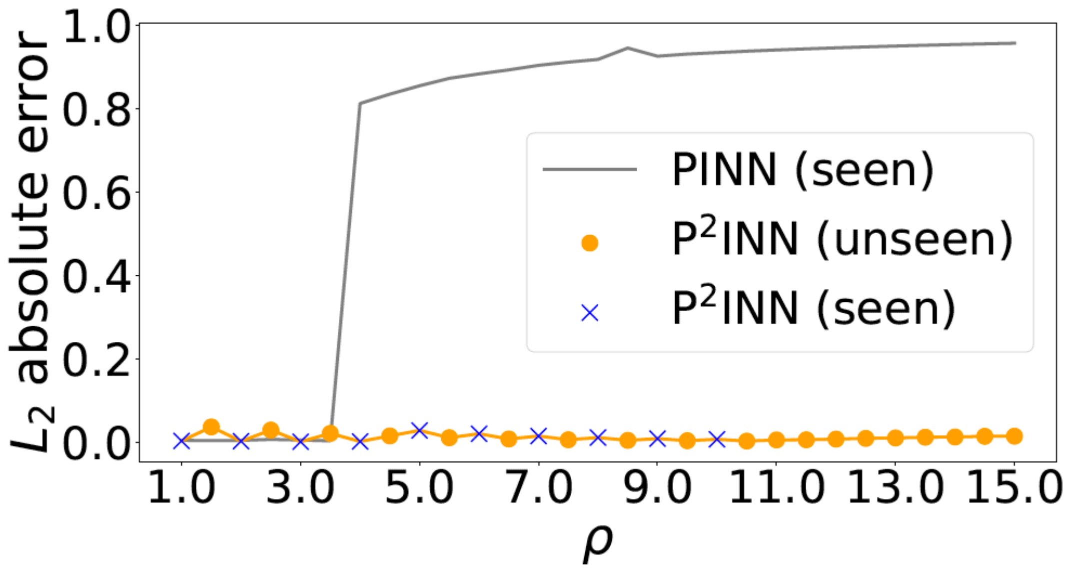
(d) Exact (e) PINN (f) P^2INN

Parameterized Physics-informed neural networks for Parameterized PDEs, **ICML 2024 (Oral)**

Experimental Results

Addressing W2: A single PINN can learn only one governing equation

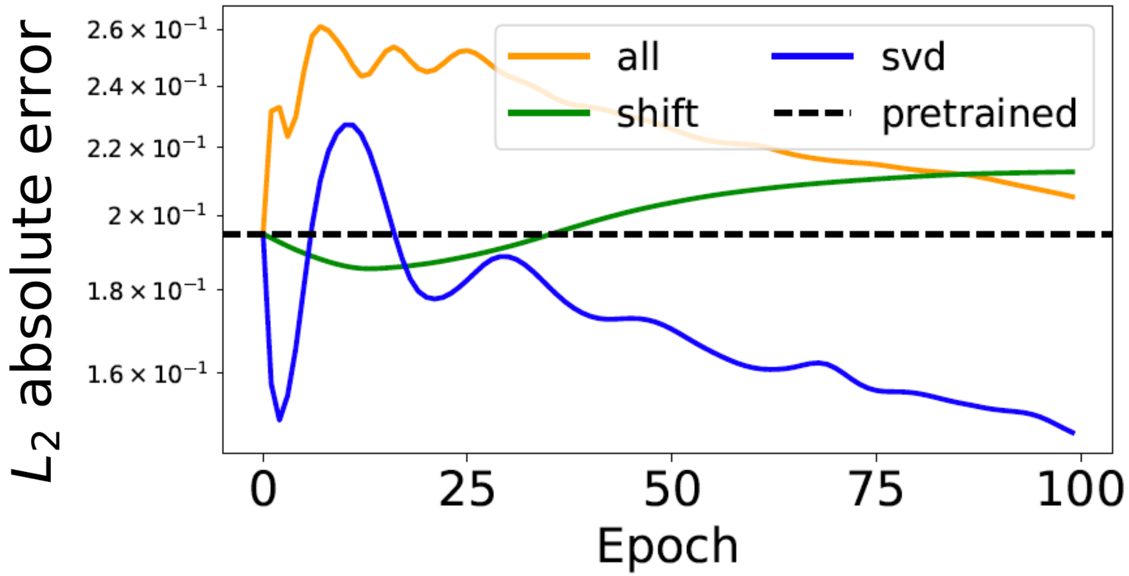
Real-time multi-query scenarios



<Reaction equation>

Fine-tuning phase

(unseen PDE parameter)



<Convection equation (β=8)>

Pretrained model is trained using convection equations with β=1~5

Parameterized Physics-informed neural networks for Parameterized PDEs, **ICML 2024 (Oral)**

Conclusion

- We design a novel neural network architecture for solving parameterized PDEs, P2INNs, which significantly improves the performance of PINNs overcoming the well-known weaknesses.

- We demonstrate that P2INNs can learn all benchmark PDEs in a single training run and significantly outperform existing PINN methods in prediction accuracy.

Paper

Presenter : Woojin Cho
Email: snowmoon@yonsei.ac.kr

Poster Presentation Schedule: Thu, July 25, 11:30 am - 1:00 pm

Reference

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equation, **Journal of Computational Physics 2019**

Characterizing possible failure modes in physics-informed neural networks, **NeurIPS 2021**