Phase III: Application/Test in Real-World Grids

Multi-Measure Benefits
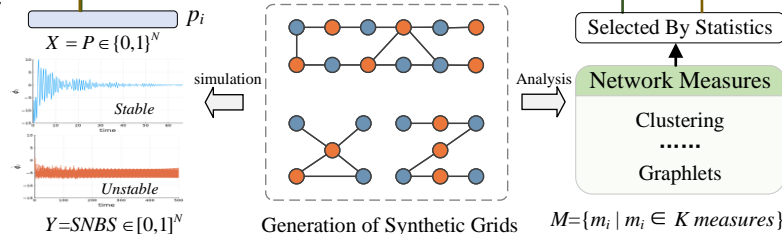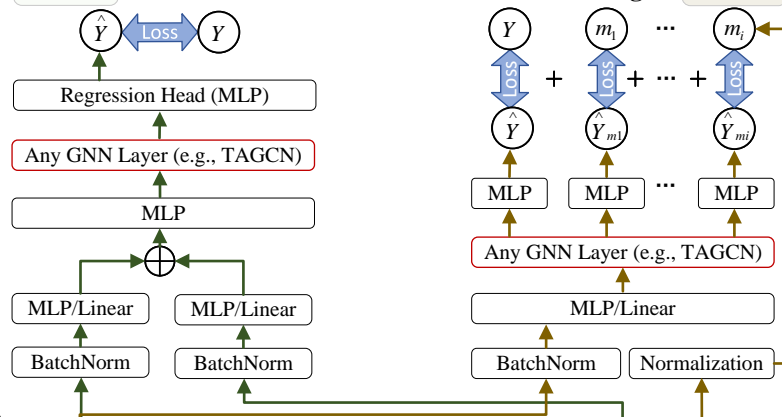
Optimized with Fewer Measures

Prediction: R^2 66%→**83%** (*tr20evTexas*); Time Reduction: 28,950 hours→**0.06** seconds

NSGNN-F **Phase II: Two GNN Network Science Strategies** NSGNN-S

$\hat{Y}$ ⟷Loss⟷ Y

Regression Head (MLP)

Any GNN Layer (e.g., TAGCN)

MLP

MLP/Linear   MLP/Linear

BatchNorm   BatchNorm

Y m_1 ⋯ m_i

Loss + ⋯ + Loss

$\hat{Y}$ $\hat{Y}_{m1}$ ⋯ $\hat{Y}_{mi}$

MLP   MLP ⋯ MLP

Any GNN Layer (e.g., TAGCN)

MLP/Linear

BatchNorm   Normalization

Selected By Statistics

Phase I: Datasets Preparation and Analysis

p_i

$X = P \in \{0,1\}^N$

Stable

Unstable

simulation

Generation of Synthetic Grids

Analysis

Network Measures

Clustering

......

Graphlets

$M = \{m_i \mid m_i \in K \text{ measures}\}$

$Y = SNBS \in [0,1]^N$

Legend:

X: Node initial features (P)

→ Data flow of NSGNN-F

→ Data flow of NSGNN-S

Loss — Training loss

Y — True Label (*SNBS*)

$\hat{Y}$ — Predicted Label (*SNBS*)

m_i — i^{th} true network measure

$\hat{Y}_{mi}$ — i^{th} predicted network measure

⊕ — Concatenation

BatchNorm — Ensures that network activations across mini-batches are within a similar range

Normalization — Scales data to a uniform range (between 0 and 1), ensuring consistent input distribution

MLP — Multi-layer perceptron which processes the encodings to learn a meaningful structure

NSGNN-F — Represents a strategy by integrating network measures as learnable input features

NSGNN-S — Represents a strategy by integrating network measures as self-supervised information

GNN Layer — Any state-of-the -art GNN can be replaced here