GRAPH NEURAL NETWORKS FOR PERMEABILITY ESTIMATION IN POROUS MEDIA

Author: Oleg Afanasiev, V. N. Karazin Kharkiv National University
Contact: oleg.afanasiev@karazin.ua

AGENDA

1. Motivation
2. Physical Setup
3. Data Generation
4. Ground-Truth Pipeline
5. Graph Features
6. Model
7. Training Protocol
8. Results — In-Range (mixed p)
9. Results — Extrapolation
10. Ablations & Insights (brief)
11. Runtime & Practical Notes
12. Related Work
13. Conclusions
14. Future Work

1.MOTIVATION

- Estimating effective permeability k_{eff} is central to filtration, batteries, and composites.
- Direct Darcy/Laplace solves are accurate but expensive at scale.
- Goal: a **fast, physics-consistent GNN surrogate** for k_{eff} .

2. PHYSICAL SETUP

- Domain: binary porous layouts 24×24.
- Dirichlet BCs: $P = 1$(left), $P = 0$(right).
- If no left–right connectivity $\Rightarrow k_{\text{eff}} = 0$.
- Ground truth k_{eff} :discrete Darcy/Laplace + flux.

3. DATA GENERATION

- Bernoulli percolation (open cell with prob. p).
- Training base: $p \in [0.40,\ 0.80]$.
- Tails for robustness:$[0.40,\ 0.40]$, $[0.80,\ 0.90]$, [0.90, 0.97].
- Graph: nodes = open cells; edges = 4-neighborhood.

4. GROUND-TRUTH PIPELINE

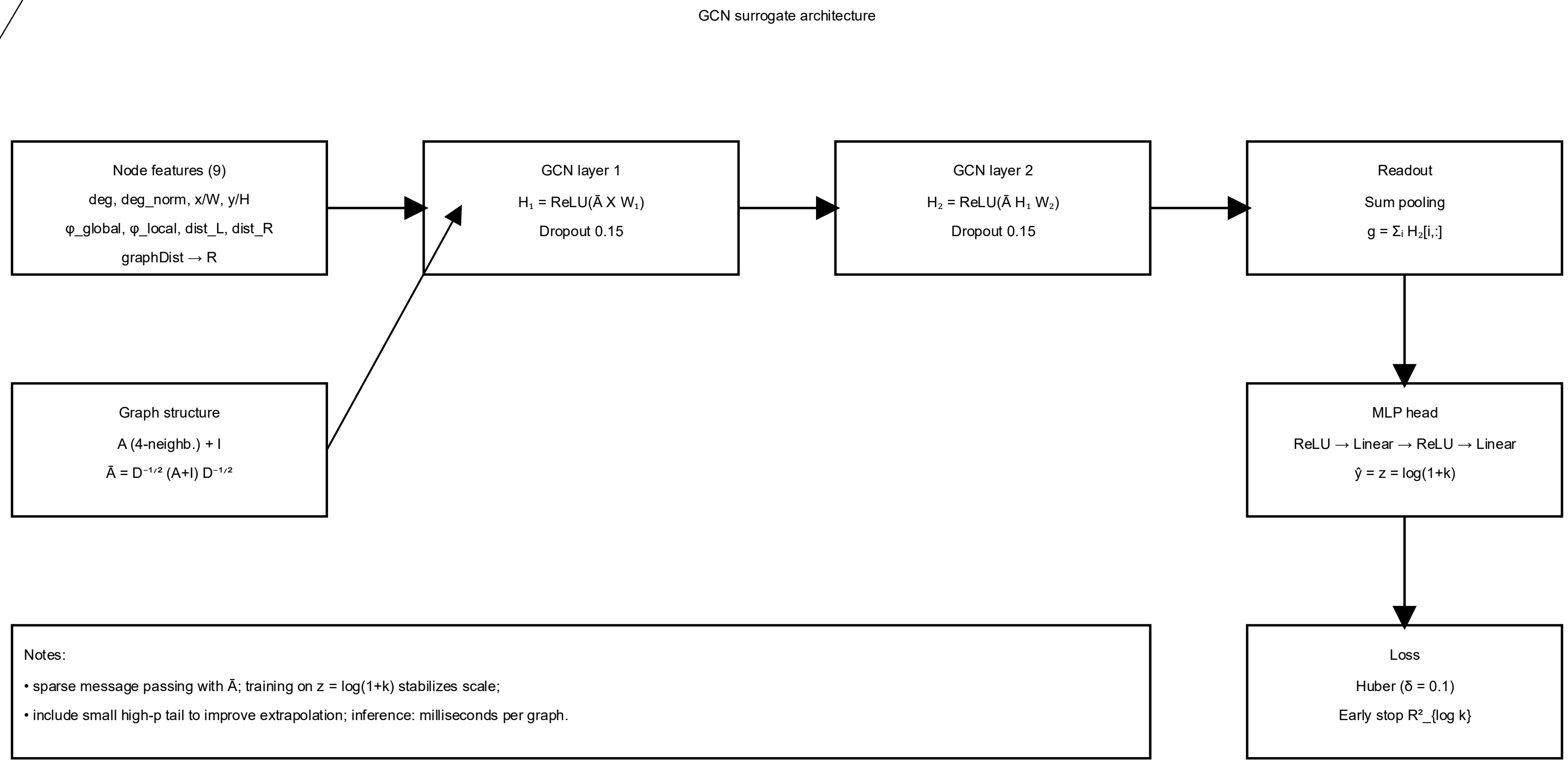
- Remove components not touching boundaries.
- If L↔R disconnected ⇒ $k_{\text{eff}} = 0$.
- Solve discrete Laplacian for P.
- Flux through right boundary ⇒ k_{eff} (proper normalization).

5. GRAPH FEATURES

- Degree; normalized degree.
- Coordinates (x/W y/H).
- Global porosity; local porosity (5×5 average).
- Distances to left/right boundaries (normalized).
- Normalized graph distance to right boundary.
- Readout: **sum pooling**.

6. MODEL

- **2-layer GCN** (sparse message passing) + MLP head.
- Dropout: 0.15.
- Target: for scale stability.
- Loss: **Huber** $(\delta = 0.1)$
- Early stopping by $R^2_{\log k}$on validation.



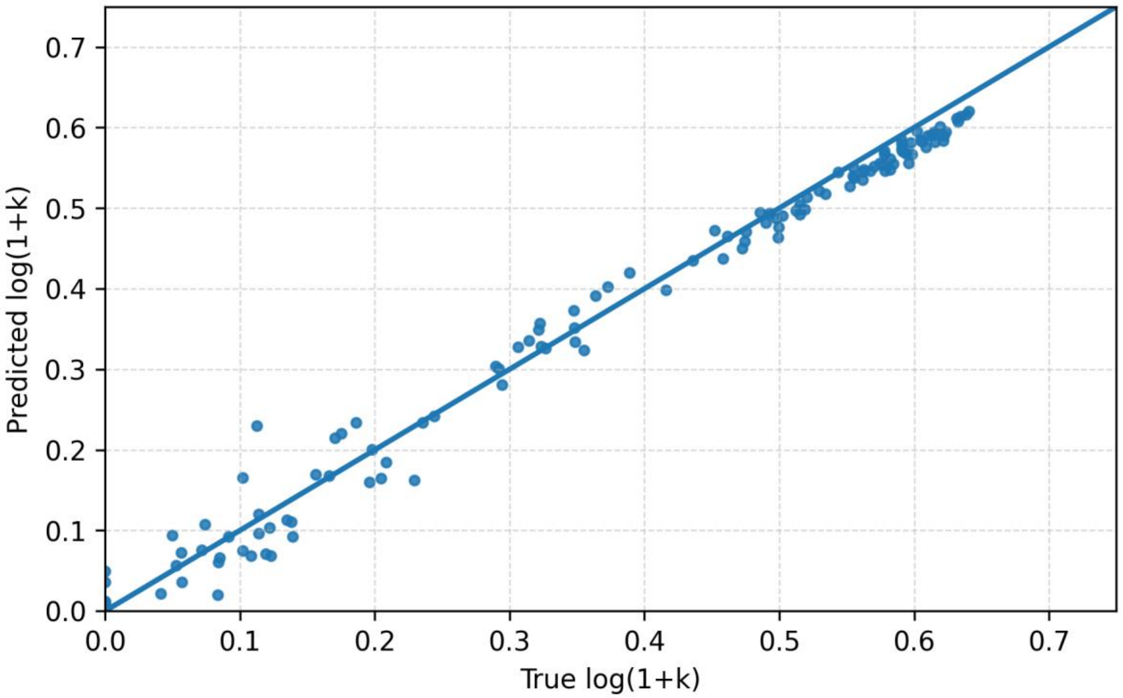
GCN surrogate architecture
Node features (9)
deg, deg_norm, x/W, y/H
φ_global, φ_local, dist_L, dist_R
graphDist → R
GCN layer 1
$H_1 = \mathrm{ReLU}(\bar{A} X W_1)$
Dropout 0.15
GCN layer 2
$H_2 = \mathrm{ReLU}(\bar{A} H_1 W_2)$
Dropout 0.15
Readout
Sum pooling
$g = \Sigma_i H_2[i,:]$
Graph structure
A (4-neighb.) + I
$\bar{A} = D^{-1/2} (A+I) D^{-1/2}$
MLP head
ReLU → Linear → ReLU → Linear
$\hat{y} = z = \log(1+k)$
Notes:
• sparse message passing with Ā; training on z = log(1+k) stabilizes scale;
• include small high-p tail to improve extrapolation; inference: milliseconds per graph.
Loss
Huber (δ = 0.1)
Early stop R²_{log k}

7. TRAINING PROTOCOL

- Stratified sampling over p(base + tails).
- Optimizer: Adam 1×10^{-3} ,weight decay 1×10^{-5} .
- Best epoch selected by max $R^2_{\log k}$.

8. RESULTS — IN-RANGE (MIXED p)

- $R^2_{\log k} \approx$ **0.993**
- $\mathrm{RMSE}_k \approx$ **0.028**
- $\mathrm{MAE}_k \approx$ **0.019**



9. RESULTS — EXTRAPOLATION

- **Low-p (0.20–0.35):** $\mathrm{MAE}_k \approx 0.0076$,$\mathrm{RMSE}_k \approx 0.0137$) $R^2_{\log k}$n/a due to near-zero variance).
- **High-p (0.93–0.97):** $R^2_{\log k} \approx \mathbf{0.727}$,$\mathrm{RMSE}_k \approx 0.0240$,$\mathrm{MAE}_k \approx 0.0211$.

10. ABLATIONS & INSIGHTS (BRIEF)

- log(1+k) target $>$linear k for stability.
- Graph-distance feature to the right boundary is important near percolation.
- Small high-p tail in training improves extrapolation.

11. RUNTIME & PRACTICAL NOTES

- Discrete Laplace solve: ms–s per graph (CPU), depends on component size.
- GCN inference: ms, easily batched.
- Implementation: PyTorch + SciPy (sparse).

12. RELATED WORK

- Yu, W. & Lyu, P. (2020). Unsupervised machine learning of phase transition in percolation, Physica A: Statistical Mechanics and its Applications, 559, 125065. https://doi.org/10.1016/j.physa.2020.125065

13. CONCLUSIONS

- A **physics-consistent GNN surrogate** for k_{eff} on random porous layouts.
- High in-range accuracy; reasonable extrapolation to extremes of p.
- Simple architecture, fast inference, interpretable features.

14. FUTURE WORK

- Weighted/anisotropic edges; richer geometry.
- 3D domains and irregular grids.
- Joint k_{eff}regression + percolation classification.
- Active learning for tail regimes of p.

THANK YOU

Oleg Afanasiev

oleg.afanasiev@karazin.ua