Implementation Plan: Physics-Informed MeshGraphNets (PI-MGN) for Structural Mechanics

# PHASE 1: Mesh + Geometry Generation

Goal: Create 2D composite RVEs with controllable fiber layout

Tasks:

* Install Gmsh & pygmsh
* Script: mesh\_gen/generate\_rve.py
* Define matrix domain (e.g., 1×1 box)
* Insert N circular fibers at random or controlled positions
* Assign fiber and matrix regions as physical groups
* Mesh the domain with triangular elements

Test:

* Export .msh file and visualize in Gmsh GUI
* Check physical groups (fiber = 2, matrix = 1)

# PHASE 2: Convert Mesh to Graph

Goal: Build PyTorch Geometric graphs from Gmsh meshes

Tasks:

* Script: graph\_builder/mesh\_to\_graph.py
* Load mesh using meshio
* Extract node coordinates, triangle connectivity, region labels
* Assign node features: position, region ID
* Edge features: relative position vectors
* Encode boundary condition flags on nodes

Test:

* Visualize graph using matplotlib or networkx
* Verify correct labeling of fiber vs matrix and BCs

# PHASE 3: Build PI-MGN Model

Goal: Define MeshGraphNet for structural mechanics

Tasks:

* Script: models/pi\_mgn.py
* Define encoder MLPs for node and edge features
* Implement message passing (e.g., GINEConv)
* Decode to predict 2D displacement field u = [ux, uy]

Test:

* Pass dummy graph through model
* Verify output shape = [num\_nodes, 2]

# PHASE 4: FEM Residual Loss Function

Goal: Replace ground-truth supervision with physics residual

Tasks:

* Script: train/loss\_fem\_residual.py
* Compute symmetric strain ε from displacements
* Apply constitutive law σ = C : ε
* Evaluate residual ∇·σ + f
* Return MSE loss over all elements/nodes

Test:

* Use small square domain with known uniform loading
* Visualize residual field

# PHASE 5: Training Script

Goal: Train on one static mechanical RVE

Tasks:

* Script: train/train\_static.py
* Load graph from .pt
* Forward pass through PI-MGN
* Compute residual loss
* Backprop + optimize (Adam, 1e-3)
* Save checkpoints and logs

Test:

* Verify training loss decreases
* Visualize displacement field

# PHASE 6: Validation + Generalization

Goal: Test on unseen RVEs and mesh resolutions

Tasks:

* Generate test RVEs with different fiber counts
* Predict with trained model
* Evaluate residual loss and displacement fields

Test:

* Compare with Abaqus results (optional)
* Check if effective E11 and symmetry are preserved

# PHASE 7: Inverse Design

Goal: Given target E11, find fiber layout

Tasks:

* Script: inverse\_design/optimize\_layout.py
* Define design parameters (fiber positions, angles)
* Build graph → predict E11 with PI-MGN
* Optimize layout to match E11target
* Use gradient-based or evolutionary optimization

Test:

* Target different stiffness values
* Visualize resulting layout