Algorithm 1: Framework of Meta-GNN.

Input: Distribution over mete-training tasks: $p(\mathcal{T})$; Meta-testing tasks: $\mathcal{T}_{mt}$; Task-learning rate: α_1; Meta-learning rate: α_2.
Output: Labels of nodes in query set of $\mathcal{T}_{mt}$.

Initialize θ randomly;
while not converged **do**
  Sample batch of meta-training tasks $\mathcal{T}_i \sim p(\mathcal{T})$;
  foreach *task in* $\mathcal{T}_i$ **do**
    Evaluate $\mathcal{L}_{\mathcal{T}_i}\left(f_\theta\right)$ using $\mathcal{S}_i$;
    Compute adapted parameters θ_i';
    Evaluate $\mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right)$ using $\mathcal{Q}_i$;
  Update θ by;
Compute adapted parameters θ_{mt}' using support set of $\mathcal{T}_{mt}$;
Predict labels of nodes in query set of $\mathcal{T}_{mt}$ using model $f_{\theta_{mt}'}$.