(Appendix)
Disentangling, Amplifying, and Debiasing: Learning Disentangled Representations for Fair Graph Neural Networks

A Detailed Equations

Fairness Analysis Metrics

Embedding-Level Fairness Metrics. A key measure of bias in GNNs is the distribution difference of node attributes or learned embeddings between subgroups (Dong et al. 2022). A greater distribution difference can lead to more-biased outcomes by making it easier for GNNs to infer a node's sensitive attribute (Buyl and Bie 2020; Chen et al. 2024). To assess this, we use two metrics related to distribution difference, which are proposed by Dong et al. (2022):

- **Attribute Bias (AttrBias)** measures the distribution difference in the attribute matrix between subgroups.

$$b_{attr} = \frac{1}{M}\sum_{m=1}^{M} W(\text{pdf}(X_m^0), \text{pdf}(X_m^1)),$$

  where X_m^i represents the values of the m-th dimension ($1 \leq m \leq M$) of X for nodes in subgroup i, and W denotes the Wasserstein-1 distance between the probability density functions (PDFs) of two subgroups.

- **Structure Bias (StruBias)** measures the distribution difference of node embeddings between subgroups after applying a GNN method.

$$b_{stru} = \frac{1}{M}\sum_{m=1}^{M} W(\text{pdf}(R_m^0), \text{pdf}(R_m^1)),$$

  where $\mathbf{R} = \mathbf{M}_H\mathbf{X}_{norm}$ represents the reachability matrix. Here, $\mathcal{R}_m^0$ and $\mathcal{R}_m^1$ correspond to the m-th attribute value sets in $\mathbf{R}$ for nodes with sensitive attributes of 0 and 1, respectively. To compute $\mathbf{R}$, we define a normalized adjacency matrix with re-weighted self-loops as $\mathbf{P}_{norm} = \alpha\mathbf{A}_{norm} + (1-\alpha)\mathbf{I}$, where $\mathbf{A}_{norm}$ and $\mathbf{I}$ represent the symmetric normalized adjacency matrix and the identity matrix, respectively; α is a hyperparameter ranging from 0 to 1. The propagation matrix is given by $\mathbf{M}_H = \sum_{h=1}^{H}\beta_h\mathbf{P}_{norm}^h$, where H indicates the number of hops and β indicates a discount factor that reduces the weight of propagation from more distant neighbors.

Neighborhood-Level Fairness Metrics. In GNNs, the aggregation of neighborhood information heavily influences node embeddings (Wu et al. 2021). When intra-subgroup connections dominate and inter-subgroup connections are sparse, GNNs tend to learn embeddings that are similar within subgroups and distinct across subgroups (Chen et al. 2024; Wang et al. 2022; Dai and Wang 2021). We measure this effect by using two neighborhood-level metrics:

- **Homophily Ratio (HomoRatio)** measures the proportion of intra-subgroup edges to all edges in a graph dataset.

$$h_G = \frac{\sum_i C_{ii}}{\sum_{i,j} C_{ij}},$$

  where C_{ij} represents the number of edges between nodes in subgroups i and j. A value close to 1 indicates a high concentration of intra-subgroup edges, while a value near 0 indicates more inter-subgroup edges.

- **Neighborhood Fairness (NbhdFair)** captures the average entropy of each node's neighbors in a graph dataset.

$$F_G = \frac{1}{|V|}\sum_{i\in V} F_{p_i},$$

  where $F_{p_i} = -\sum_s p_{is}\log p_{is}$ represents the entropy of node i's neighbors, and p_{is} denotes the probability that node i's neighbors belong to subgroup s. A higher value indicates a more balanced neighborhood distribution, while a lower value indicates bias toward certain subgroups.

Fairness Evaluation Metrics

We evaluate fairness by using *Statistical Parity* (SP) (Dwork et al. 2012) and *Equality of Opportunity* (EO) (Hardt, Price, and Srebro 2016), where lower values indicate better model fairness. Specifically, SP measures the difference in prediction rates between subgroups with different sensitive attributes:

$$\Delta SP = |P(\hat{\mathbf{y}} = 1|\mathbf{s} = 0) - P(\hat{\mathbf{y}} = 1|\mathbf{s} = 1)|, \quad (12)$$

where $\hat{\mathbf{y}}$ represents the predicted labels and $\mathbf{s}$ indicates the sensitive attribute values.

On the other hand, EO measures the difference in true positive rates across different sensitive attributes:

$$\Delta EO = |P(\hat{\mathbf{y}} = 1|\mathbf{s} = 0, \mathbf{y} = 1) - P(\hat{\mathbf{y}} = 1|\mathbf{s} = 1, \mathbf{y} = 1)|, \quad (13)$$

where $\mathbf{y}$ indicates the actual labels.

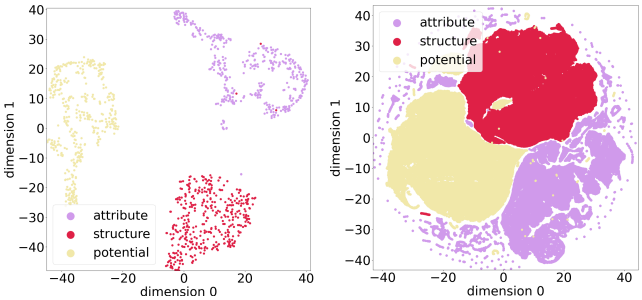
B Detailed Implementation Details

The experiments were run on a Linux system with a 36-core CPU, RTX A6000 48GB GPU, and 256GB of RAM. All experiments were conducted by using five different seed settings, and we report the average accuracy. We used GCN as the backbone for both the competitors and **DAB-GNN**, carefully tuning their hyperparameters via grid search.

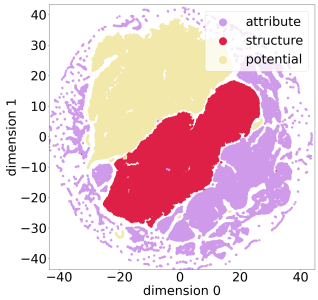
For **DAB-GNN**, we used a 3-layer GCN with a hidden layer of size 16, a gradient penalty (Gulrajani et al. 2017) of 10, 1,000 epochs, the embedding dimensionality of 48, and a weight decay of 1e-5. The optimal values for the remaining hyperparameters are as follows:

- **Learning Rates**: 0.001 for *AbDisen* (all datasets); 0.0003 for *SbDisen* (all datasets); 0.0005 for *PbDisen* (NBA) and 0.003 for *PbDisen* (Recidivism, Credit, Pokec_n, Pokec_z)
- $\boldsymbol{\alpha}$: 1 (NBA, Recidivism, Pokec_n), 15 (Credit), 100 (Pokec_z)
- $\boldsymbol{\beta}$: 0.0003 (NBA, Pokec_z), 0.0005 (Recidivism), 0.005 (Pokec_n), 0.0095 (Credit)
- $\boldsymbol{k}$: 1 (Recidivism), 8 (Credit), 40 (NBA), 100 (Pokec_n, Pokec_z)

For hyperparameters of competitors, we used the best settings found via grid search. We tuned the hyperparameters over the following ranges:



(a) **NBA** (b) **Pokec_n**



(c) **Pokec_z**

Figure I: Visualization of disentangled node embeddings by using t-SNE: attribute (purple), structure (red), and potential (yellow) bias embeddings.

- **FairGNN**: $\alpha \in \{0.0001, 0.001, 0.01, 0.1, 1\}$ and $\beta \in \{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$
- **NIFTY**: `sim_coeff` $\in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **EDITS**: $u_1 \in \{0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7, 1\}$ and $u_2 \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$; the proportion was tuned among the values $\{0.02, 0.29, 0.015\}$
- **FairVGNN**: `ratio` $\in \{0, 0.5, 1\}$ and the `clip_e` $\in \{0.01, 0.1, 1\}$
- **CAF**: `causal_coeff`, `disentangle_coeff`, and `rec_coeff` parameters were all tuned $\in \{0.0, 0.5, 1.0, 1.5, 2.0\}$
- **GEAR**: `sim_coeff` $\in \{0, 0.25, 0.5, 0.75, 1.0\}$, `subgraph_size` $\in \{5, 10, 15, 20, 25\}$, and `n_order` $\in \{0, 1, 3, 5, 10\}$
- **BIND**: we selected the best trade-off value among the available budgets
- **PFR-AX**: `pfr_nn_k` $\in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, `pfr_A_nn_k` $\in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, and `pfr_gamma` $\in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **PostProcess**: `flip_frac` $\in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **FairSIN**: `delta` $\in \{0.1, 0.5, 1, 2, 5, 10\}$

C Further Experimental Results

Disentangled Embeddings Analysis

Figure I shows the results of visualizing the disentangled embeddings on NBA, Pokec_z, and Pokec_n datasets.

Hyperparameter Analysis

Figure II shows the results of the hyperparameters α and β, while Figure III shows the results of a hyperparameter k. These results are displayed on the following pages.

Model Complexity Analysis

Computational Cost: DAB-GNN was trained faster than FairVGNN but slower than NIFTY, both of which are the best competitors. On the Recidivism dataset, the training times for DAB-GNN, FairVGNN, and NIFTY were approximately 233, 385, and 50 seconds, respectively. However, as shown in Table 4, DAB-GNN significantly outperforms NIFTY in fairness across all datasets, achieving an average SP improvement of 78.51% – highlighting the small trade in complexity as worthwhile.

Scalability: We observed a generally linear increase in training time with the number of nodes in DAB-GNN. On the Recidivism dataset, the training times at 20%, 40%, 60%, 80%, and 100% node sampling were approximately 129, 150, 185, 220, and 233 seconds, respectively.

Parameter Count: Among FGNN methods, DAB-GNN has the third-highest number of learnable parameters, following EDITS and GEARS. On the Recidivism dataset, DAB-GNN, GEARS, and EDITS had approximately 350 thousand, 620 thousand, and 350 million parameters, respectively. Despite this, DAB-GNN's streamlined GNN architecture supports competitive computational cost and scalability, as shown in prior experiments.

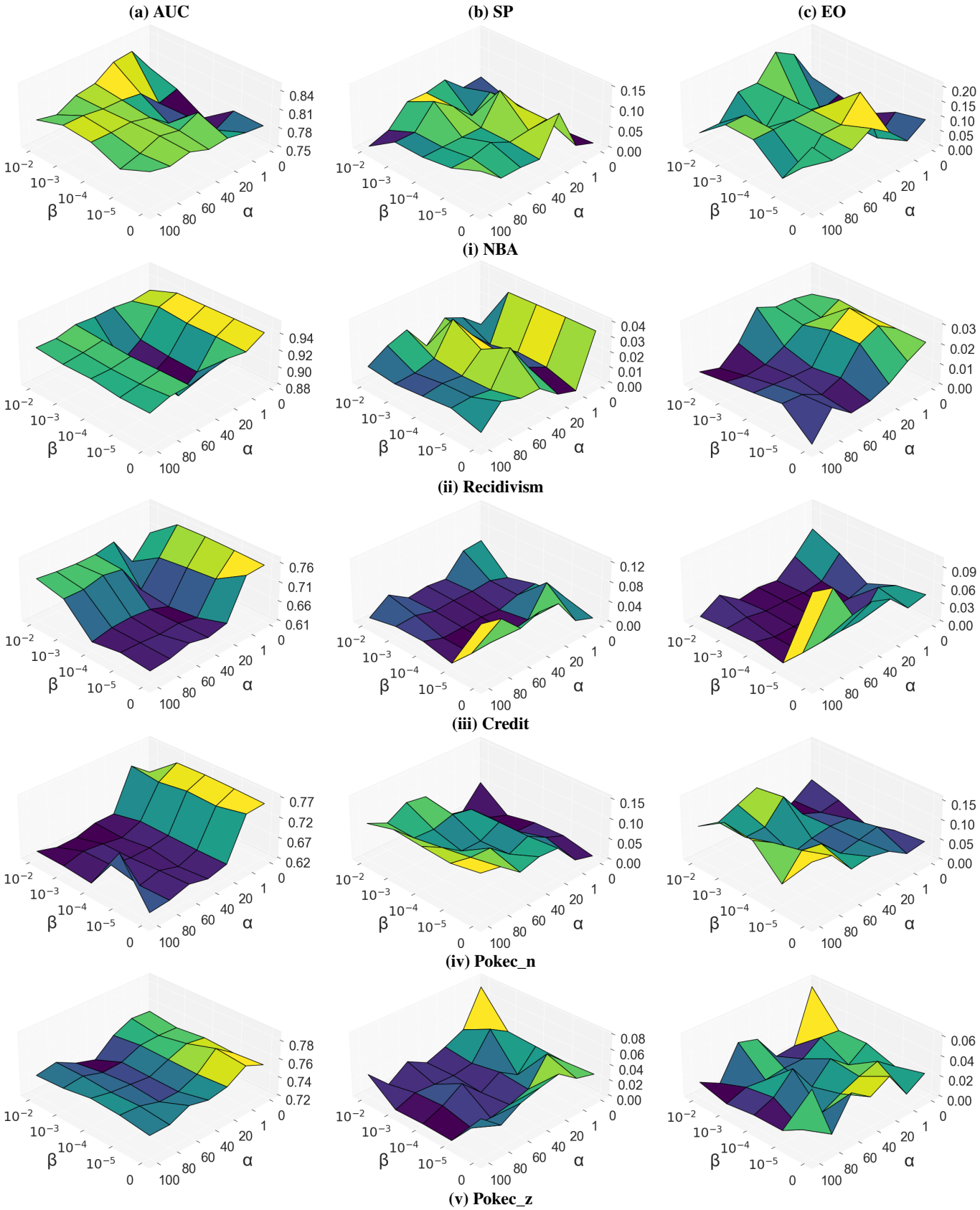


Figure II: The effects of α and β on AUC (↑), SP (↓), and EO (↓) across different datasets.

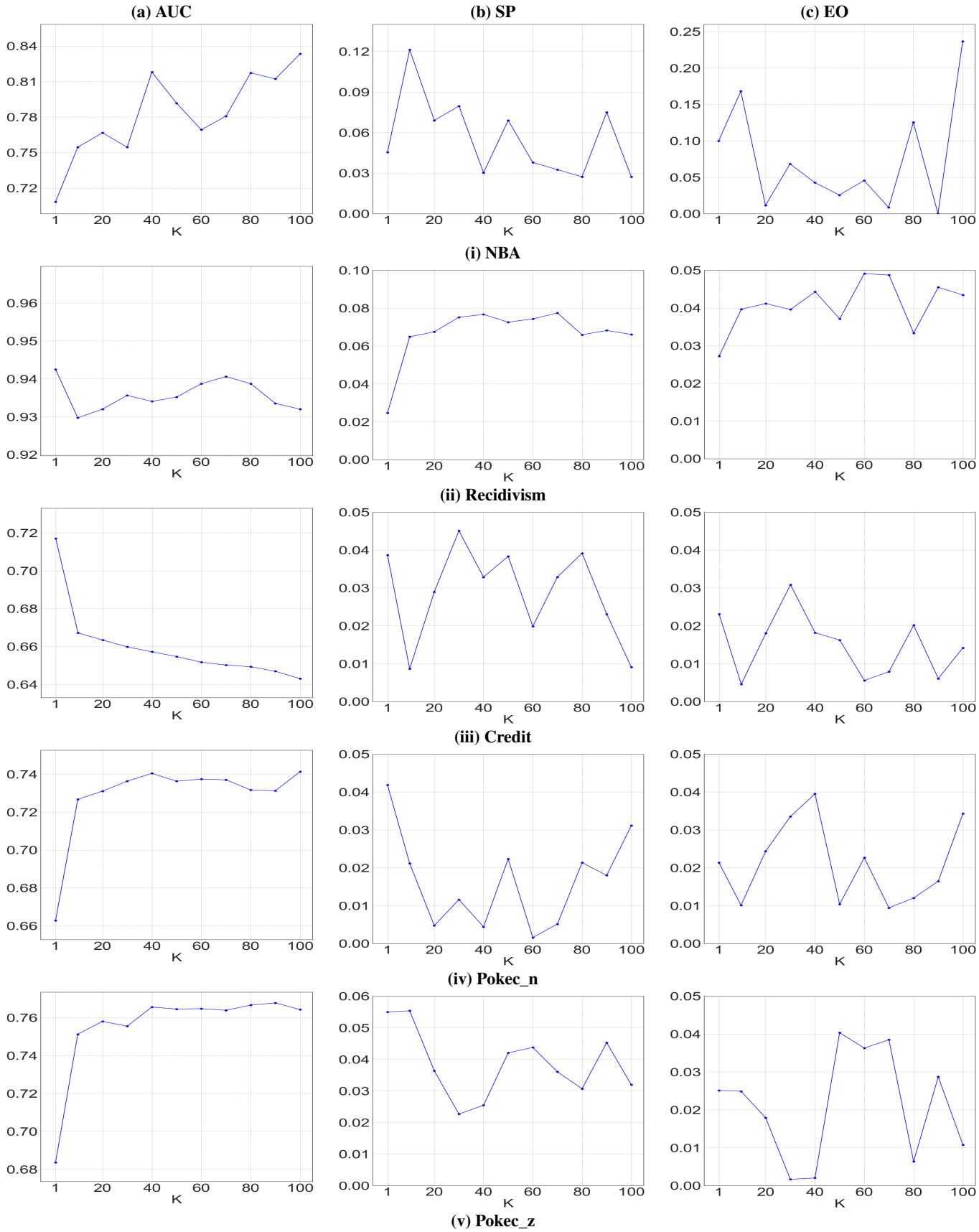


Figure III: The effects of k on AUC (↑), SP (↓), and EO (↓) across different datasets.