

# # Weekly Report 2019.07.22-2019.07.28

Jingtun ZHANG

WHERE WE ARE:

JINGTUN ZHANG SUMMER INTERNSHIP AT UCSB					
TIME	DFF MODEL	HAG	GNN	GDyNet	void
2019.07.08 ----- 2019.07.14	<div>Single Conv Layer Linear Transform</div> <div>↓</div> <div>Multi Conv Layer: Magnitude of residual will not be magnified Exponentially</div>	<div>Implementation Question In Redundancy Computing</div> <div>↓</div> <div>Not Optimal</div>	<div>PyG tutorial</div> <div>↓</div> <div>GNN review and MPNN Model paper reading</div> <div>↓</div> <div>unreasonable profiling of MPNN</div>		
2019.07.15 ----- 2019.07.21		<div>O(V3) complexity is not reasonable</div> <div>↓</div> <div>More Understanding of HAG and O(VlogV) redundancy computation</div>			
2019.07.12 ----- 2019.07.28	<div>Output Flaw and detail problem</div>			<div>CGCNN and GDyNet Paper reading and Code review</div> <div>↓</div> <div>Code reading question Not test new pooling algorithm</div>	

## ## Work and Progress

### 1. Reimplementation of [HAG](#):

#### 1. Understanding of the model:

$$H^{(k+1)} = A_u A_{aggr} H^{(k)}$$

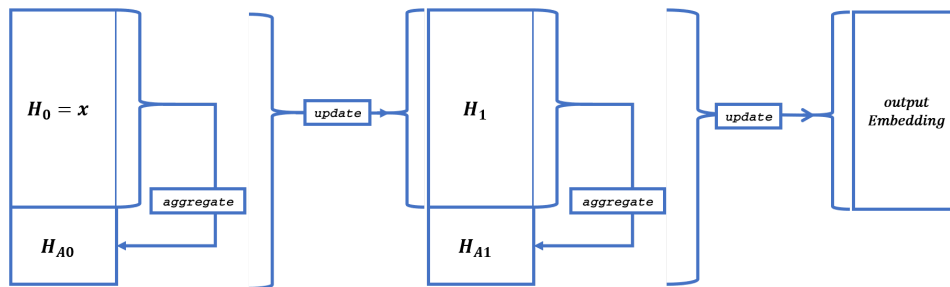
$A_u$  ,  $A_{aggr}$ : Get New Topology at Preprocessing stage and use at all layers

$A_{aggr}$ : First Phase:

compute embedding of aggregate node by Algorithm 2

$A_u$ : Second Phase:

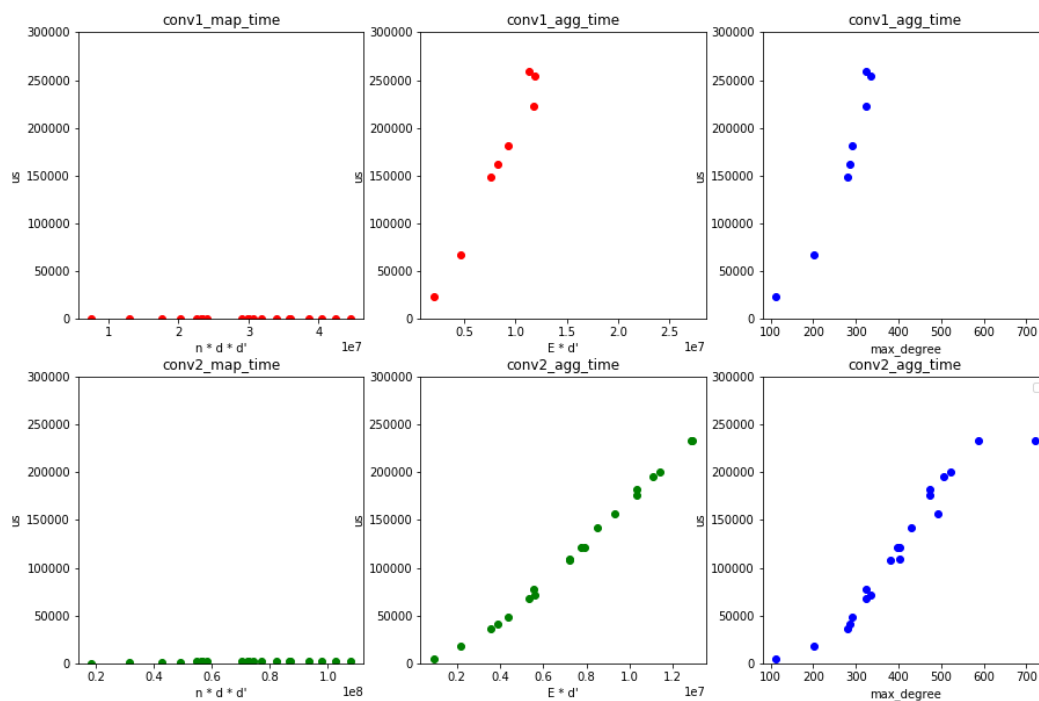
compute embedding of original node in GNN model method



2. Idea of redundancy computation: build node-pair heap every iteration in HAG building process -->  $O(V \log V)$
3. Coding: in coding now...

### 2. Reasonable complexity analysis of MPNN network:

1. aggregate time is proportional to Edge in the Graph:  $\text{agg\_time} \sim O(E)$
2. aggregate time can be optimized by CUDA to graph invariant (graph with thousands of nodes)
3. mapping time complexity --> still unreasonable --> need profiling of `tf.matmul` procedure



3. [GDyNet](#) and [CGCNN](#) model: Test different pooling method in GDyNet

1. Paper Reading:

1. CGCNN: node structure --> property --> Graph classification
2. GDyNet: do not understand the task in detail, from model point:  
Markovain Process & node embedding (for trarget node) --> dynamic  
property of atoms

2. Code reading:

1. CGCNN: good pytorch code and understandable
2. GDyNet: tensorflow.keras is not a good style --> not figure out the  
content of data flow and the pooling function in release code seems  
no pooling function

## ## This week plan

1. give out a Good HAG Code

-----