Physics-Informed Neural Network in Groundwater Inverse Modeling

Quan Guo

Advisor: Prof. Jian Luo

Civil and Environmental Engineering
Georgia Institute of Technology

October 2022

Georgia Tech

Outline

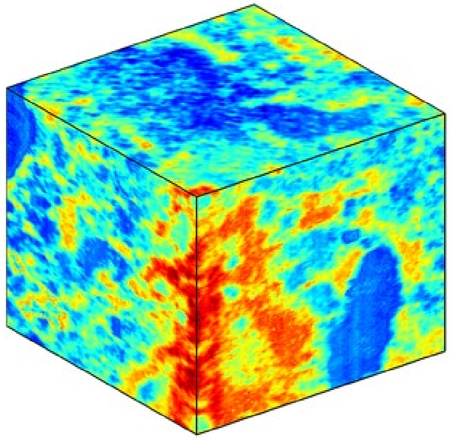
- Introduction & Motivation
- Background
- HT-PINN for Groundwater Modeling
- Numerical Experiment & Results
- Model Investigation & Discussion
- Future Work

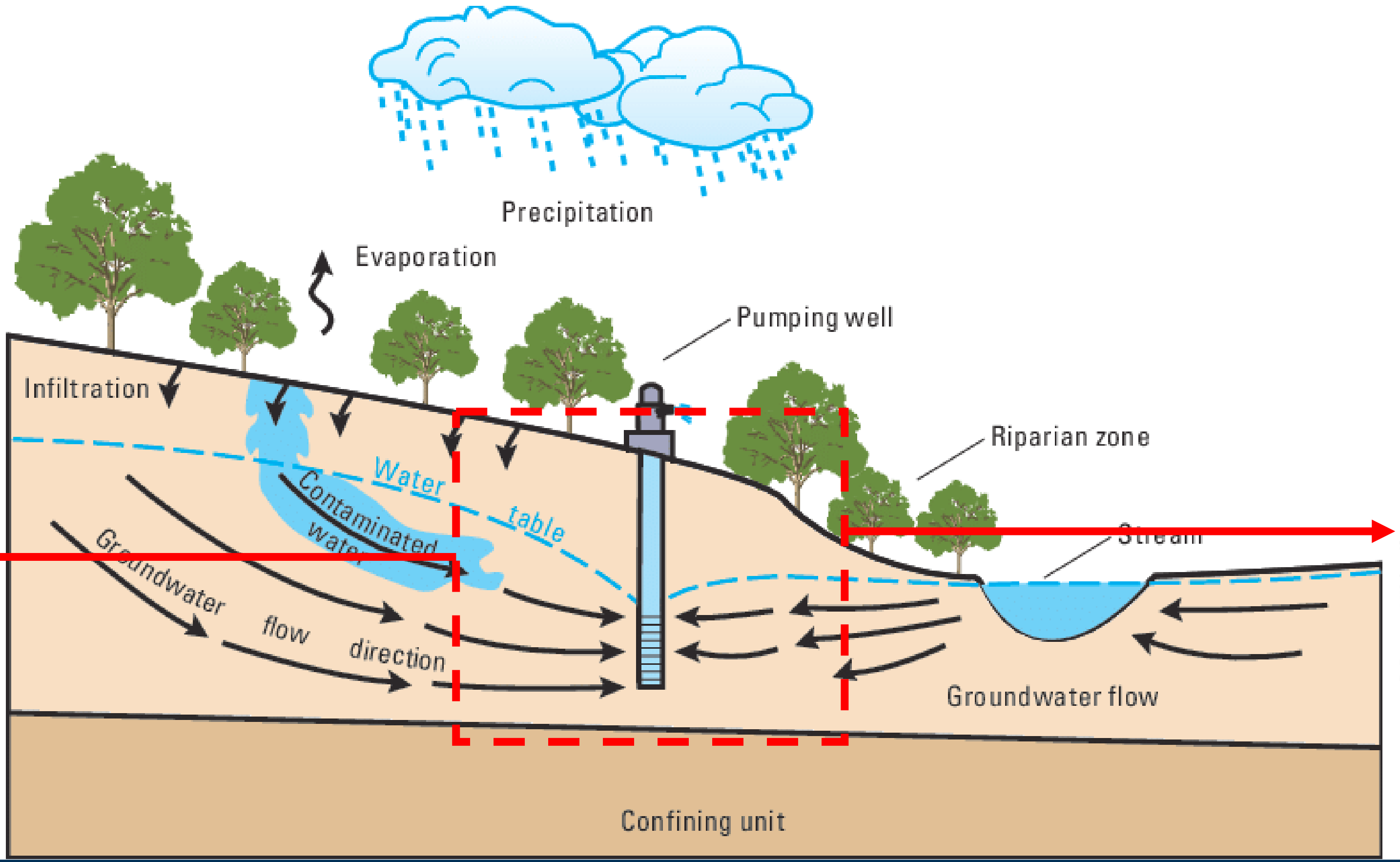
Note: HT-PINN abbreviates Hydraulic Tomography-Physics Informed Neural Network

Groundwater

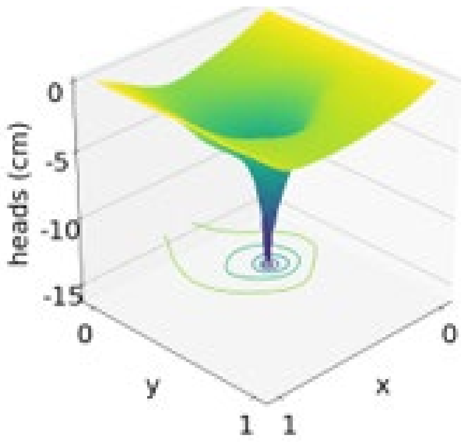
Groundwater (GW) plays an important role in natural water cycle.

Porous media





Water table



Introduction & Motivation

GW flow simulation

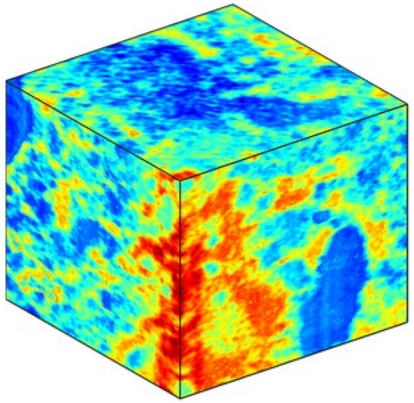
GW flow simulation (GWFS) solves for hydraulic heads with:

- Governing equations
- Initial & boundary conditions
- Hydrogeological parameters

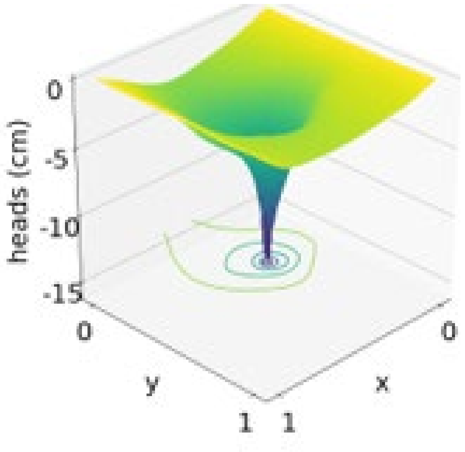
$$S_s \frac{\partial h}{\partial t} = -\nabla \cdot \mathbf{q} + Q \qquad \text{Mass conservation}$$

$$\mathbf{q} = T \nabla h \qquad \text{Darcy's Law}$$

S_s – specific storage; T – hydraulic transmissivity
h – hydraulic head; $\mathbf{q}$ – flux; Q – source/sink



forward problem



Forward problem is solving parameterized PDEs,
Forward model is numerical method, e.g., finite element method (FEM)

Field dimension: n
Complexity: $Q(n^2)$

Introduction & Motivation

GW inverse problem

Characteristics of hydrogeological parameters:

- Large-dimensional – 10^6 unknowns
- Spatially distributed – geostatistics
- Expensive to measure at field sites

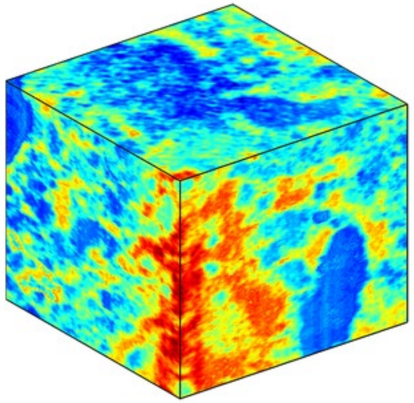
Inverse problem is estimating parameters in PDEs,

- **non-linear** without analytical solution
- **ill-posed** with infinite solutions

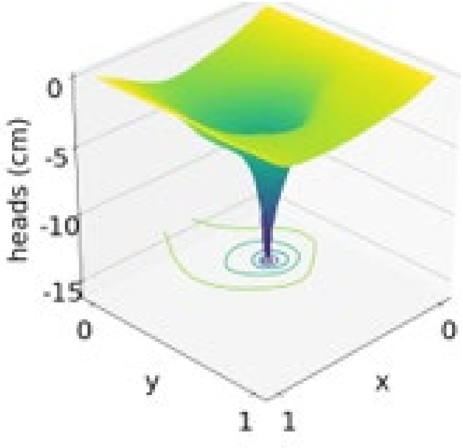
Solve with **gradient-based iterative method** and **regularization.**

During iteration, FEM solver is run many times to determine Jacobian matrix

Field dimension: n
Complexity: $Q(n^3)$



inverse problem



Introduction & Motivation

Surrogate model

PDE constraints:

Mass conservation → $S_s \frac{\partial h}{\partial t} = -\nabla \cdot \mathbf{q} + Q$

Darcy's Law → $\mathbf{q} = T \nabla h$

governs → $T(x^*, y^*), h(x^*, y^*)$

PDE residual:

$$\mathcal{L}(T^*, h^*; x^*, y^*) = \phi_n\left(T^*, h^*, \frac{\partial T^*}{\partial x^*}, \frac{\partial T^*}{\partial y^*}, \frac{\partial h^*}{\partial x^*}, \frac{\partial h^*}{\partial y^*}, \frac{\partial^2 T^*}{\partial x^{*2}}, \ldots, \frac{\partial^n h^*}{\partial y^{*n}}\right) = 0$$

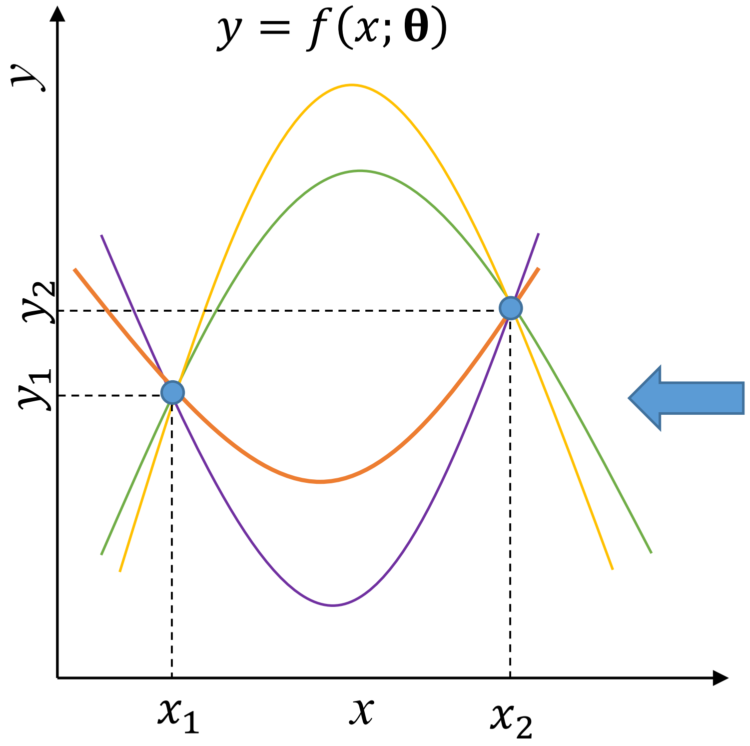
$TNN(x^*, y^*) = T^*$

$NN(x^*, y^*) = h^*$

minimize → $||\mathcal{L}(T^*, h^*; x^*, y^*)||$

Background

Regression of ill conditions



Assumption: f is second order polynomial with three degrees of freedom, $\boldsymbol{\theta} = \{a, b, c\}$, i.e.,

$$y = ax^2 + bx + c$$

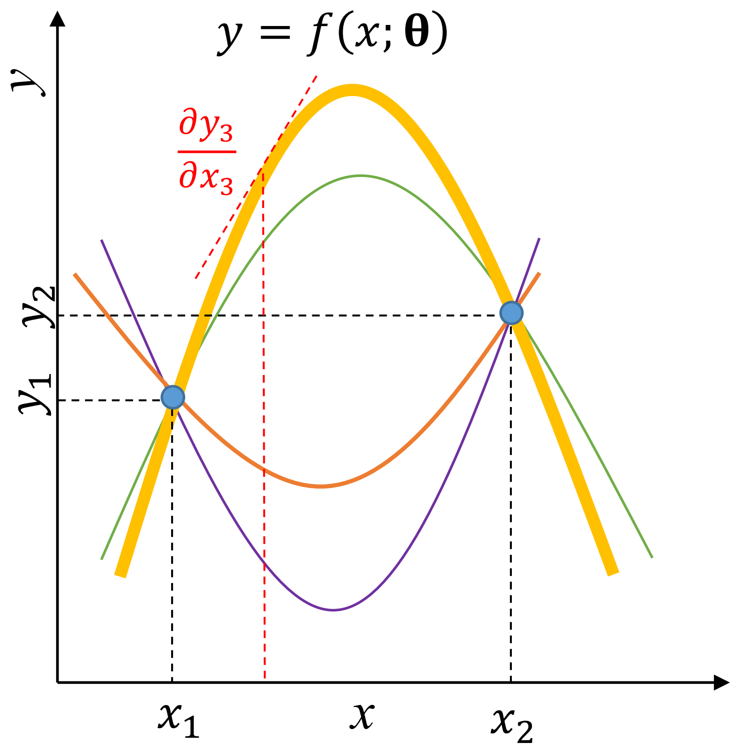
Data:

$$y_1 = f(x_1) \approx ax_1^2 + bx_1 + c$$
$$y_2 = f(x_2) \approx ax_2^2 + bx_2 + c$$

Problem is ill-posed, infinite solutions!

Background

Regularization on derivatives



Assumption: f is second order polynomial with three degree of freedom, $\boldsymbol{\theta} = \{a, b, c\}$, i.e.,

$$y = ax^2 + bx + c$$

Data:

$$y_1 = f(x_1) \approx ax_1^2 + bx_1 + c$$
$$y_2 = f(x_2) \approx ax_2^2 + bx_2 + c$$

Regularization:

$$\frac{\partial y_3}{\partial x_3} = f'(x_3) \approx 2ax_3^2 + b$$

Ideally, resulting in unique optimal solution

HT-PINN for GW Modeling

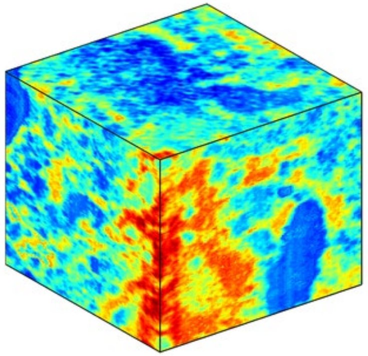
Physical constraints in groundwater

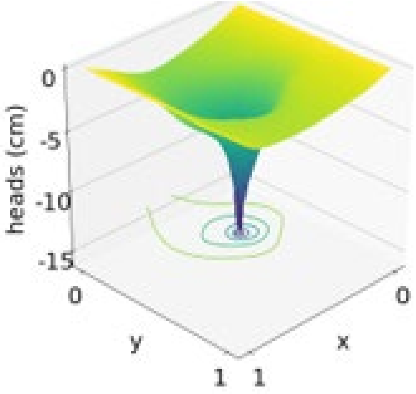
Pumping test on a confined, isotropic, heterogeneous aquifer (2D domain):

$$S_s \frac{\partial h}{\partial t} = -\nabla \cdot \mathbf{q} + Q \qquad \text{Mass conservation}$$

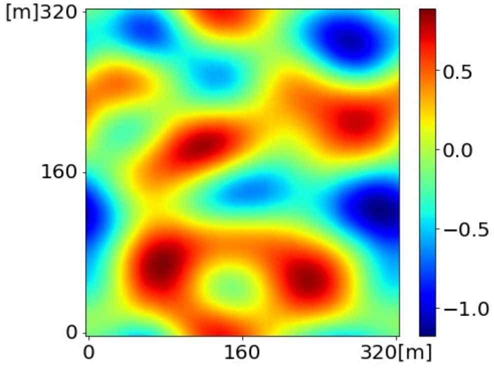
$$\mathbf{q} = T \nabla h \qquad \text{Darcy's Law}$$

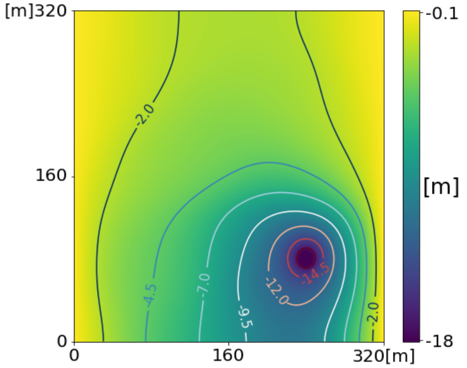
S_s − specific storage; T − hydraulic transmissivity
h − hydraulic head; $\mathbf{q}$ − flux; Q − source/sink





PLAN View





PDE for non-pumping grid	$S_s \frac{\partial h(x_e,y_e,t_e)}{\partial t} - \nabla \cdot [T(x_e,y_e)\nabla h(x_e,y_e,t_e)] = 0,$	$(x_e,y_e) \in \Omega, t_e \in (0,\mathrm{T}]$
PDE for pumping grid	$S_s \frac{\partial h(x_p,y_p,t_p)}{\partial t} - \nabla \cdot [T(x_p,y_p)\nabla h(x_p,y_p,t_p)] = Q_p,$	$(x_p,y_p) \in \Omega, t_p \in (0,\mathrm{T}]$
Neumann Boundary Condition	$\mathbf{n} \cdot \nabla h(x_N,y_N,t_N) = q_N,$	$(x_N,y_N) \in \Gamma_N, t_N \in (0,\mathrm{T}]$
Dirichlet Boundary Condition	$h(x_D,y_D,t_D) = h_D,$	$(x_D,y_D) \in \Gamma_D, t_D \in (0,\mathrm{T}]$
Initial Condition	$h(x_{init},y_{init},0) = h_{init},$	$(x_{init},y_{init}) \in \Omega$

HT-PINN for GW Modeling

Georgia Tech

Transient Forward & Inverse Networks

Assumption:

$$h(x,y,t) \approx NN(x,y,t)$$
$$T(x,y) \approx TNN(x,y)$$

Data (reference):

Monitored hydraulic heads: $NN(x_m, y_m) = h_m$

Measurements of transmissivity: $TNN(x_T, y_T) = T(x_T, y_T)$

Regularization (collocation):

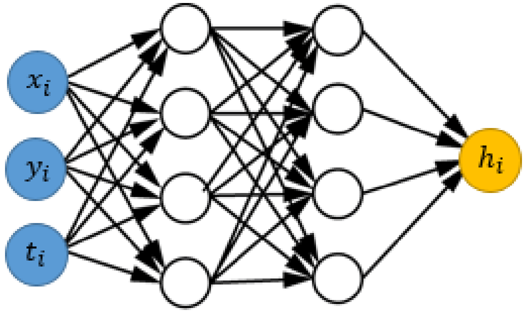
$$S_s \frac{\partial NN(x_e,y_e,t_e)}{\partial t} - \nabla \cdot [TNN(x_e,y_e) \nabla NN(x_e,y_e,t_e)] = 0$$

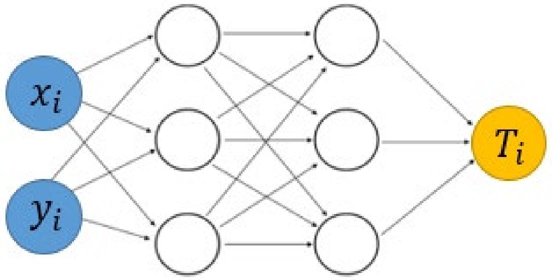
$$S_s \frac{\partial NN(x_p,y_p,t_p)}{\partial t} - \nabla \cdot [TNN(x_p,y_p) \nabla NN(x_p,y_p,t_p)] = Q_p$$

$$\mathbf{n} \cdot \nabla NN(x_N, y_N, t_N) = q_N$$

$$NN(x_D, y_D, t_D) = h_D$$

$$NN(x_{init}, y_{init}, 0) = h_{init}$$





Network Architecture

	Transient Forward	**Inverse**
Input variables	Spatial & temporal (x, y, t)	Spatial (x, y)
Output variables	Hydraulic heads (h)	Transmissivity (T)
Number of layers	7	
Hidden dimensions	50	
Activation function	Hyperbolic $(tanh)$	
Output layer type	Linear	

HT-PINN for GW Modeling

Georgia Tech

Batch Training

Training data for HT-PINN include:

- Reference data: direct/indirect measurements
- Collocation data: pumping/non-pumping, B.C., I.C.

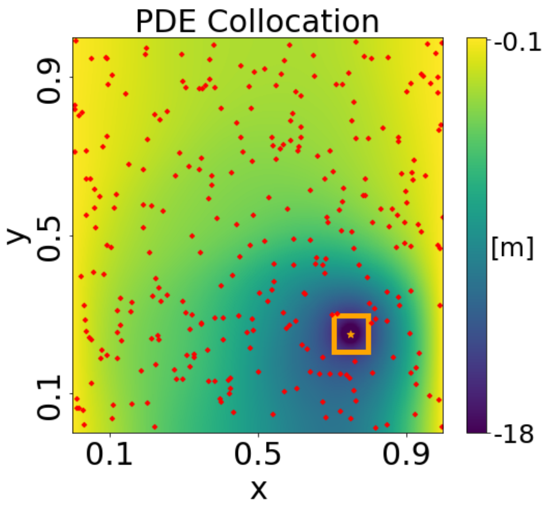
Batch training based on collocation data:

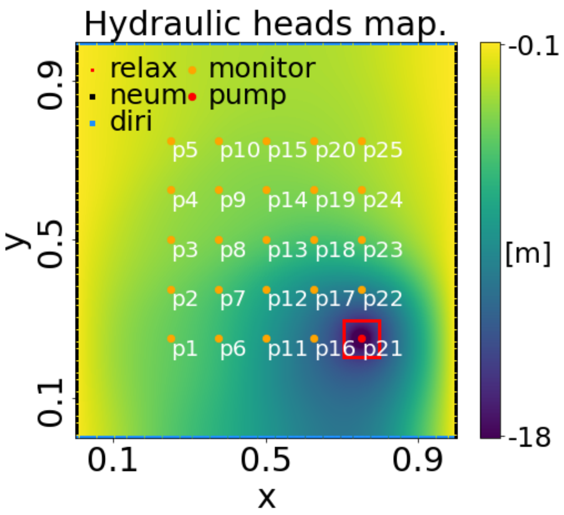
- One batch has 300 randomly selected non-pumping grids
- One time step has 10 batches of data, transient has 10 times steps
- 5 pumping tests in HT

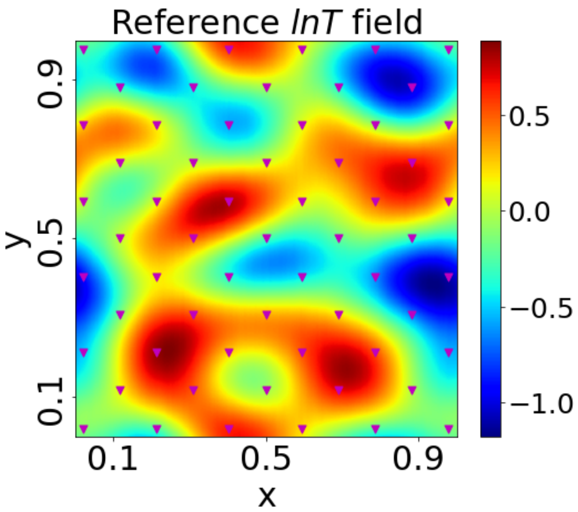
Number and property of different grids in each batch of training data for HT-PINN

Type of points	Pumping	Time	Batch	Number
Pumping (x_p, y_p)	Invariant	Invariant	Invariant	1
Neumann (x_N, y_N)	Invariant	Invariant	Invariant	64×2
Dirichlet (x_D, y_D)	Invariant	Invariant	Invariant	64×2
Direct (x_T, y_T)	Invariant	Invariant	Invariant	61
Initial (x_{init}, y_{init})	Variant	Invariant	Invariant	25
Monitored (x_m, y_m, t_m)	Variant	Variant	Invariant	24
Non-pumping (x_e, y_e, t_e)	Variant	Variant	Variant	300

Total data: 667





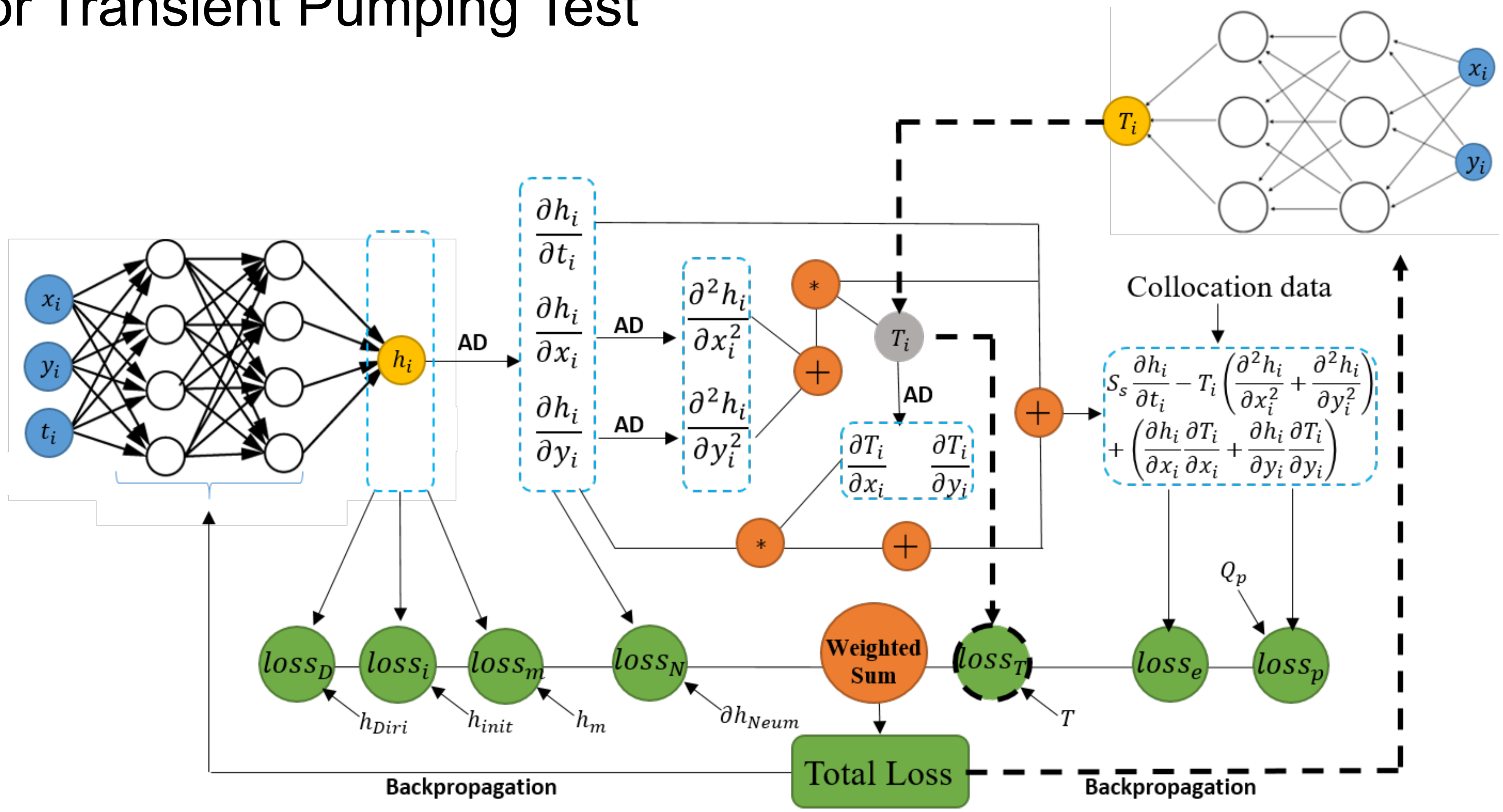


Data Batch

Non-pumping	(x_e, y_e, t_e),	h_e
Pumping	(x_p, y_p),	Q_p
Neumann	(x_N, y_N),	0
Dirichlet	(x_D, y_D),	h_D
Direct	(x_T, y_T),	T
Initial	(x_{init}, y_{init}),	h_{init}
Monitored	(x_m, y_m, t_m),	h_m

HT-PINN for GW Modeling

PINN for Transient Pumping Test

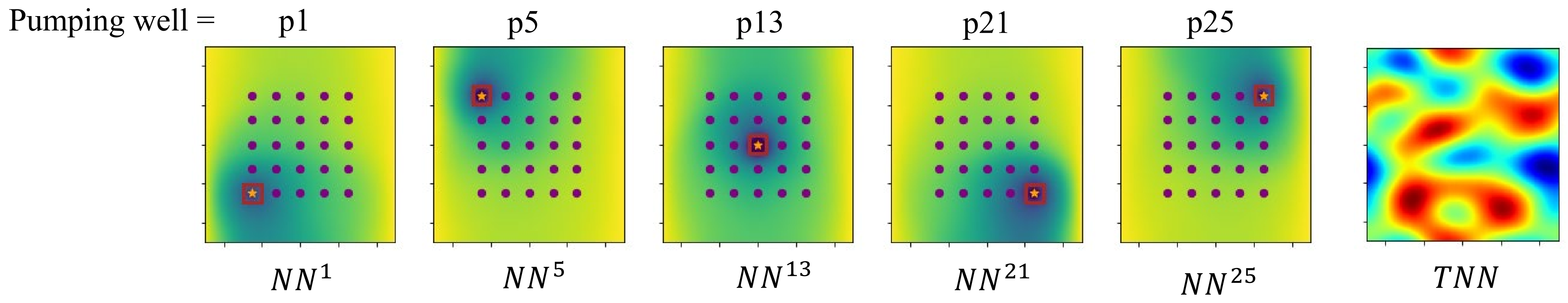


HT-PINN for GW Modeling

HT-PINN Composition

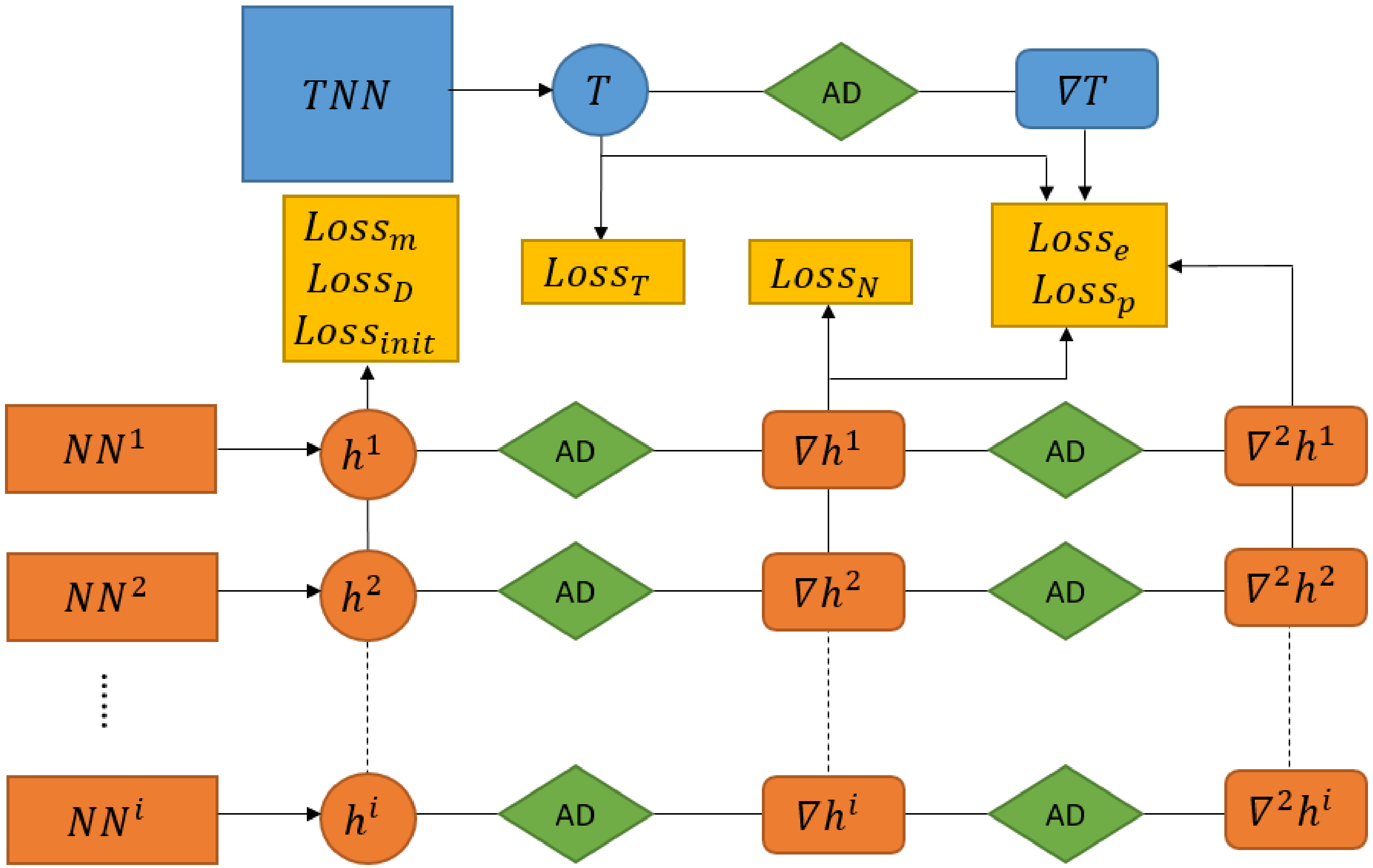
Each pumping test has a forward network NN^i (labeled by pumping well).

Only one inverse network TNN in HT.



HT-PINN for GW Modeling

HT-PINN flowchart



Losses from each pumping test are summed:

Total Loss of HT-PINN:

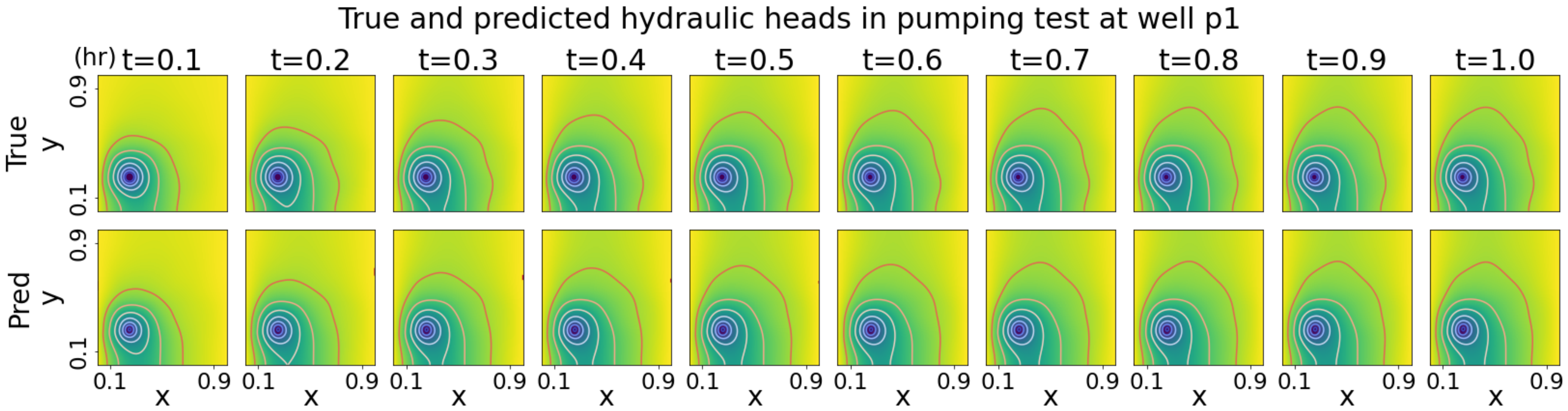
$$\ell_{\text{HT-PINN}} = \sum \ell_{NN^i} + \ell_{TNN}$$

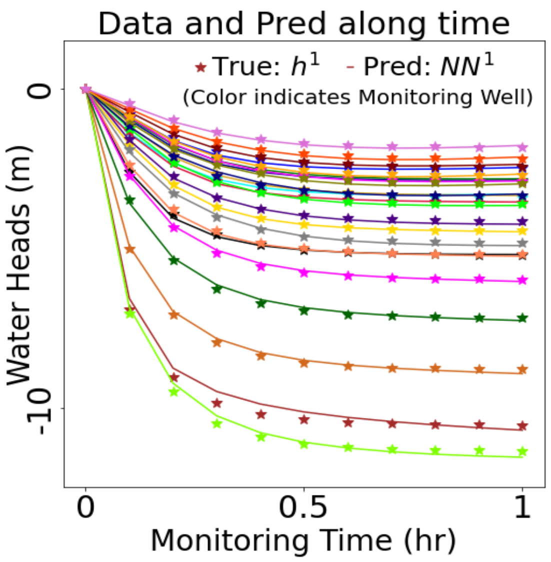
Best estimate:

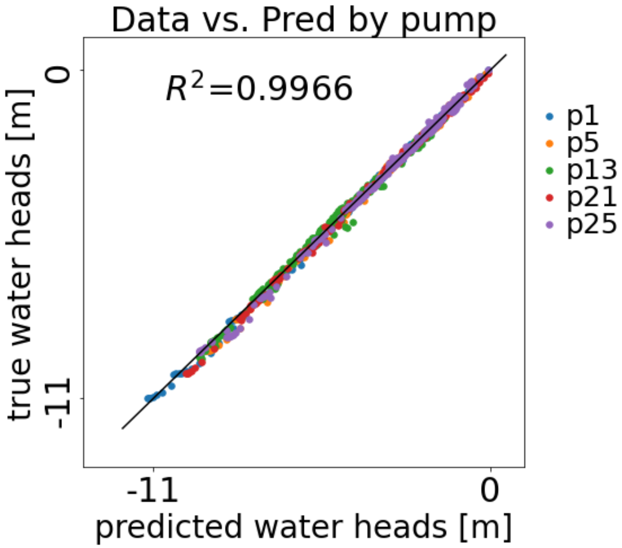
$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(\sum \ell_{NN^i} + \ell_{TNN}\right)$$

Result

Transient Forward Results





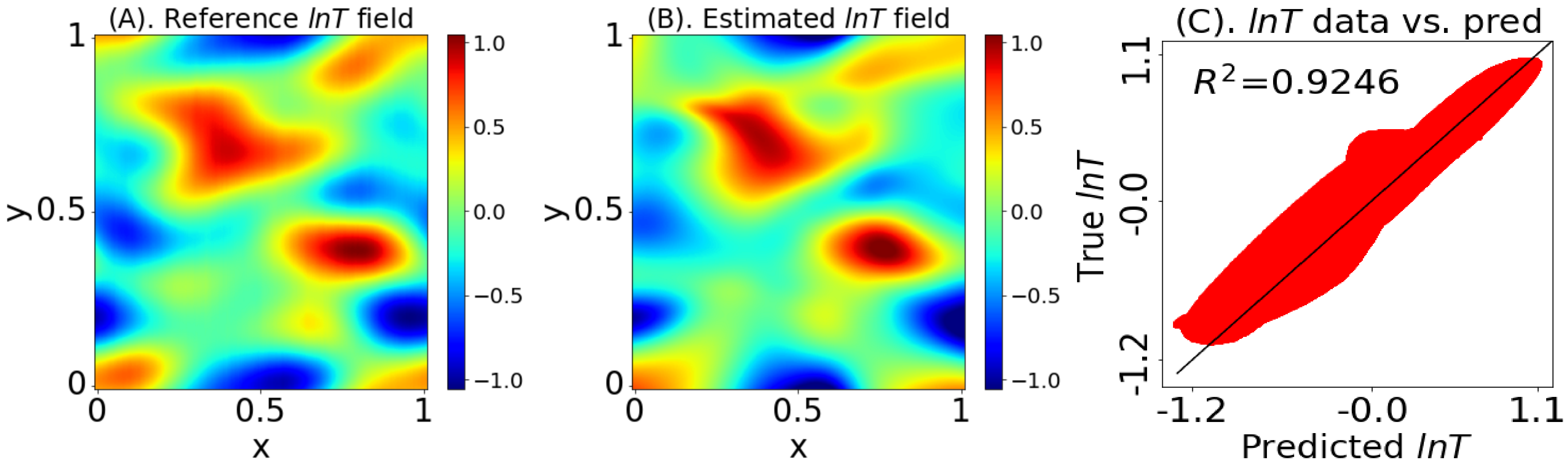


Average relative residual $\epsilon_{NN_t^i}$ average on all time steps $t = 0.1 - 1.0$

- P1: 6.14%
- P5: 6.26%
- P13: 6.23%
- P21: 6.58%
- P25: 6.53%

Result

Transient Inverse Results



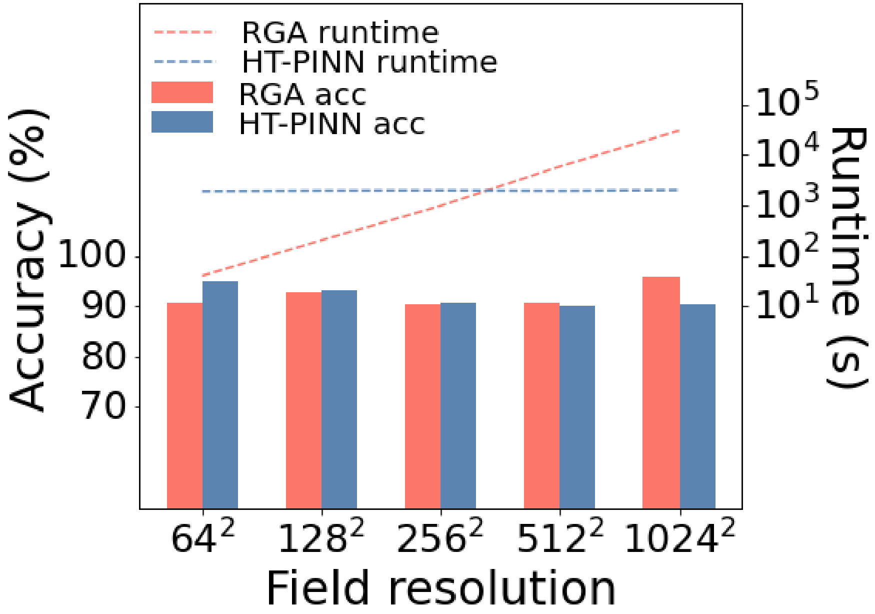
The relative residual ϵ_{TNN} is 10.32%, and the accuracy is 94.93%.

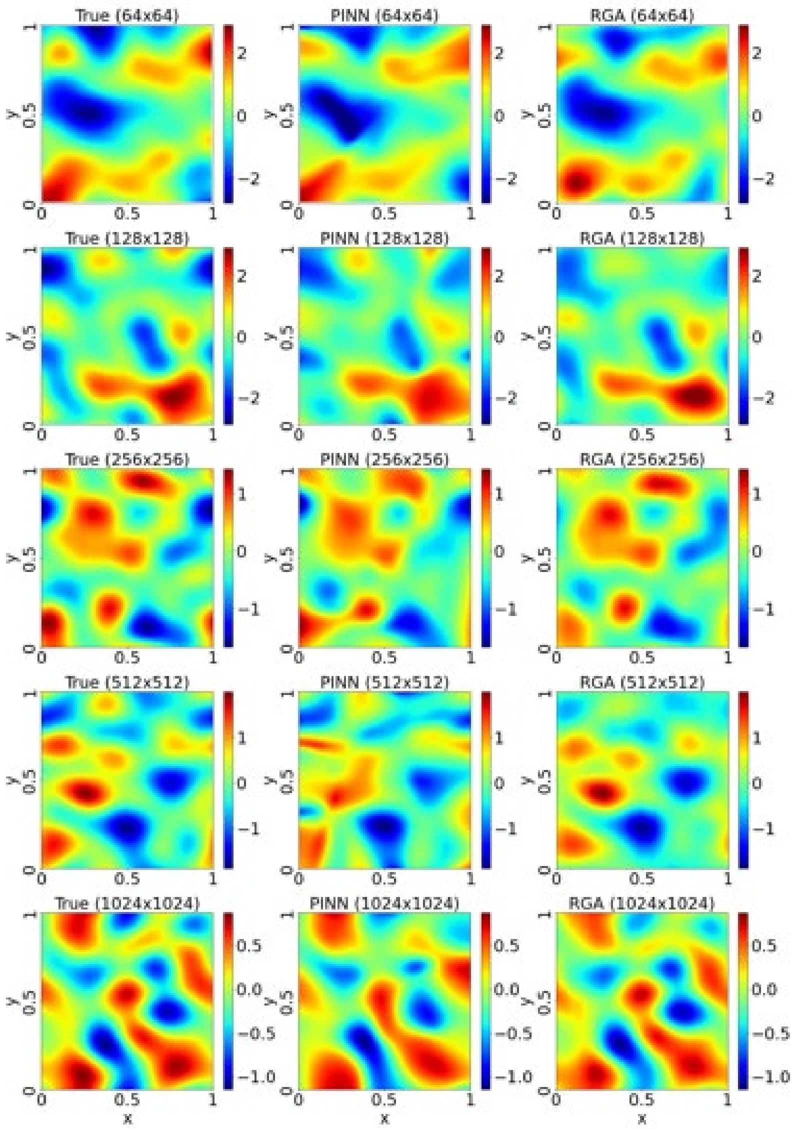
Training time is about 9.5 hours.

Model Investigation

Model Scalability

Model	RGA	HT-PINN
Accuracy	> 90%	> 90%
N_h	24×5	24×5
N_{lnT}	0	61
Covariance	Yes	No
Scalability	Linear	Constant





Discussion

Georgia Tech

Model Comparison & Future Work

Model	HT-PINN	GA Inverse model
Type	Pointwise approximator	Gradient-based approach
Regularization	Physical constraints (PDE)	Geostatistical assumption
Pros	Scalable	Do not require direct measurements
Cons	Require direct measurements	Not scalable due to matrix computation

Current state:

Suitable for inferring high-resolution, smooth field with direct measurements available.

In future:

1. Reduce the required data of HT-PINN through incorporating other constraints such as geostatistics.
2. Generalize HT-PINN model for inverse modeling of non-Gaussian fields.

Q & A

Many Thanks!

Appreciate any questions

Reference

Bandai, T., & Ghezzehei, T. A. (2021), Physics-Informed Neural Networks With Monotonicity Constraints for Richardson-Richards Equation: Estimation of Constitutive Relationships and Soil Water Flux Density From Volumetric Water Content Measurements, *Water Resources Research*, *57*(2), e2020WR027642.
Bottou, L., & Bousquet, O. (2008), The tradeoffs of large scale learning, *Adv. Neur. In*, *20*, 161-168.
Broyden, C. G. (1965), A class of methods for solving nonlinear simultaneous equations, *Math. Comput.*, *19*(92), 577-593.
Cardiff, M., Barrash, W., & Kitanidis, P. K. (2013), Hydraulic conductivity imaging from 3-D transient hydraulic tomography at several pumping/observation densities, *Water Resour. Res.*, *49*(11), 7311-7326.
Carrera, J., & Neuman, S. P. (1986), Estimation of Aquifer Parameters Under Transient and Steady State Conditions: 1. Maximum Likelihood Method Incorporating Prior Information, *Water Resources Research*, *22*(2), 199-210.
Cheng, S., Cheng, L., Qin, S., Zhang, L., Liu, P., Liu, L., Xu, Z., & Wang, Q. (2022), Improved Understanding of How Catchment Properties Control Hydrological Partitioning Through Machine Learning, *Water Resources Research*, *58*(4), e2021WR031412.
Cho, E., Jacobs, J. M., Jia, X., & Kraatz, S. (2019), Identifying Subsurface Drainage using Satellite Big Data and Machine Learning via Google Earth Engine, *Water Resources Research*, *55*(10), 8028-8045.
Fienen, M. N., Clemo, T., & Kitanidis, P. K. (2008), An interactive Bayesian geostatistical inverse protocol for hydraulic tomography, *Water Resources Research*, *44*(12).
Goldberg, Y. (2016), A primer on neural network models for natural language processing, *J. Artif. Int. Res.*, *57*(1), 345–420.
Goldstein, E. B., & Coco, G. (2014), A machine learning approach for the prediction of settling velocity, *Water Resources Research*, *50*(4), 3595-3601.
Griewank, A. (2003), A mathematical view of automatic differentiation, *Acta Numer.*, *12*, 321-398.
He, Q., & Tartakovsky, A. M. (2021), Physics-Informed Neural Network Method for Forward and Backward Advection-Dispersion Equations, *Water Resources Research*, *57*(7), e2020WR029479.
He, Q., Barajas-Solano, D., Tartakovsky, G., & Tartakovsky, A. M. (2020a), Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, *Advances in Water Resources*, *141*, 103610.
He, Q., Barajas-Solano, D., Tartakovsky, G., & Tartakovsky, A. M. (2020b), Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, *Adv. Water Resour.*, *141*, 103610.
Hoffer, E., Hubara, I., & Soudry, D. (2018), Train longer, generalize better: closing the generalization gap in large batch training of neural networks, *arXiv*.
Hofmann, T. (2004), Latent semantic models for collaborative filtering, *ACM Trans. Inf. Syst.*, *22*(1), 89-115.
Ioffe, S., & Szegedy, C. (2015), Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in *Proceedings of the 32nd International Conference on Machine Learning*, edited by B. Francis and B. David, pp. 448--456, PMLR, Proceedings of Machine Learning Research.
Jacob, Dillavou, S., Stern, M., Andrea, & Douglas (2022), Learning Without a Global Clock: Asynchronous Learning in a Physics-Driven Learning Network, *arXiv*.
Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020), Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, *Comput. Methods Appl. Mech. Eng.*, *365*, 113028.
Kang, P. K., Lee, J., Fu, X., Lee, S., Kitanidis, P. K., & Juanes, R. (2017), Improved characterization of heterogeneous permeability in saline aquifers from transient pressure data during freshwater injection, *Water Resour. Res.*, *53*(5), 4444-4458.

Reference

Karniadakis, G. (2019), VPINNs: Variational physics-informed neural networks for solving partial differential equations, *arXiv*.

Kharazmi, E., Zhang, Z., & Karniadakis, G. E. M. (2021), hp-VPINNs: Variational physics-informed neural networks with domain decomposition, *Comput. Methods Appl. Mech. Eng.*, *374*, 113547.

Kingma, D. P., & Ba, J. (2017), Adam: A Method for Stochastic Optimization, *arXiv*.

Kitanidis, P. K. (1995), Quasi-Linear Geostatistical Theory for Inversing, *Water Resour. Res.*, *31*(10), 2411-2419.

Kitanidis, P. K., & Vomvoris, E. G. (1983), A geostatistical approach to the inverse problem in groundwater modeling (steady state) and one-dimensional simulations, *Water Resources Research*, *19*(3), 677-690.

Klein, O., Cirpka, O. A., Bastian, P., & Ippisch, O. (2017), Efficient geostatistical inversion of transient groundwater flow using preconditioned nonlinear conjugate gradients, *Adv. Water Resour.*, *102*, 161-177.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012), ImageNet classification with deep convolutional neural networks, paper presented at Adv. Neur. In.

Laloy, E., Hérault, R., Lee, J., Jacques, D., & Linde, N. (2017), Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network, *Advances in water resources*, *110*, 387-405.

LeCun, Y., Bengio, Y., & Hinton, G. (2015), Deep learning, *Nature*, *521*(7553), 436-444.

Lee, J., & Kitanidis, P. (2014), Large scale hydraulic tomography and joint inversion of head and tracer data using the Principal Component Geostatistical Approach (PCGA), *Water Resour. Res.*, *50*, 5410-5427.

Li, J., & Tartakovsky, A. M. (2022), Physics-informed Karhunen-Loéve and neural network approximations for solving inverse differential equation problems, *J. Comput. Phys.*, *462*, 111230.

Li, J., Wang, Z., Wu, X., Xu, C.-Y., Guo, S., Chen, X., & Zhang, Z. (2021), Robust Meteorological Drought Prediction Using Antecedent SST Fluctuations and Machine Learning, *Water Resources Research*, *57*(8), e2020WR029413.

Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2021), Efficient mini-batch training for stochastic optimization, ACM, 2014.

Liu, X., & Kitanidis, P. (2011), Large-scale inverse modeling with an application in hydraulic tomography, *Water Resour. Res.*, *47*(2).

Liu, Y., Sun, W., & Durlofsky, L. (2019), A Deep-Learning-Based Geological Parameterization for History Matching Complex Models, *Mathematical Geosciences*, *51*.

Masters, D., & Luschi, C. (2018), Revisiting Small Batch Training for Deep Neural Networks, *arXiv*.

McCandlish, S., Kaplan, J., Amodei, D., & OpenAi (2018), An Empirical Model of Large-Batch Training, *arXiv*.

Meyer, D., Grimmond, S., Dueben, P., Hogan, R., & van Reeuwijk, M. (2022), Machine Learning Emulation of Urban Land Surface Processes, *Journal of Advances in Modeling Earth Systems*, *14*(3), e2021MS002744.

Neuman, S. P. (1980), A statistical approach to the inverse problem of aquifer hydrology: 3. Improved solution method and added perspective, *Water Resources Research*, *16*(2), 331-346.

Nitish, Mudigere, D., Nocedal, J., Smelyanskiy, M., & Ping (2017), On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, *arXiv*.

Nowak, W., Tenkleve, S., & Cirpka, O. A. (2003), Efficient computation of linearized cross-covariance and auto-covariance matrices of interdependent quantities, *Math. Geol.*, *35*(1), 53-66.

Reference

Pang, G., D'Elia, M., Parks, M., & Karniadakis, G. (2020), nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications, *J. Comput. Phys.*, *422*, 109760.
Rahmati, O., Naghibi, S. A., Shahabi, H., Bui, D. T., Pradhan, B., Azareh, A., Rafiei-Sardooi, E., Samani, A. N., & Melesse, A. M. (2018), Groundwater spring potential modelling: Comprising the capability and robustness of three different modeling approaches, *Journal of Hydrology*, *565*, 248-261.
Raissi, M., Perdikaris, P., & Karniadakis, G. (2017a), Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations, *arXiv*, *abs/1711.10566*.
Raissi, M., Perdikaris, P., & Karniadakis, G. (2017b), Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, *arXiv*, *abs/1711.10561*.
Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019), Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, *378*, 686-707.
Ren, P., Rao, C., Liu, Y., Wang, J.-X., & Sun, H. (2021), *PhyCRNet: Physics-informed Convolutional-Recurrent Network for Solving Spatiotemporal PDEs*.
Saibaba, A. K., Ambikasaran, S., Li, J. Y., Kitanidis, P. K., & Darve, E. F. (2012), Application of hierarchical matrices to linear inverse problems in geostatistics, *Oil. Gas. Sci. Technol*, *67*(5), 857-875.
Shahnas, M. H., Yuen, D. A., & Pysklywec, R. N. (2018), Inverse Problems in Geodynamics Using Machine Learning Algorithms, *J. Geophys. Res.: Solid Earth*, *123*(1), 296-310.
Snodgrass, M. F., & Kitanidis, P. K. (1998), Transmissivity identification through multi-directional aquifer stimulation, *Stochastic Hydrology and Hydraulics*, *12*(5), 299-316.
Sun, A. Y., Scanlon, B. R., Save, H., & Rateb, A. (2021), Reconstruction of GRACE Total Water Storage Through Automated Machine Learning, *Water Resources Research*, *57*(2), e2020WR028666.
Tahmasebi, P. (2017), HYPPS: A hybrid geostatistical modeling algorithm for subsurface modeling, *Water Resources Research*, *53*(7), 5980-5997.
Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., & Barajas-Solano, D. (2020), Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems, *Water Resour. Res., 56*(5), e2019WR026731, doi: https://doi.org/10.1029/2019WR026731.
Vo, H. X., & Durlofsky, L. J. (2014), A New Differentiable Parameterization Based on Principal Component Analysis for the Low-Dimensional Representation of Complex Geological Models, *Mathematical Geosciences*, *46*(7), 775-813.
Wang, N., Chang, H., & Zhang, D. (2021a), Deep-Learning-Based Inverse Modeling Approaches: A Subsurface Flow Example, *J. Geophys. Res.: Solid Earth*, *126*(2), e2020JB020549.
Wang, N., Chang, H., & Zhang, D. (2021b), Efficient uncertainty quantification for dynamic subsurface flow with surrogate by Theory-guided Neural Network, *Comput. Methods Appl. Mech. Eng.*, *373*, 113492.
Wang, N., Zhang, D., Chang, H., & Li, H. (2020), Deep learning of subsurface flow via theory-guided neural network, *J. Hydrol.*, *584*, 124700.
Wilson, D. R., & Martinez, T. R. (2003), The general inefficiency of batch training for gradient descent learning, *Neural Networks*, *16*(10), 1429-1451.

Reference

Xu, R., Wang, N., & Zhang, D. (2021a), Solution of diffusivity equations with local sources/sinks and surrogate modeling using weak form Theory-guided Neural Network, *Adv. Water Resour.*, *153*, 103941.

Xu, R., Zhang, D., Rong, M., & Wang, N. (2021b), Weak form theory-guided neural network (TgNN-wf) for deep learning of subsurface single- and two-phase flow, *J. Comput. Phys.*, *436*, 110318.

Yan, J., Jia, S., Lv, A., & Zhu, W. (2019), Water Resources Assessment of China's Transboundary River Basins Using a Machine Learning Approach, *Water Resources Research*, *55*(1), 632-655.

Yang, L., Zhang, D., & Karniadakis, G. E. (2020), Physics-Informed Generative Adversarial Networks for Stochastic Differential Equations, *SIAM J. Sci. Comput.*, *42*(1), A292-A317.

Yang, L., Meng, X., & Karniadakis, G. E. (2021), B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, *J. Comput. Phys.*, *425*, 109913.

Yang, Y., & Perdikaris, P. (2019), Adversarial Uncertainty Quantification in Physics-Informed Neural Networks, *J. Comput. Phys.*, *394*, 136-152.

Yeh, T., & Liu, S. (2000), Hydraulic tomography: Development of a new aquifer test method, *Water Resour. Res.*, *36*, 2095-2105.

Yeh, T., Jin, M., & Hanna, S. (1995), An Iterative Stochastic Inverse Method: Conditional Effective Transmissivity and Hydraulic Head Fields.

Yeh, T. C., & Lee, C. H. (2007), Time to change the way we collect and analyze data for aquifer characterization, *Ground water*, *45*(2), 116-118.

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., Lin, C., Li, X., & Qiu, G. Y. (2019), Physics-Constrained Machine Learning of Evapotranspiration, *Geophysical Research Letters*, *46*(24), 14496-14507.

Zhao, Y., & Luo, J. (2020), Reformulation of Bayesian Geostatistical Approach on Principal Components, *Water Resour. Res.*, *56*.

Zhao, Y., & Luo, J. (2021a), Bayesian inverse modeling of large-scale spatial fields on iteratively corrected principal components, *Adv. Water Resour.*, *151*, 103913.

Zhao, Y., & Luo, J. (2021b), A Quasi-Newton Reformulated Geostatistical Approach on Reduced Dimensions for Large-Dimensional Inverse Problems, *Water Resour. Res.*, *57*(1), e2020WR028399.

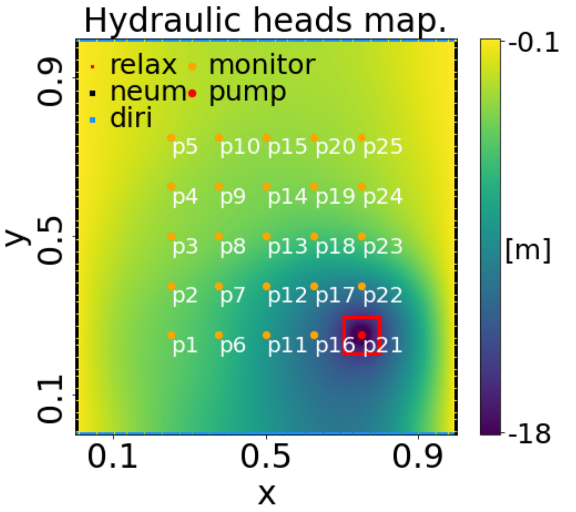
Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019), Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.*, *394*, 56-81.

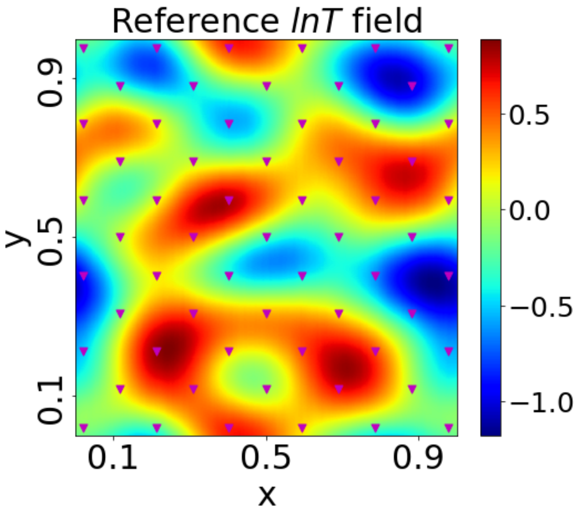
Appendix

Experimental Domain

Hydrogeological and geostatistical parameters for the hydraulic tomography experiment

Parameter	Values
Domain size, $L_x \times L_y$	320m × 320m
Grid spacing, $\Delta x \times \Delta y$	0.3125m × 0.3125m
Spatial resolution, $n_x \times n_y$	1024 × 1024
Transmissivity, T [m^2/hr]	
Geometric mean	0
Variance of ln$T$, σ^2_{lnT}	1
Correlation length, $\lambda_x \times \lambda_y$	64m × 48m
Left Boundary	h=0m
Right Boundary	h=0m
Initial Condition	h=0m
Pumping Time [hr]	1
Monitor Time Step [hr]	0.1
Pumping Rate [m^3/hr]	3.6

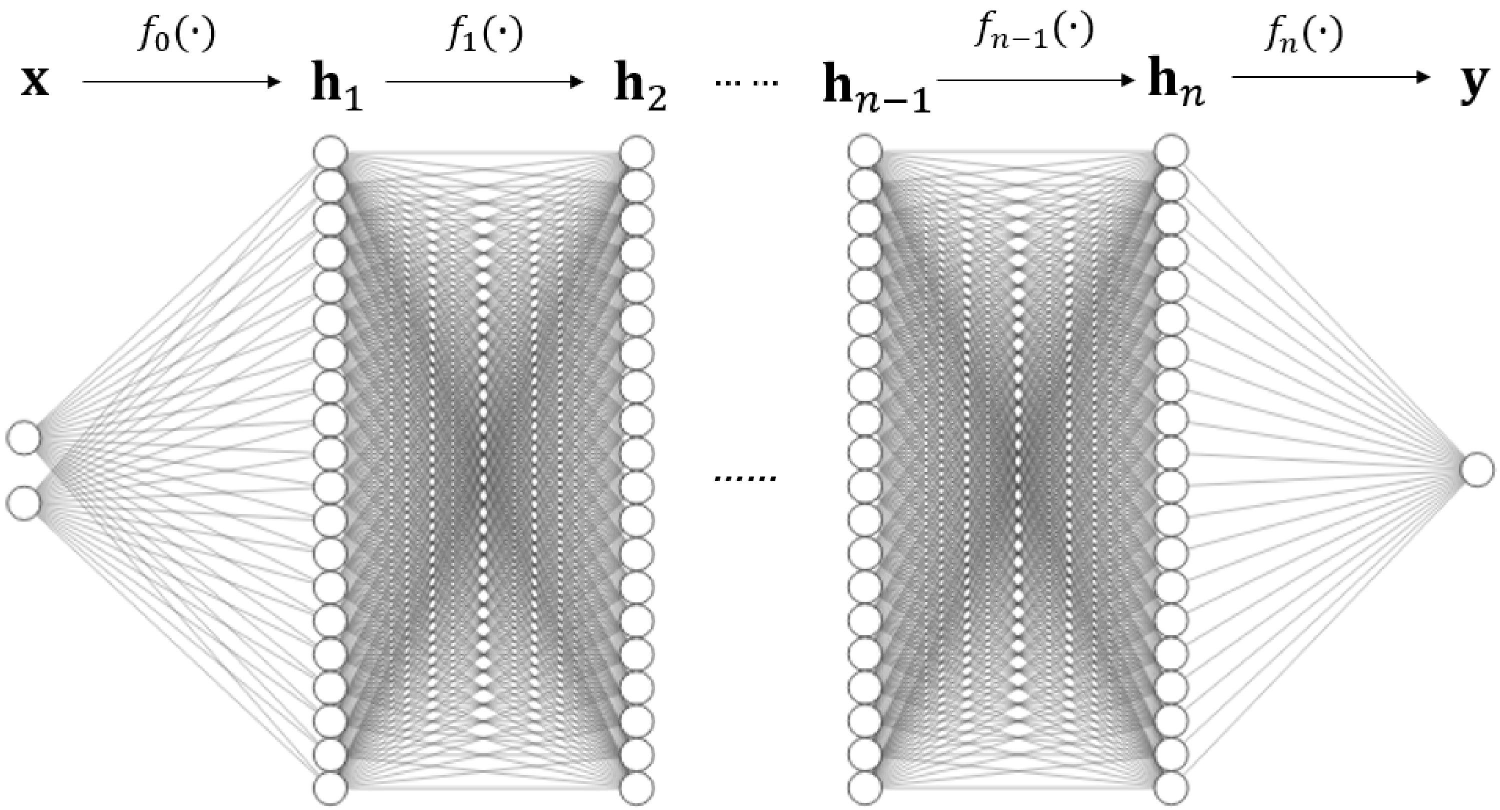




Appendix

Deep Neural Network

Deep neural network (DNN) is a typical Machine Learning (ML) or Deep Learning (DL) model



$$\mathbf{h}_1 = f_0(\mathbf{x}) = \sigma_0(\mathbf{W}_0\mathbf{x} + \mathbf{b}_0)$$
$$\mathbf{h}_2 = f_1(\mathbf{h}_1) = \sigma_1(\mathbf{W}_1\mathbf{h}_1 + \mathbf{b}_1)$$

……

$$\mathbf{h}_n = f_{n-1}(\mathbf{h}_{n-1}) = \sigma_{n-1}(\mathbf{W}_{n-1}\mathbf{h}_{n-1} + \mathbf{b}_{n-1})$$
$$\mathbf{y} = f_1(\mathbf{h}_n) = \sigma_n(\mathbf{W}_n\mathbf{h}_n + \mathbf{b}_n)$$
$$\downarrow$$
$$\mathbf{y} = f_n(f_{n-1}(\dots f_1(f_0(\mathbf{x}))))$$

$\sigma_i(\cdot)$ is chosen activation function;
$\mathbf{W}_i$ is learnable weight matrix;
$\mathbf{b}_i$ is learnable bias vector.

$\mathbf{x}$ is input variable vector; $\mathbf{h}_i$ is hidden feature vector; $\mathbf{y}$ is output variable vector

Appendix

DNN Training

DNN is purely data-driven, loss function is based on **data match:** $\ell(\mathbf{y}, \hat{\mathbf{y}})$

Loss function (Mean Squared Error):

$$\ell = \ell_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N_{data}} \sum_{p=1}^{N_{data}} \|\hat{\mathbf{y}}^p - \mathbf{y}^p\|$$

Learnable coefficients:

$$\boldsymbol{\theta} = \{\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_n; \mathbf{b}_2, \mathbf{b}_1, \dots, \mathbf{b}_i\}$$

Best estimate:

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \ \ell \\ &= \arg\min_{\boldsymbol{\theta}} \ell_{\mathrm{MSE}}(F(\mathbf{x}^p; \boldsymbol{\theta}), \hat{\mathbf{y}})\end{aligned}$$

Appendix

Backpropagagtion

Minimize ℓ through tuning $\boldsymbol{\theta}_i$:

$$\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k + \eta \frac{\partial \ell}{\partial \boldsymbol{\theta}_i^k}$$

Closed-form function:

$$\mathbf{y}^p = f_n(f_{n-1}(\dots f_1(f_0(\mathbf{x}))) = F(\mathbf{x}^p; \boldsymbol{\theta})$$

Gradient w.r.t. $\boldsymbol{\theta}_i$ through chain rule:

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}_i} = \frac{\partial \ell}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}_i} \quad \leftarrow \text{chain rule}$$

Automatic Differentiation (AD)

$$= \frac{\partial \ell}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{h}_n} \frac{\partial \mathbf{h}_n}{\partial \mathbf{h}_{n-1}} \cdots \frac{\partial \mathbf{h}_i}{\partial \boldsymbol{\theta}_i} \quad \leftarrow \mathbf{y} = f_n(f_{n-1}(\dots f_1(f_0(\boldsymbol{x}))) = F(\mathbf{x}; \boldsymbol{\theta})$$

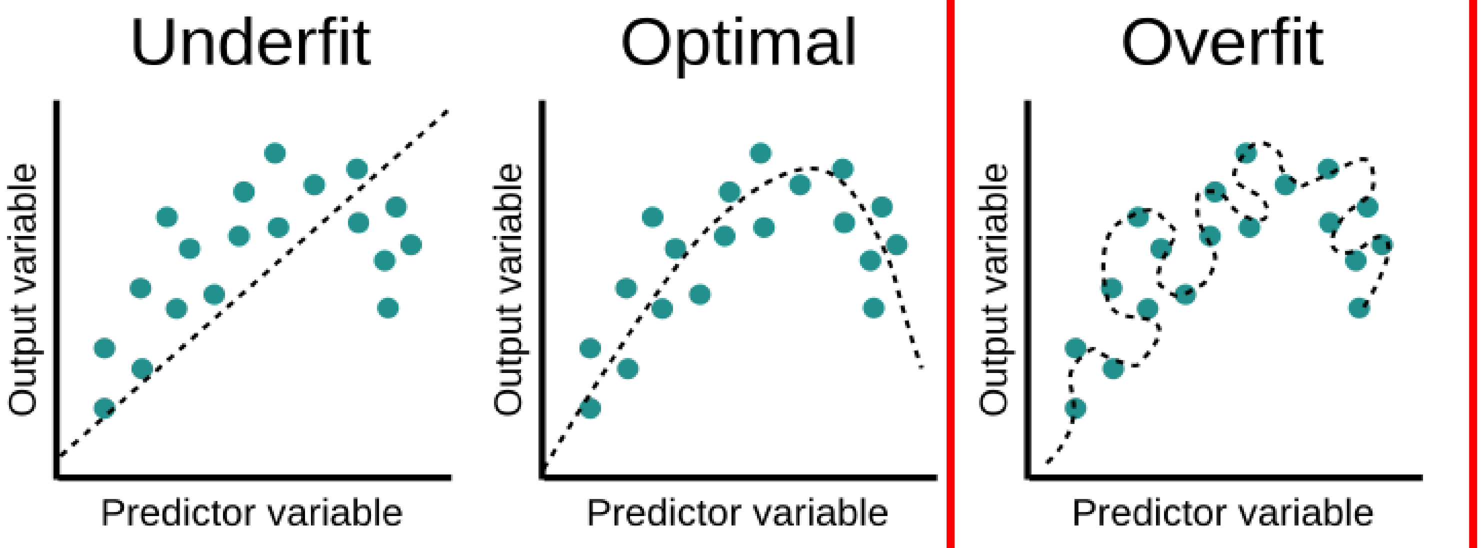
$$= g' f_n' f_{n-1}' \dots f_i'$$

$$= g' F_{\boldsymbol{\theta}_i}(\mathbf{x}; \boldsymbol{\theta}) \quad \leftarrow \text{abbreviation}$$

Appendix

Overfitting

$\dim(\boldsymbol{\theta}) \ll N_{data}$ | $\dim(\boldsymbol{\theta}) \approx N_{data}$ | $\dim(\boldsymbol{\theta}) \gg N_{data}$



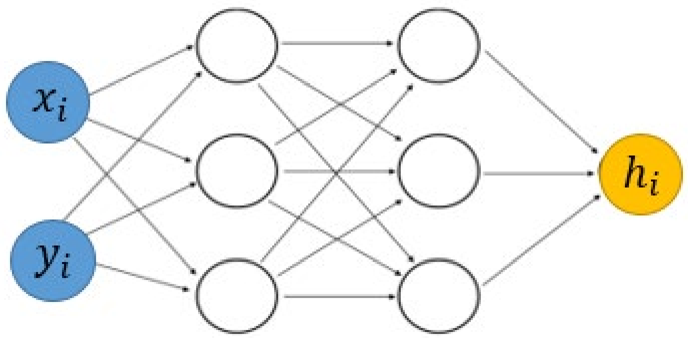
In groundwater inverse problem, data is limited!

Appendix

Steady-State Forward Network

Steady-state pumping test only monitors hydraulic heads at steady state (time invariant).

NN has no input time variables



Steady-state Forward Network Architecture

Input variables	Spatial (x, y)
Output variables	Steady-state hydraulic heads (h)
Number of hidden layers	7
Dimension of hidden variables	50
Hidden layer activation function	Hyperbolic (*tanh*)
Output layer type	Linear

Loss of Steady-State PINN

Steady-state HT-PINN is more efficient because of smaller training data and fewer and simpler constraints.

- Hydraulic heads are only monitored at steady state (no intermediate time step)
- No constraints of initial condition
- No temporal gradients in PDE constraints for pumping and non-pumping grids

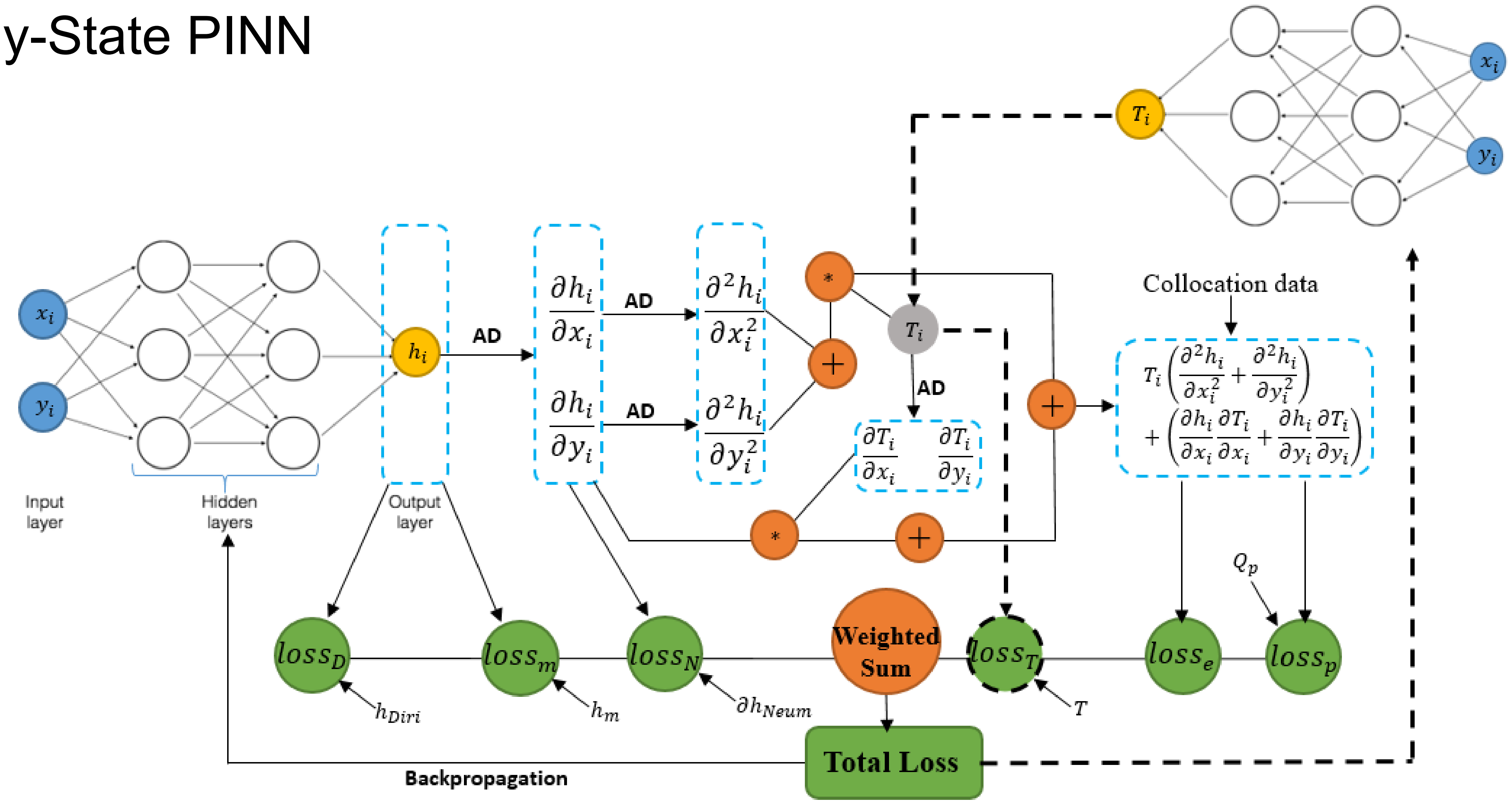
Monitored hydraulic heads: $NN(x_m, y_m) = h_m$

Measurements of transmissivity: $TNN(x_T, y_T) = T(x_T, y_T)$

PDE for non-pumping grid	$\nabla \cdot [T(x_e, y_e)\nabla h(x_e, y_e)] = 0,$	$\nabla \cdot [TNN(x_e, y_e)\nabla NN(x_e, y_e)] = 0,$	$(x_e, y_e) \in \Omega$
PDE for pumping grid	$\nabla \cdot \left[T\left(x_p, y_p\right)\nabla h\left(x_p, y_p\right)\right] = Q_p,$	$\nabla \cdot \left[TNN\left(x_p, y_p\right)\nabla NN\left(x_p, y_p\right)\right] = Q_p,$	$(x_p, y_p) \in \Omega$
Neumann Boundary Condition	$\mathbf{n} \cdot \nabla h(x_N, y_N) = q_N,$	$\mathbf{n} \cdot \nabla NN(x_N, y_N) = q_N,$	$(x_N, y_N) \in \Gamma_N$
Dirichlet Boundary Condition	$h(x_D, y_D) = h_D,$	$NN(x_D, y_D) = h_D,$	$(x_D, y_D) \in \Gamma_D$

Appendix

Steady-State PINN



Appendix

PDE Loss Terms

Physical Constraints:

$$S_s \frac{\partial h(x_e, y_e, t_e)}{\partial t} - \nabla \cdot [T(x_e, y_e) \nabla h(x_e, y_e, t_e)] = 0,$$

PDE residual:

$$f_{NN^i, TNN}(x, y, t) = S_s \frac{\partial NN^i(x, y, t)}{\partial t} - \nabla \cdot [TNN(x, y) \nabla NN^i(x, y, t)],$$

PDE for non-pumping grid:

$$Loss_e = \frac{1}{N_e} \sum_{j=1}^{N_e} \left| f_{NN,TNN}\left(x_e^j, y_e^j, t_e^j\right) \right|^2$$

PDE for pumping grid:

$$Loss_p = \frac{1}{N_p} \sum_{j=1}^{N_p} \left| f_{NN,TNN}\left(x_p^j, y_p^j, t_p^j\right) - Q_p \right|^2$$

Appendix

BC Loss Terms

Dirichlet B.C.:

$$Loss_D = \frac{1}{N_D}\sum_{j=1}^{N_D}\left|NN\left(x_D^j, y_D^j, t_D^j\right) - h\left(x_D^j, y_D^j, t_D^j\right)\right|^2$$

Neumann B.C.:

$$Loss_N = \frac{1}{N_N}\sum_{j=1}^{N_N}\left|\boldsymbol{n}\cdot\nabla NN\left(x_N, y_N, t_N\right) - q_N\right|^2$$

Initial Condition:

$$Loss_{init} = \frac{1}{N_{init}}\sum_{j=1}^{N_{init}}|NN(x_{init}, y_{init}, 0) - h(x_{init}, y_{init}, 0)|^2$$

Appendix

Data Match Loss Terms

Monitored Hydraulic Heads:

$$Loss_m = \frac{1}{N_m}\sum_{j=1}^{N_m}\left|NN\left(x_m^j, y_m^j, t_m^j\right) - h\left(x_m^j, y_m^j, t_m^j\right)\right|^2$$

Measured Transmissivity:

$$Loss_T = \frac{1}{N_T}\sum_{j=1}^{N_T}\left|TNN\left(x_T^j, y_T^j\right) - T\left(x_T^j, y_T^j\right)\right|^2$$

Appendix

Loss Function of HT-PINN

$$Loss_{NN} = \lambda_m Loss_m + \lambda_e Loss_e + \lambda_N Loss_N + \lambda_D Loss_D + \lambda_p Loss_p + \lambda_{init} Loss_{init}$$

$$Loss_{HT-PINN} = \sum_{i=1,2,\ldots n} Loss_{NN}^{i} + \lambda_T Loss_T$$

$$\lambda_m = 10^4, \lambda_f = 50, \lambda_p = 1, \lambda_N = 10^4, \lambda_D = 2 \times 10^4, \lambda_T = 10^3, \lambda_{init} = 10^4$$

Appendix

Loss Function of steady state HT-PINN

Physical Constraints:

$$\nabla \cdot [T(x_e, y_e)\nabla h(x_e, y_e)] = 0,$$

PDE residual:

$$f_{NN,TNN}(x, y) = \nabla \cdot [TNN(x, y)\nabla NN(x, y)]$$

PDE for non-pumping grid:

$$Loss_e = \frac{1}{N_e}\sum_{j=1}^{N_e} \left| f_{NN,TNN}\left(x_e^j, y_e^j, t_e^j\right) \right|^2$$

PDE for pumping grid:

$$Loss_p = \frac{1}{N_p}\sum_{j=1}^{N_p} \left| f_{NN,TNN}\left(x_p^j, y_p^j, t_p^j\right) - Q_p \right|^2$$

$$Loss_{NN} = \lambda_m Loss_m + \lambda_e Loss_e + \lambda_N Loss_N + \lambda_D Loss_D + \lambda_p Loss_p$$

$$Loss_{HT-PINN} = \sum_{i=1,2,\dots n} Loss_{NN}^i + \lambda_T Loss_T$$

$$\lambda_m = 10^4, \lambda_f = 50, \lambda_p = 1, \lambda_N = 10^4, \lambda_D = 2 \times 10^4, \lambda_T = 10^3$$

Appendix

HT-PINN Training

- 5 forward networks + 1 inverse network are trained together.
- Reference data are corrupted with 5% white noises.
- Input and output variables are normalized.
- Different loss terms are weighted to similar magnitude.
- Each training iteration takes a batch of data to feed HT-PINN.
- Each epoch has 50 iterations for steady-state and 500 iterations for transient HT.
- HT-PINN is trained for 3000 epochs with Adam optimizer.
- Learning rate = 10^{-3} for 1-1000, 10^{-4} for 1000-2000, 10^{-5} for 2000-3000.
- Training hardwares are Google Colab GPU