Checkpoint 4

Research Questions

1. Utilizing graph neural networks, and the attention mechanism from NLP literature, make a general predictor which can understand network structures of the web (made up of the three dimensions discussed above), by extracting the attention layer of the GNN we can acquire an importance weight vector of different dimensions of data. The GNN predictor will output the predicted number of officer complaints for the next year when given a complete history trajectory of that officer. The history trajectory includes officer/civilian allegation events, award events, etc.

1.1 Preface:

Graph neural networks (GNN) are pretty similar to traditional convolution networks, however, since the graph topology is not unified at every vertex (while CNN deals with fixed grid topology like pixels in images), There are mainly two categories of GNN methods: spectral based and non-spectral methods. We are using a non-spectral method here in our research since it is more flexible and allows us to integrate the well-known attention mechanism in ML to our model.

Attention mechanism was originally introduced in the transformer model in NLP literature, it was introduced to separate a special kind of parameter in the neural network model which computes the relation between two words in a sentence vector. Later on it was adapted to machine vision research and combined with CNN by Yin et al. It was also adapted to GNN research afterwards, and a famous example is the GAT by Veličković et al. We will use GAT and further adapt it to our CPDB data in this research question.

1.2 The original GAT model:

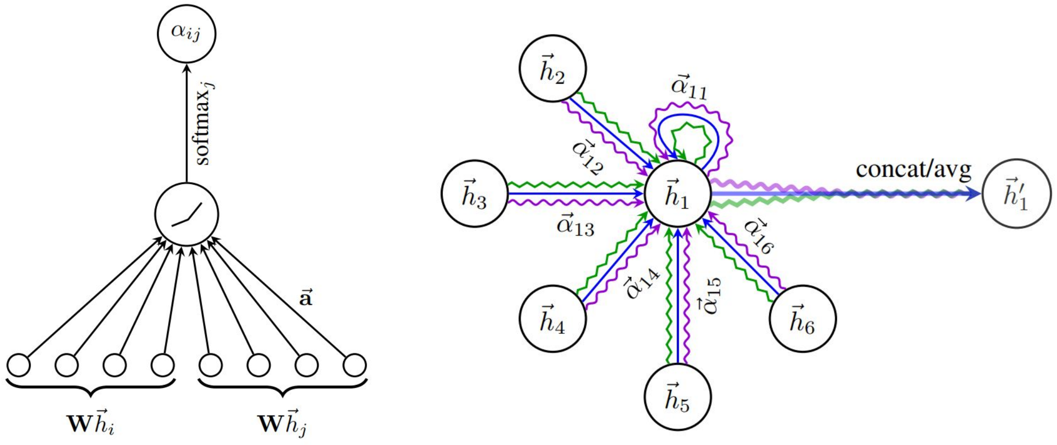


Figure 1.3.1 GAT model (*left is attention mechanism, right is node info aggregation*).

In each GAT component, three things are done in order:

1. Perform a linear transformation on input features $h' = Wh$
2. Compute attention weight for each pair of adjacent nodes:
    a. $a = softmax(\ leakyrelu(\ a_{param}\ [h_{i,}h_j]))$
    b. a_{param} is a vector of shape [2 * h’ row width, 1] (Figure 1.3.1 left)
    c. $[h_{i,}h_j]$ means concatenating features of connecting nodes
3. Multiply attention weight with input, to get new features of all output nodes. Concatenate them (Figure 1.3.1 right)

There are two things worth checking in this process, the first thing is a_{param}, because it reflects how our neural network weights each input feature, and computed a, because a is essentially how important a node is, given feature importance vector a_{param}.

A side note about perception range: because in each GAT layer, a node can perceive information from its direct adjacent neighbors, two layers will enable a perception distance of 2, and 3 will enable 3, and so on.

1.3 Our model:

Our model is adapted on the basis of GAT, the original GAT model is composed of two layers of GAT layers (we will use GAT to reference both the model and layer components). The input layer is called “heads”, which is also borrowed from NLP literature, each “head” is essentially equivalent to a channel of CNN mask:

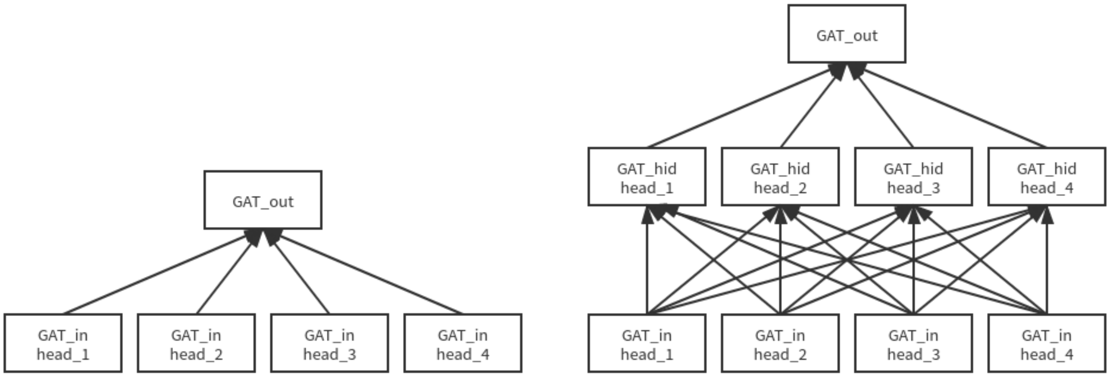


Figure 1.2.1 Models (*left is the original GAT model, right is ours*).

We have appended an additional layer to GAT, because the additional input layer does a identity transform to input features (while other layers will do a linear transformation), and we wish train our model to predict the number of civilian allegations some officer might receive in the next year, and inspect the attention weight of input layer after training to analyze how our model percepts the input features.

Another modification we have tried (but not used in the final version) is multiplying the attention weight with an additional adjacency weight, this weight is computed from civilian allegation count (cumulative or limited to one year) and normalized to the range of [0, 1], however there is only a very slight significant performance increase (with weight: test set MSE loss=0.7772, no weight: test set MSE loss = 0.7815, averaged on three training each). So we have removed this modification to prevent unwanted effects on attention weight.

1.3 Our data and processing method:

We have tried two types of features, the first type will aggregate data from the start year to current year (since we iterate from start year like 2006 to end year like 2010 and treat officers in each year as a independent individual to enrich available data), the second type will just look at the current year and predict. We find that no-aggregation greatly improves accuracy (with aggregation: test set MSE loss=0.7972, no aggregation: test set MSE loss = 0.7815, averaged on three training each).

We have removed some of the features we deemed inefficient in Checkpoint 2, such as officer salary and victim gender, since they do not exhibit any significant difference between different allegation percentile groups.

We have done a manual feature clustering process on some categorical features, like "**final outcome category**", "**Allegation type**", etc. Because some categories in this features are extremely rare (like only 1 officer has it), or that there are too many similar categories, causing feature dimension to explode, we choose to manually merge some of these categories, for example:

There are 63 categories of final allegation outcomes, you can get their count by doing the following SQL query:

```
select final_outcome, count(*)

from data_officerallegation

group by final_outcome

order by final_outcome;
```

We reduce the category number to 7 by doing the following seriousness clustering:

Warning	No result	Miscellaneous	Unknown	Short suspension	Medium suspension	Long suspension or permanently fired
Reprimand	No action taken	Reinstated by court	Unknown	1 day suspension	10 day suspension	31 day suspension

Violation noted	Penalty not served	Reinstated by police board		2 day suspension	11 day suspension	32 day suspension
	...	...		...	...	...

Table 1.3.1 new categories of final allegation outcomes.

For more information on how we cluster things, please reference “src/GNN/table.py”.

The last thing worth noting is encoding. We use one-hot encoding for categorical data, and perform normalization on contiguous data like age, disciplined rate, etc to ensure that their value ranges are [0, 1]. Normalization greatly stabilizes the training process, because neural networks are prone to “NaN” and “Inf” values.

In total we are using 12 types of features, since some of the features are made up of one-hot encodings, our feature vector is 42 in length, we present the meaning of each dimension in the table below:

Meaning	Officer gender	Officer race	Officer career time	Officer age	Officer allegation num	Officer allegation sustained rate
Index	[0 ... 1]	[2 … 7]	[8]	[9]	[10]	[11]

Meaning	Allegation final outcome	Allegation type	Disciplined rate	Is officer complaint rate	Victim race	Victim age
Index	[12 … 18]	[19 … 32]	[33]	[34]	[35 … 40]	[41]

Table 1.3.2 Meaning of each feature dimension.

1.3 Our findings:

1.3.1 Attention param

Remember that we have mentioned a_{param} used in attention calculation in section 1.2, it takes two features into consideration for each feature row, the first feature is the current node i, the second node is a node j selected from the list of nodes adjacent to node i. Therefore, attention is of dimension 84 in this case, the pink area in the figure below shows attention weight on node

i, and the right part show attention weight on node j s. We use absolute attention weight to analyze because it is safer to say that attention weight close to 0 has less effect than weights with bigger absolute values. (Negative values can have positive correlation with output because there is an additional hidden layer in our model).

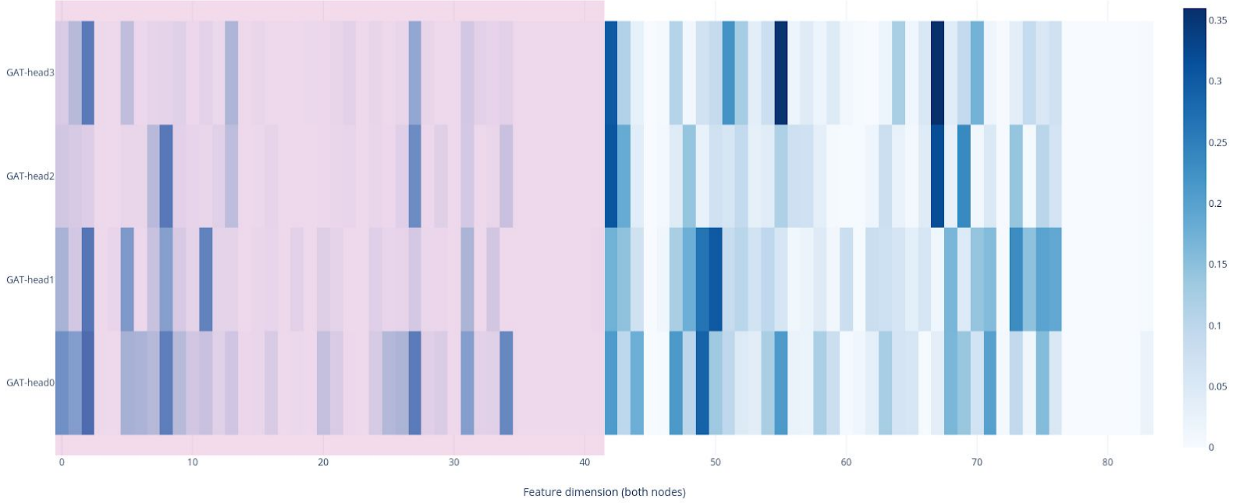


Figure 1.3.1.1 a_{param} heatmap visualization of all heads (absolute value).

We discover that our the input layer of our GAT model, which directly deals with raw features since it performs nothing but an identity transform, lays different weight on the current node and its adjacent nodes:

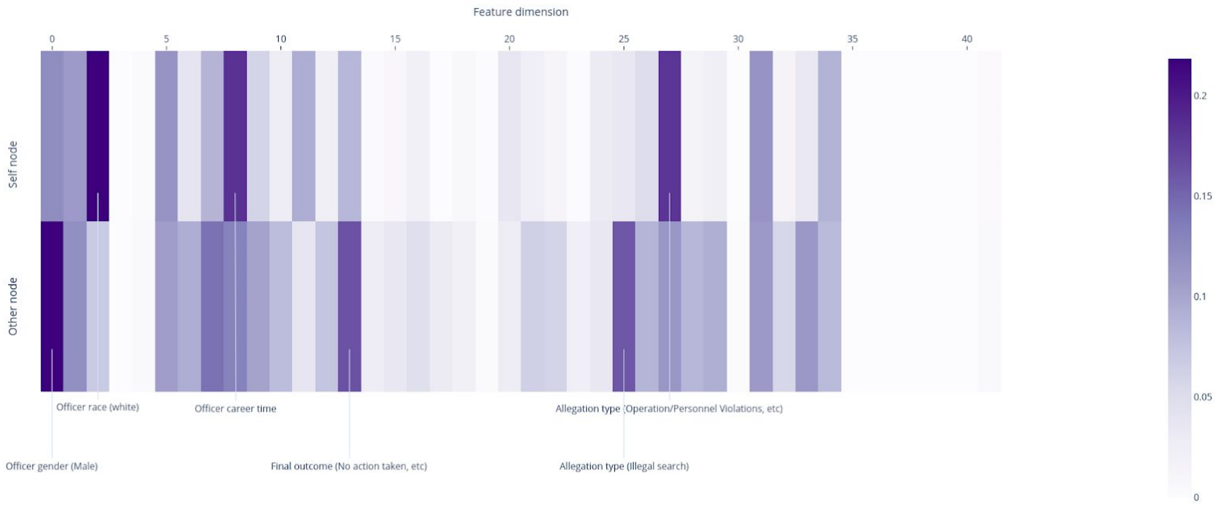


Figure 1.3.1.2 averaged a_{param} heatmap visualization (absolute value).

The first row represents weights on the current node i, we find that GAT will focus on:

1. dimension 3 (Officer race, white)
2. dimension 8 (Officer career time)
3. dimension 27 (Allegation type, "Operation/Personnel Violations", "First Amendment", "Racial Profiling","Medical")

The second row represents weights on some adjacent node j, GAT will focus on:

1. dimension 0 (Officer gender, male)
2. dimension 13 (Allegation final come, no action taken)
3. dimension 25 (Allegation type, “illegal search”)

Our conclusion is simple: GAT will focus on different features for the current officer, and neighbor officers which will probably affect the current officer (our data can only represent possible relations, because adjacency is reflected by common allegations between two officers in some year X.)

Due to the limitation of our context knowledge of police behavior, we cannot perform further interpretations on why GAT will select these features.

1.3.2 Attention on neighbor nodes

Since our train year range is 2006-2009 (larger range will introduce more variance and data unreliability, causing prediction error to increase.) We choose to inspect officers with yearly allegation numbers larger than 4 (“bad officers”), and officers with yearly allegation numer lesser or equal to 1 (“good officers”), and see how our model perceives their related officers.

We choose 1 and 4 as two thresholds, on the basis of our first research question in Checkpoint 2:

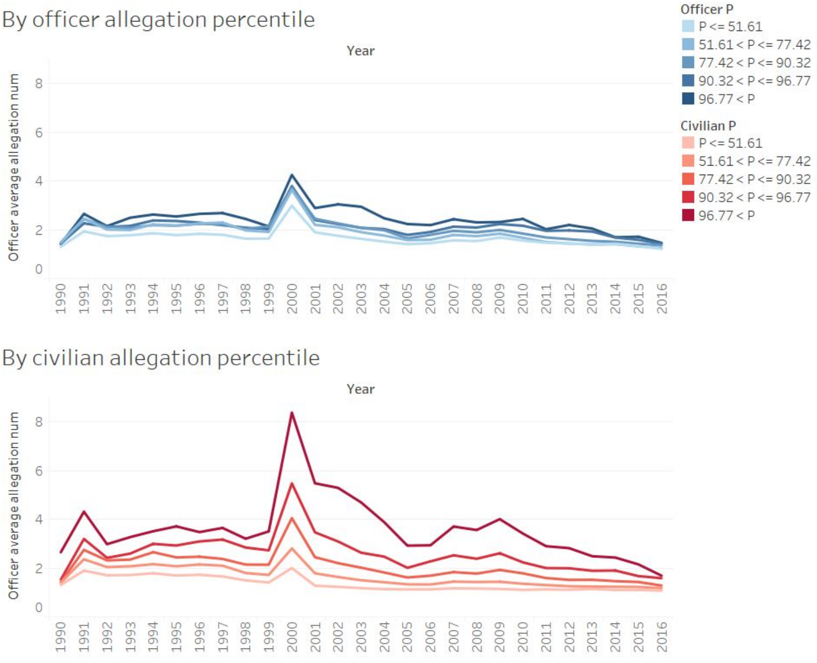


Figure 1.3.2.1 Q1 in checkpoint 2.

From 2006 to 2009, officers with civilian allegation percentiles larger than 96.77, their average yearly allegation number is approximately 4, and most officers with P <= 77.42 are below 2 (or even 1 in this case). Therefore we choose 4 and 1.

Since there are still hundreds or even thousands of such officers, we choose to draw 3 officers with non-empty co-accusal relations from our dataset for a closer inspection. Unfortunately, since many “good” officers do not have any co-accusal relations with other officers, they are filtered out and we only have one such good officer in this case. Data in table 1.3.2.1 and figure 1.3.2.2 are not produced from the same training session, but they are close.

Our first finding is that our GAT model puts **nearly equal attention** on all neighbors, no matter whether the current officer is a bad officer or a good officer. This observation is contradictory to our initial belief that our model will focus on some specific nodes more than other nodes. The table below shows the average attention weight and attention variance for each inspected officer:

Officer: 31873 (bad)	Neighbor num: 18 Neighbors: [1662, 3605, 6114, 8993, 14442, 14598, 15353, 15592, 16567, 17176, 19813, 20885, 21133, 23786, 29299, 29511, 30409, 31904] Attention mean of neighbors: 0.0555555559694767

	Attention variance of neighbors: 2.1776411813334562e-05
Officer: 16550 (bad)	Neighbor num: 2 Neighbors: [25994, 28529] Attention mean of neighbors: 0.5 Attention variance of neighbors: 8.382755368074868e-06
Officer: 28878 (bad)	Neighbor num: 2 Neighbors: [6548, 8442] Attention mean of neighbors: 0.5 Attention variance of neighbors: 0.0003526972432155162
Officer: 5417 (good)	Neighbor num: 2 Neighbors: [1883, 9262] Attention mean of neighbors: 0.5 Attention variance of neighbors: 0.0011290998663753271

Table 1.3.2.1 Officer info.

We further multiplied the computed attention weight to neighbor officers to simulate the accumulated feature vector our neural network will see:

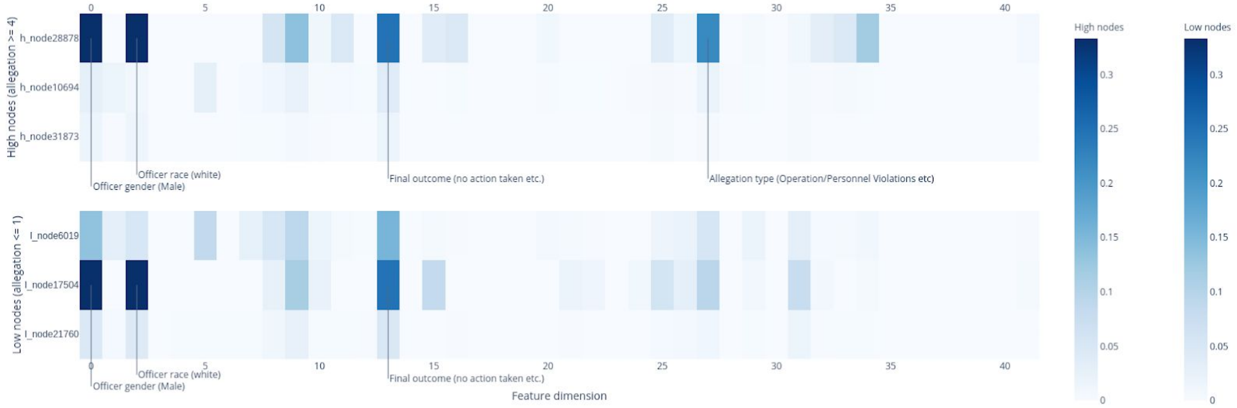


Figure 1.3.2.2 Aggregated neighbor features.

For “bad” officers (h_node) and “good” officers (l_node), their neighbor view do share some similarities:

1. Focus on dimension 0 (Officer gender, male).

2. Focus on dimension 2 (Officer race, white).
3. Focus on dimension 13 (Allegation final outcome, no action taken, etc).

And the feature weight distribution between dimension 8 and dimension 10 (career time, age, allegation num), are also very similar, although other details may vary (good officers have some weight on dimension 25 to 31, which mean various allegation types). This is also interesting because we originally expected averaged neighbor feature vectors to have different high feature weights between “bad” officers and “good" officers.

1.4 Summary

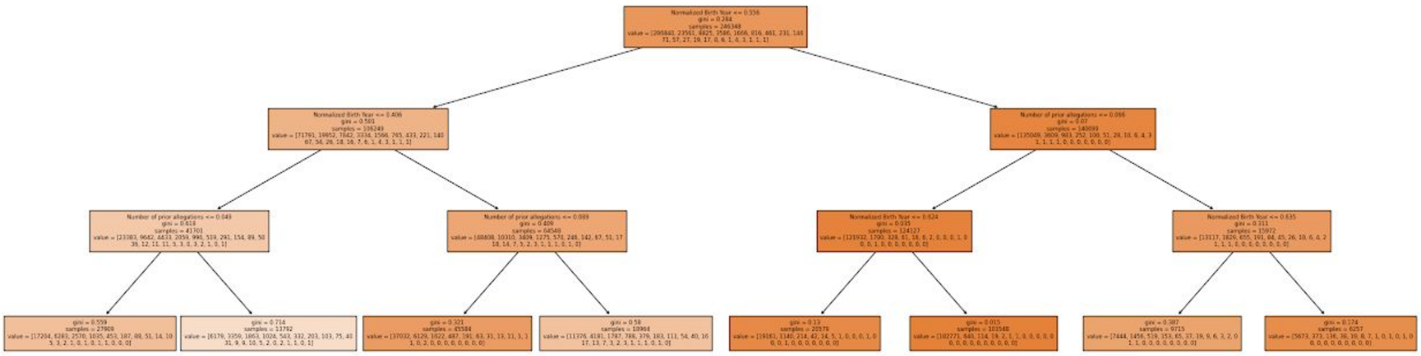
In this section we have discussed how our team uses the GAT model to predict the number of the next year's allegation number of each officer. In this process, we have poked into various data and weight vectors computed by our model and performed close and detailed analysis on specific officer cases. Several discoveries have been made, but we still need more background information on officer behaviours to further understand why our model chooses these features as key features, which dictates the output value.

2. Using a decision tree, first we will preprocess features of officers in the three dimensions discussed above and group them into bins. Then we can utilize the tree to not only make predictions such as the number of complaints, allegations, sustained allegations, and use of force reports that officers will have in the following year, but also extract most important, defining features by extracting the topmost level of decision trees.

What we ended up with for this model, is a decision tree model that predicts the number of allegations an officer is likely to be involved in for a particular year, given a dense set of officer features (e.g. gender, race, appointment year, resignation year, etc.) and information about past allegations the officer has been involved in (e.g. allegation outcome, allegation category, complainant demographic information, etc.). Information included in the training and test set ranged in date from 2000 to 2013. In making our evaluations, we examined two decision tree variants: one using the gini classification impurity and one using entropy classification impurity. However, we will not dive into the details of the differences, as both models performed with exactly the same performance. The other major parameter used in the decision tree was the maximum depth of 3 (both variants), allowing for a fairly comprehensive analysis, without being too computationally demanding. The performance of the model was as follows for each of the data sets:

Data Set	**Accuracy**
Training	83.9%
Validation	84.1%
Testing	83.7%

As is clearly evident from the table above, the accuracy for all three datasets is consistently around 84%. The fact that such a notably high accuracy is achieved with this model, particularly on the held out sets, indicates that there is a strong correlation between the included officer/allegation features, and the number of allegations an officer is likely to have.



The above visualization of the gini tree (the entropy is almost identical) shows that, interesting enough, officer birth year is the most influential factor in predicting how many allegations an officer is expected to have for a given year. This indicates that age might be a major factor in officer misconduct. Specifically, officers born later (younger officers) are more likely to receive misconduct allegations (apologies for the small font in figure 2.1, please see image in src directory for bigger text). The second most prominent feature in predicting the number of future officer allegations is the number of prior allegations. Not surprisingly, a greater number of prior allegations contributes to a greater prediction of future allegations. An interesting area of future analysis might be to increase the depth of the tree and examine what other officer features play a major role in their likelihood of receiving future misconduct allegations.