Scalable and Consistent Graph Neural Networks for Distributed Mesh-based Data-driven Modeling

https://doi.org/10.48550/arXiv.2103.07719

Distribute GNN

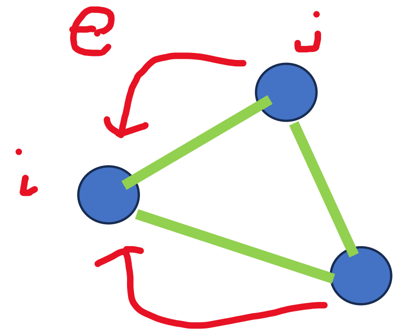
2024-11-01

Distribute GNN

- Scale No of nodes form 1e5 to 1e9
  - Finer grained mesh provide more insight
- Maintain consistence.
  - Result from 1 whole graph is the same as from distributed graph.

Graph NN notion

- Node feature, edge feature, adjacent matrix, parameter
- Message Passing
  - (1) an edge feature update conditioned on its corresponding sender and receiver nodes => msg formation
  - (2) a summation-aggregation of the updated edge features
  - (3) a node feature update conditioned on the aggregated edge features.
- Readout
  - Postprocessing layer
  - Node level -> one output each node
  - Graph level -> one output each graph



$$\underline{\text{Edge update}} : \mathbf{e}_r^{ij} = \mathrm{MLP}(\mathbf{x}_r^i, \mathbf{x}_r^j, \mathbf{e}_r^{ij}),$$

$$\underline{\text{Local edge aggr.}} : \mathbf{a}_r^i = \sum_{j \in N(i)} \frac{1}{d_r^{ij}} \mathbf{e}_r^{ij},$$

$$\underline{\text{Node update}} : \mathbf{x}_r^i = \mathrm{MLP}(\mathbf{a}_r^{i,*}, \mathbf{x}_r^i).$$

Consistent Neural Message Passing Layer

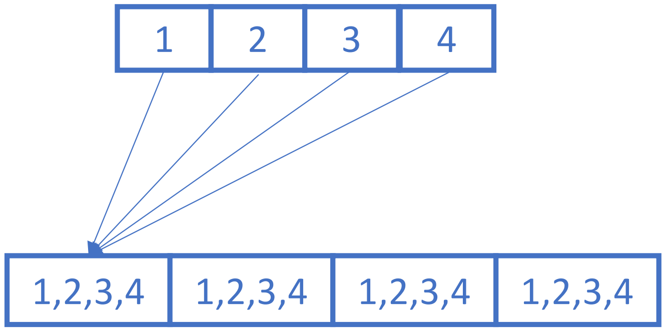
- Consist message passing operator

$$\mathcal{S}\left(\{\mathbf{Y}_1^{\text{local}}\}_{R=1}\right) = \mathcal{S}\left(\text{cat}\{\mathbf{Y}_1^{\text{local}}, \ldots, \mathbf{Y}_R^{\text{local}}\}_{R>1}\right)$$

- Consist Loss function
  - AllReduce to get the aggregation of partitioned Loss on each nodes

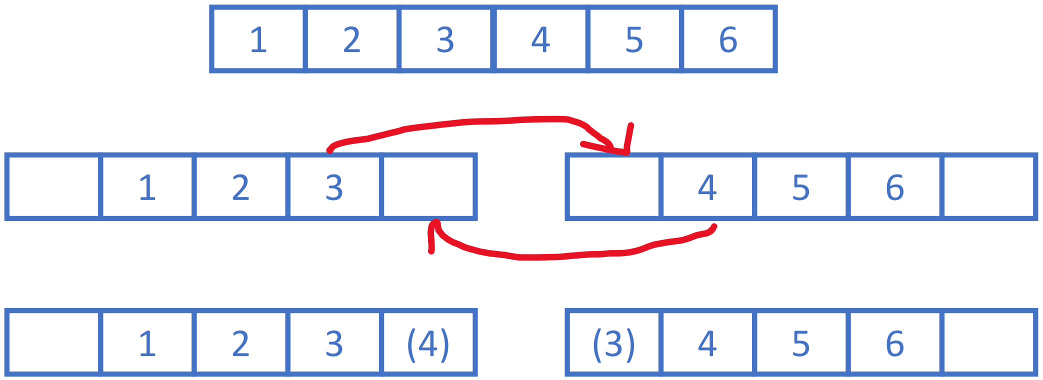
$$\left.\frac{\partial \mathcal{S}}{\partial \theta}\right|_{R=1} = \left.\frac{\partial \mathcal{S}}{\partial \theta}\right|_{R>1}.$$

$$\mathcal{L} = \frac{1}{N_{\text{eff}} F_y} \text{AllReduce}(\mathcal{S}_r),$$

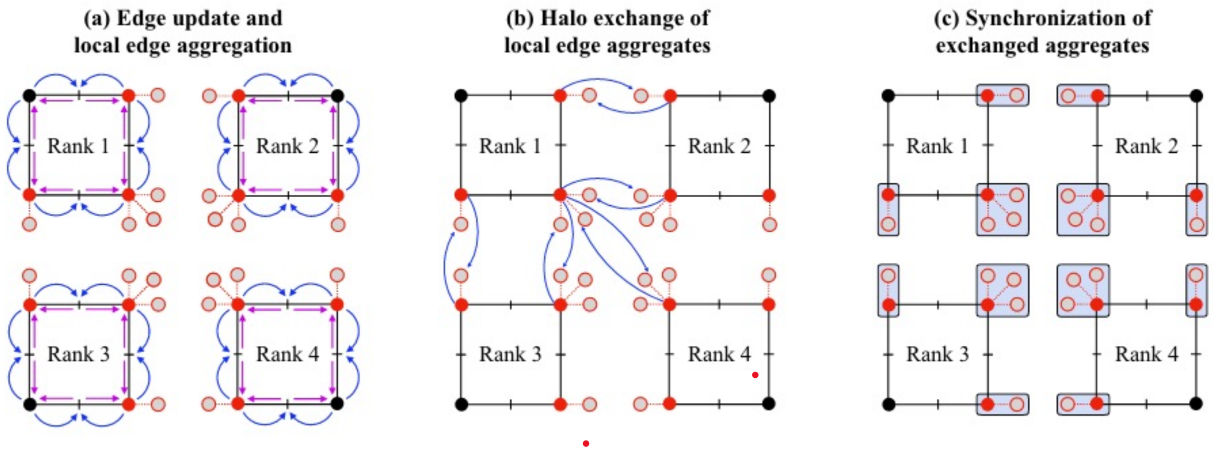


Halo swap

- An operation that need to know the value of its neighbors
  - such as x[i] = mean(x[i-1], x[i+1])



Neural halo swap



$$\underline{\text{Edge update}} : \mathbf{e}_r^{ij} = \mathrm{MLP}(\mathbf{x}_r^i, \mathbf{x}_r^j, \mathbf{e}_r^{ij}),$$

$$\underline{\text{Local edge aggr.}} : \mathbf{a}_r^i = \sum_{j \in N(i)} \frac{1}{d_r^{ij}} \mathbf{e}_r^{ij},$$

$$\underline{\text{Halo swap}} : \mathbf{a}_r^{i,\text{halo}} = \mathbf{a}_s^{k,\text{local}} \text{ if } G_r(i) = G_s(k),$$

$$\underline{\text{Synchronization}} : \mathbf{a}_r^{i,*} = \sum_{\substack{j \\ G_r(j)=G_r(i)}} \mathbf{a}_r^j,$$

$$\underline{\text{Node update}} : \mathbf{x}_r^i = \mathrm{MLP}(\mathbf{a}_r^{i,*}, \mathbf{x}_r^i).$$

Blue array: Edge update => msg
Red array: sum of local msg

Halo swap: get msg adjacent node which is on another rank

After sync:
summation of all aggregates on nodes (including halo nodes) sharing the same global index