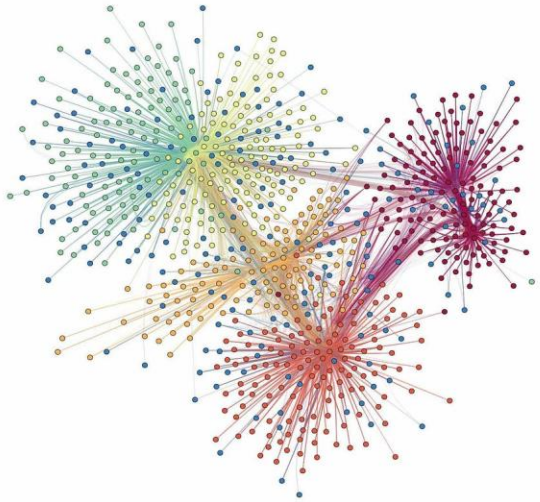
Node Identifiers: Compact, Discrete Representations for Efficient Graph Learning

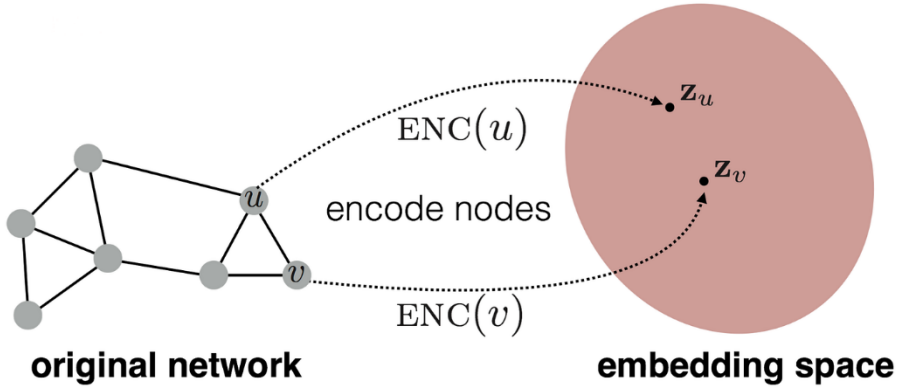
Yuankai Luo, Hongkang Li, Qijiong Liu, Lei Shi, Xiao-Ming Wu

ICLR 2025

Introduction: Challenges in GNNs

- Efficiency in Large-Scale Graphs
  - Message-passing requires full graph loading → high inference compute costs (billions of edges).
  - High-dimensional embeddings (128-256 dim) → storage and interpretability issues.

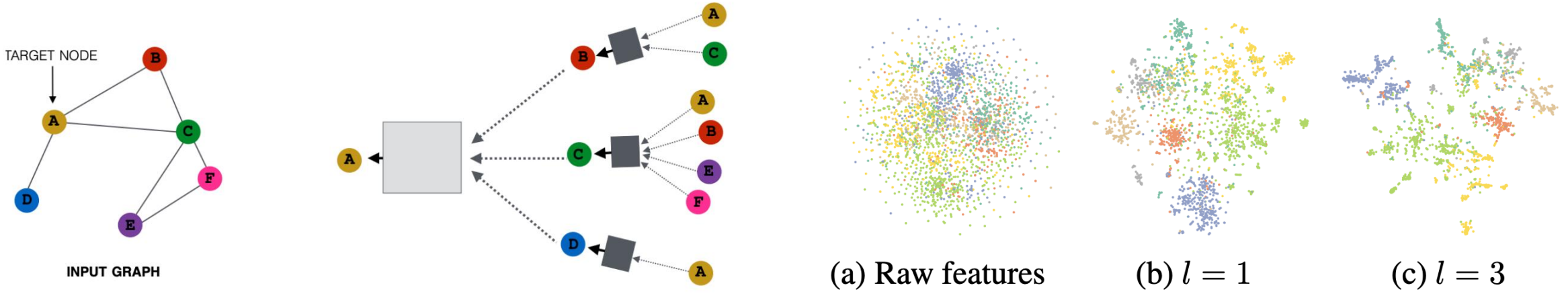




Message-passing Mechanism

The message-passing mechanism creates new node representations, where each node gathers information from its neighbors and combines it to update its own embedding.

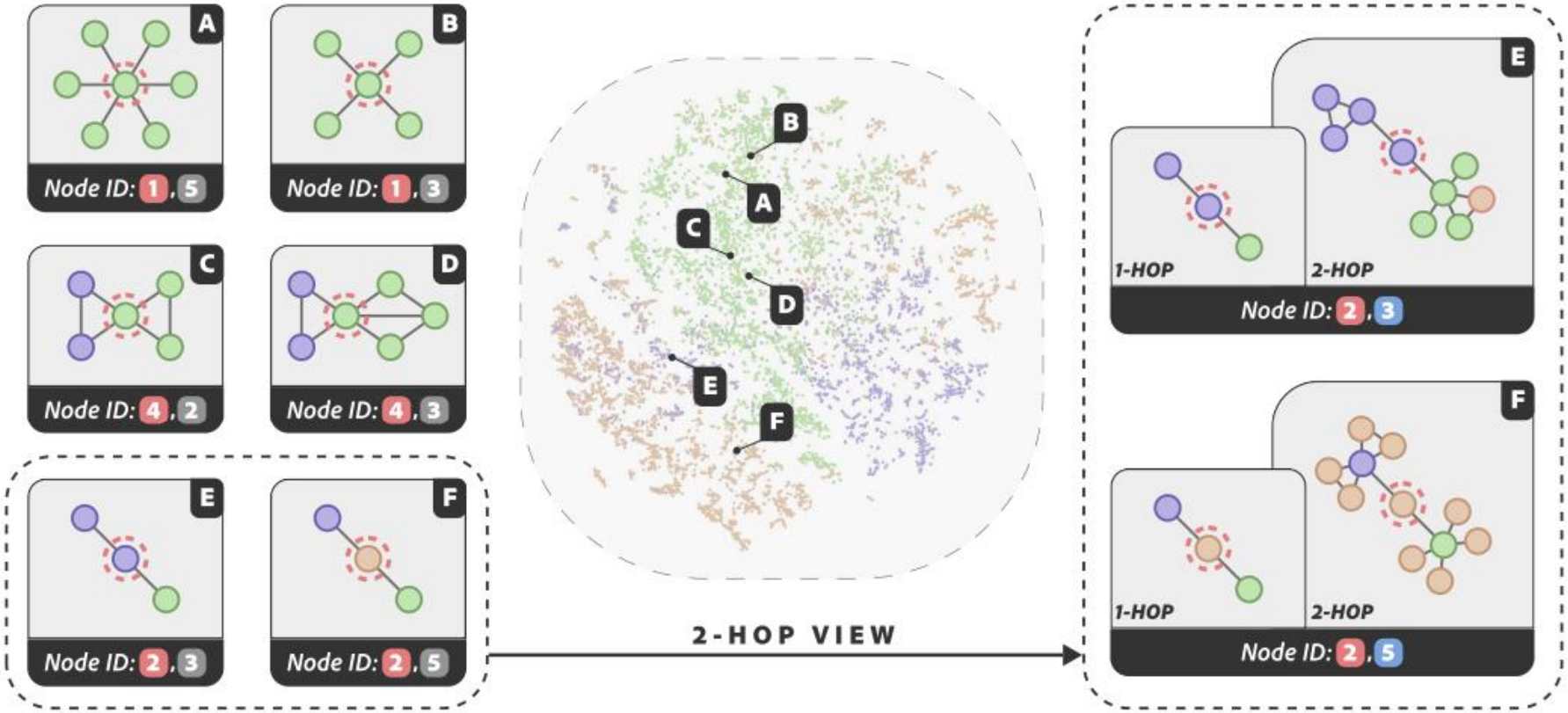
Due to the smoothing effect of message passing, the node representations generated by GNNs exhibit distinct clustering patterns.



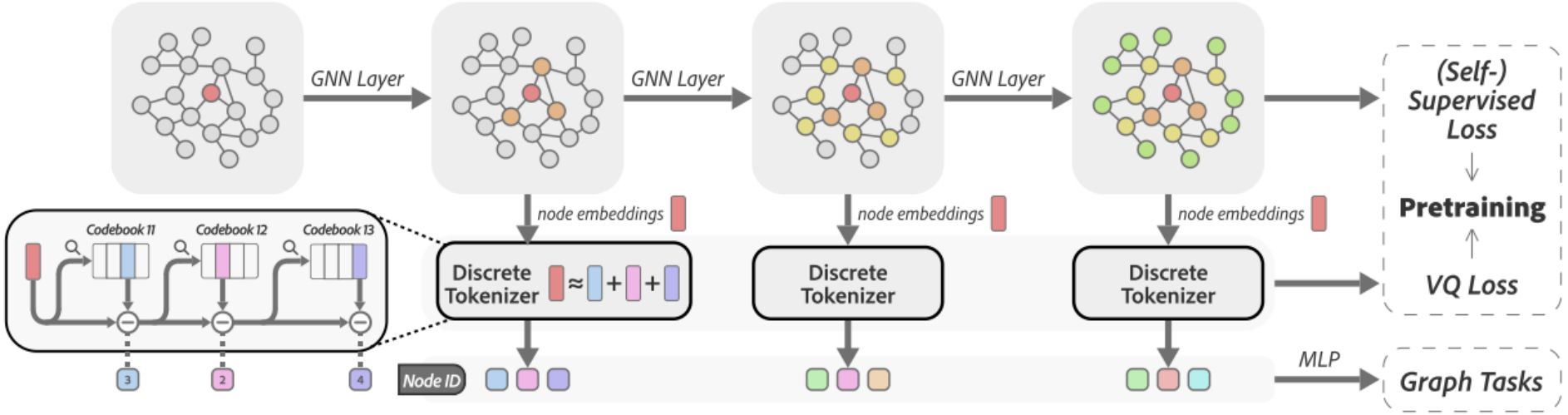
(a) Raw features (b) $l = 1$ (c) $l = 3$

Our Solution: Node IDs

We present a novel end-to-end framework that generates highly compact (typically 6-15 dimensions), discrete (int4 type), and interpretable node representations—termed node identifiers (node IDs).



Methodology: NID Framework



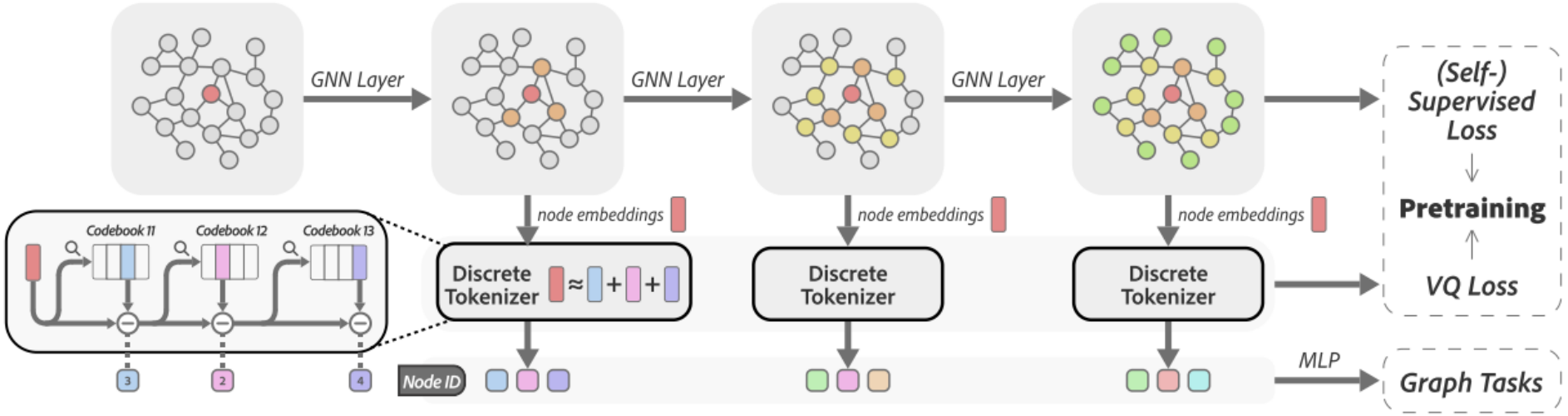
1. Node ID Generation:

- Residual VQ (RVQ) quantizes embeddings per layer into codewords (integer indices) tuples: (Each layer has M codebooks (RVQ))

$$\text{Node\_ID}(v) = \left(c_{\{11\}}, \ldots, c_{\{1M\}}, \ldots, c_{\{L1\}}, \ldots, c_{\{LM\}}\right)$$

2. Downstream Task Application.

Node ID Generation via RVQ



$$\mathcal{L}_{\text{NID}} = \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\text{VQ}},$$

$$\mathcal{L}_{\text{VQ}} = \sum_{l=1}^{L} \sum_{m=1}^{M} \|\text{sg}(\boldsymbol{r}_{lm}) - \boldsymbol{e}_{c_{lm}}^{lm}\| + \beta \|\boldsymbol{r}_{lm} - \text{sg}(\boldsymbol{e}_{c_{lm}}^{lm})\|$$

No reconstruction loss → guided by VQ loss.

Theoretical Analysis

Theorem 1. *The optimizer $\boldsymbol{C}^*$ of VQ objective (7) satisfies that, for any $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$, $u, v \in \mathcal{V}$ with different labels, Node_ID$(u) \neq$ Node_ID(v). Then, as long as $\mathcal{V}_R$ uniformly include node IDs from all the classes, by training the linear head $\boldsymbol{V}$ with sufficient gradient descent steps, we can achieve that the classification error $\mathbb{1}[y_v \neq \arg\max_{i\in[P]} \hat{\boldsymbol{p}}_{v,i}] = 0$ for any $v \in \mathcal{V}$.*

Theorem 1 illustrates that the optimized $C*$ of VQ objective (7) ensures that the obtained IDs from different classes are distinct. Then, we demonstrate that with node IDs in the training set, a linear head can be learned to achieve a zero classification error.

Experiments: Key Results

Supervised Tasks (Node/Graph Classification, Link Prediction):

- Matches or outperforms SOTA GNNs.
- 10-1000x Faster Inference:

ogbn-products: 11.9s → 0.7ms per inference.

Table 1: Node classification results in supervised representation learning over homophilic and heterophilic graphs (%). The baseline results are primarily taken from Polynormer (Deng et al., 2024).

Transductive	Cora	CiteSeer	PubMed	Computer	Photo	CS	Physics	WikiCS	Squirrel	Chameleon	Ratings	Questions
# nodes	2,708	3,327	19,717	13,752	7,650	18,333	34,493	11,701	2223	890	24,492	48,921
# edges	5,278	4,732	44,324	245,861	119,081	81,894	247,962	216,123	46,998	8,854	32,927	153,540
Metric	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	ROC-AUC↑
GPRGNN	87.95 ± 1.18	77.13 ± 1.67	87.54 ± 0.38	89.32 ± 0.29	94.49 ± 0.14	95.13 ± 0.09	96.85 ± 0.08	78.12 ± 0.23	38.95 ± 1.99	39.93 ± 3.30	44.88 ± 0.34	55.48 ± 0.91
APPNP	87.87 ± 0.82	76.53 ± 1.16	88.43 ± 0.15	90.18 ± 0.17	94.32 ± 0.14	94.49 ± 0.07	96.54 ± 0.07	78.87 ± 0.11	36.88 ± 1.27	41.62 ± 3.13	52.74 ± 0.73	77.82 ± 1.31
SGFormer	87.83 ± 0.92	77.24 ± 0.74	89.31 ± 0.54	92.42 ± 0.66	95.58 ± 0.36	95.71 ± 0.24	96.75 ± 0.26	80.05 ± 0.46	42.65 ± 2.41	45.21 ± 3.72	54.14 ± 0.62	73.81 ± 0.59
Polynormer	88.11 ± 1.08	76.77 ± 1.01	87.34 ± 0.43	93.18 ± 0.18	96.11 ± 0.23	95.51 ± 0.29	97.22 ± 0.06	79.53 ± 0.83	40.87 ± 1.96	41.82 ± 3.45	54.46 ± 0.40	78.92 ± 0.89
Graph-MLP	87.06 ± 1.38	76.43 ± 1.44	88.93 ± 0.63	90.78 ± 0.41	95.43 ± 0.76	94.68 ± 0.28	95.45 ± 0.24	75.35 ± 0.55	-	-	-	-
VQGraph	86.11 ± 1.26	75.64 ± 0.92	88.03 ± 0.63	90.28 ± 0.47	94.98 ± 0.59	93.82 ± 0.17	95.93 ± 0.28	77.92 ± 0.61	-	-	-	-
GCN	88.77 ± 0.61	77.53 ± 0.92	90.04 ± 0.25	93.78 ± 0.31	96.14 ± 0.21	95.94 ± 0.28	97.36 ± 0.07	80.91 ± 0.81	44.50 ± 1.92	46.11 ± 3.16	53.57 ± 0.32	77.40 ± 1.07
NID$_{GCN}$	87.88 ± 0.69	76.89 ± 1.09	89.42 ± 0.44	93.41 ± 0.08	96.17 ± 0.04	95.52 ± 0.10	97.34 ± 0.04	78.55 ± 0.15	**45.09** ± 1.72	**46.29** ± 2.92	53.55 ± 0.13	96.85 ± 0.10
GAT	88.22 ± 1.24	77.08 ± 0.84	89.47 ± 0.25	93.53 ± 0.18	96.27 ± 0.15	94.46 ± 0.14	97.17 ± 0.09	80.98 ± 0.83	38.72 ± 1.46	43.44 ± 3.00	54.88 ± 0.74	78.35 ± 1.16
NID$_{GAT}$	87.35 ± 0.57	76.13 ± 1.35	88.97 ± 0.36	93.38 ± 0.16	**96.47** ± 0.27	94.75 ± 0.16	97.13 ± 0.08	79.56 ± 0.43	37.68 ± 2.04	42.83 ± 3.42	**54.92** ± 0.42	**97.03** ± 0.02

Table 2: Node classification results in supervised representation learning on large-scale graphs (%).

Transductive	ogbn-proteins	ogbn-arxiv	ogbn-products	pokec
# nodes	132,534	169,343	2,449,029	1,632,803
# edges	39,561,252	1,166,243	61,859,140	30,622,564
Metric	ROC-AUC↑	Accuracy↑	Accuracy↑	Accuracy↑
GPRGNN	75.68 ± 0.49	71.10 ± 0.12	79.76 ± 0.59	78.83 ± 0.05
LINKX	71.37 ± 0.58	66.18 ± 0.33	71.59 ± 0.71	82.04 ± 0.07
GraphGPS	76.83 ± 0.26	70.97 ± 0.41	OOM	OOM
SGFormer	79.53 ± 0.38	72.63 ± 0.13	74.16 ± 0.31	73.76 ± 0.24
Polynormer	75.97 ± 0.47	71.82 ± 0.23	82.97 ± 0.28	85.95 ± 0.07
SAGE	79.43 ± 0.75	72.67 ± 0.31	83.27 ± 0.35	85.97 ± 0.21
Infer. Time	158.1ms	416.5ms	11.9s	129.6s
Storage Space	129.4MB	165.7MB	1.9GB	1.6GB
NID$_{SAGE}$	76.78 ± 0.59	70.52 ± 0.14	81.83 ± 0.26	85.63 ± 0.31
Infer. Time	0.4ms	0.3ms	0.7ms	27.1ms
Storage Space	0.4MB	1.2MB	17.5MB	16.4MB

Table 3: Graph-level performance in supervised representation learning from LRGB.

Inductive	Peptides-func	Peptides-struct
Avg. # nodes	150.9	150.9
Avg. # edges	307.3	307.3
Metric	AP↑	MAE↓
GT	0.6326 ± 0.0126	0.2529 ± 0.0016
GraphGPS	0.6535 ± 0.0041	0.2500 ± 0.0012
GRIT	0.6988 ± 0.0082	0.2460 ± 0.0012
Exphormer	0.6527 ± 0.0043	0.2481 ± 0.0007
Graph ViT	0.6970 ± 0.0080	0.2449 ± 0.0016
GCN	0.6762 ± 0.0053	0.2512 ± 0.0007
Infer. Time	471.1ms	424.9ms
NID$_{GCN}$	0.6608 ± 0.0058	0.2589 ± 0.0014
Infer. Time	0.4ms	0.4ms

Node IDs, typically comprising 6 to 15 int4 integers, serve as effective node representations.

Experiments: Key Results

Unsupervised Tasks (Graph Clustering, Unsupervised Prediction):

- Matches or outperforms SOTA GNNs.
- Faster Clustering.

Table 5: Attributed graph clustering results; normalized mutual information, and F1-score (%).

	Cora		CiteSeer		PubMed		Computer		Photo		Physics		ogbn-arxiv	
	NMI↑	F1↑	NMI↑	F1↑	NMI↑	F1↑	NMI↑	F1↑	NMI↑	F1↑	NMI↑	F1↑	NMI↑	F1↑
SBM	36.2	30.2	15.3	19.1	16.4	16.7	48.4	34.6	59.3	47.4	45.4	30.4	31.9	28.3
AGC	34.1	28.9	25.5	27.5	18.2	18.4	51.3	35.3	59.0	44.2	-	-	-	-
SDCN	27.9	29.9	31.4	41.9	19.5	29.9	24.9	45.2	41.7	45.1	50.4	39.9	15.3	28.8
DAEGC	8.3	13.6	4.3	18.0	4.4	11.6	42.5	37.3	47.6	45.0	-	-	-	-
NOCD	46.3	36.7	20.0	24.1	25.5	20.8	44.8	37.8	62.3	60.2	51.9	28.7	20.7	38.2
DiffPool	32.9	34.4	20.0	23.5	20.2	26.3	22.1	38.3	35.9	41.8 -	-	-	-	
MinCut	35.8	25.0	25.9	20.1	25.4	15.8	-	-	-	-	48.3	24.9	36.0	27.1
Ortho	38.4	26.6	26.1	20.5	20.3	13.9	-	-	-	-	44.7	23.7	35.6	26.7
DMoN	48.8	48.8	33.7	43.2	29.8	33.9	49.3	45.4	63.3	61.0	56.7	42.4	37.6	45.7
DGCluster	62.1	54.5	41.0	32.2	32.6	34.6	60.4	52.2	77.3	75.9	65.7	49.2	31.2	32.4
Clustering Time	93.6ms		119.6ms		405.5ms		286.1ms		204.6ms		547.4ms		2.7s	
$NID_{DGCluster}$	**70.5**	**73.9**	**54.1**	**63.3**	**40.6**	**50.9**	**62.1**	**58.2**	75.6	75.4	**69.8**	**65.4**	32.4	35.6
Clustering Time	78.3ms		77.2ms		292.5ms		223.6ms		140.6ms		442.0ms		1.8s	

Table 6: Node classification results in unsupervised representation learning (%).

Metric	Cora Accuracy↑	CiteSeer Accuracy↑	PubMed Accuracy↑	dim
GAE	71.5 ±0.4	65.8 ±0.4	72.1 ±0.5	16
DGI	82.3 ±0.6	71.8 ±0.7	76.8 ±0.6	512
MVGRL	83.5 ±0.4	73.3 ±0.5	80.1 ±0.7	512
InfoGCL	83.5 ±0.3	73.5 ±0.4	79.1 ±0.2	512
CCA-SSG	84.0 ±0.4	73.1 ±0.3	81.0 ±0.4	512
MLP	57.8 ±0.5	54.7 ±0.4	73.3 ±0.6	500
GraphMAE	84.2 ±0.4	73.4 ±0.4	81.1 ±0.4	512
NID_{MAE}	80.8 ±0.7	**74.2 ±0.6**	76.4 ±0.8	**6**

Table 7: Graph classification results in unsupervised representation learning on TUDataset; Accuracy (%).

# graphs Avg. # nodes	NCI1 4,110 29.8	PROTEINS 1,113 39.1	DD 1,178 284.3	MUTAG 188 17.9	COLLAB 5,000 74.5	RDT-B 2,000 429.7	RDT-M5K 4,999 508.5	IMDB-B 1,000 19.8
InfoGraph	76.2 ±1.0	74.4 ±0.3	72.8 ±1.7	89.0 ±1.1	70.6 ±1.1	82.5 ±1.4	53.4 ±1.0	73.0 ±0.8
MVGRL	-	-	-	89.7 ±1.1	-	84.5 ±0.6	-	74.2 ±0.7
JOAO	78.3 ±0.5	74.0 ±1.1	77.4 ±1.1	87.6 ±0.7	69.3 ±0.3	86.4 ±1.4	56.0 ±0.2	70.8 ±0.2
GraphMAE	80.4 ±0.3	75.3 ±0.4	-	88.1 ±1.3	80.3 ±0.5	88.0 ±0.2	-	75.5 ±0.6
AD-GCL	69.6 ±0.5	73.5 ±0.6	74.4 ±0.5	-	73.3 ±0.6	85.5 ±0.7	53.0 ±0.8	71.5 ±1.0
GraphCL	77.8 ±0.4	74.3 ±0.4	78.6 ±0.4	86.8 ±1.3	71.3 ±1.1	89.5 ±0.8	55.9 ±0.2	71.1 ±0.4
NID_{CL}	75.9 ±0.6	75.1 ±0.5	77.8 ±1.1	88.6 ±1.7	76.9 ±0.3	**90.7** ±0.9	55.0 ±0.5	72.3 ±1.2
AutoGCL	82.0 ±0.2	75.8 ±0.3	77.5 ±0.6	88.6 ±1.0	70.1 ±0.6	88.5 ±1.4	56.7 ±0.1	73.3 ±0.4
$NID_{AutoGCL}$	78.2 ±1.5	**75.9** ±0.6	77.2 ±0.9	**90.4** ±0.8	74.5 ±1.1	89.8 ±0.7	54.2 ±0.6	72.4 ±0.8

Node IDs, typically comprising 6 to 15 int4 integers, serve as effective node representations.

Analysis of Node IDs

Table 8: Comparison of codebook usage rates (%).

Usage rate↑	Cora	CiteSeer	PubMed
VQGraph	1.3	0.8	18.1
$\textbf{NID}_{\text{GCN}}$	84.7	97.9	79.1
$\textbf{NID}_{\text{GCN}(M=1)}$	83.3	81.3	78.1

Table 9: Average GEDs of 1-hop subgraphs among nodes.

GEDs↓	Cora	CiteSeer	PubMed
Random	7.21	4.83	9.61
VQGraph	6.85	4.73	9.03
$\textbf{NID}_{\text{GCN}}$	6.15	3.89	6.22

Table 10: Accuracy vs Inference Time.

Metric	Computer		ogbn-products	
	Acc↑	Time↓	Acc↑	Time↓
GCN	93.78	119.6ms	82.33	12.8s
SAGE	93.59	95.7ms	83.27	11.9s
VQGraph	90.28	1.4ms	79.17	1.6ms
$\textbf{NID}_{\text{SAGE}}$	**93.32**	**0.5ms**	**81.83**	**0.7ms**

Avoiding Codebook Collapse:

It was observed that VQGraph suffers from severe codebook collapse. In contrast, NID achieves high codebook utilization, effectively avoiding codebook collapse.

Analysis of Node IDs

Table 8: Comparison of codebook usage rates (%).

Usage rate↑	Cora	CiteSeer	PubMed
VQGraph	1.3	0.8	18.1
$\textbf{NID}_{\text{GCN}}$	84.7	97.9	79.1
$\textbf{NID}_{\text{GCN}(M=1)}$	83.3	81.3	78.1

Table 9: Average GEDs of 1-hop subgraphs among nodes.

GEDs↓	Cora	CiteSeer	PubMed
Random	7.21	4.83	9.61
VQGraph	6.85	4.73	9.03
$\textbf{NID}_{\text{GCN}}$	6.15	3.89	6.22

Table 10: Accuracy vs Inference Time.

Metric	Computer		ogbn-products	
	Acc↑	Time↓	Acc↑	Time↓
GCN	93.78	119.6ms	82.33	12.8s
SAGE	93.59	95.7ms	83.27	11.9s
VQGraph	90.28	1.4ms	79.17	1.6ms
$\textbf{NID}_{\text{SAGE}}$	**93.32**	**0.5ms**	**81.83**	**0.7ms**

Subgraph Retrieval:

Using the Graph Edit Distance (GED) of Node IDs for subgraph matching yields superior results compared to the existing VQGraph tokenizer.

Analysis of Node IDs

Table 8: Comparison of codebook usage rates (%).

Usage rate↑	Cora	CiteSeer	PubMed
VQGraph	1.3	0.8	18.1
$\mathbf{NID}_{\mathrm{GCN}}$	84.7	97.9	79.1
$\mathbf{NID}_{\mathrm{GCN}(M=1)}$	83.3	81.3	78.1

Table 9: Average GEDs of 1-hop subgraphs among nodes.

GEDs↓	Cora	CiteSeer	PubMed
Random	7.21	4.83	9.61
VQGraph	6.85	4.73	9.03
$\mathbf{NID}_{\mathrm{GCN}}$	6.15	3.89	6.22

Table 10: Accuracy vs Inference Time.

Metric	Computer		ogbn-products	
	Acc↑	Time↓	Acc↑	Time↓
GCN	93.78	119.6ms	82.33	12.8s
SAGE	93.59	95.7ms	83.27	11.9s
VQGraph	90.28	1.4ms	79.17	1.6ms
$\mathbf{NID}_{\mathrm{SAGE}}$	**93.32**	**0.5ms**	**81.83**	**0.7ms**

Inference Acceleration:

On the ogbn-products dataset (with millions of nodes), NID reduces inference time from 11.9 seconds to 0.7 milliseconds.

Analysis of Node IDs

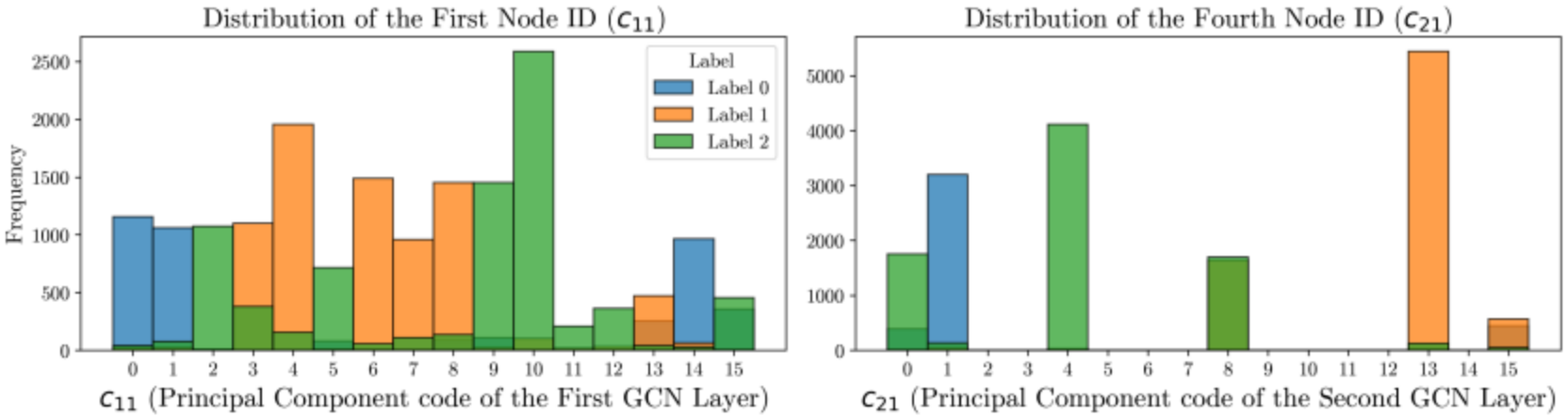


Figure 5: Codeword distributions of c_{11} and c_{21} in PubMed colored by the ground-truth labels.

Interpretability:

By analyzing the distribution of Node IDs' codeword indices, it was found that they can effectively distinguish between nodes of different categories, demonstrating strong interpretability.

Conclusions

Node IDs without reconstruction task:

- No performance drop → seamless integration with existing GNNs
- Compact (6-15 dim int4) → ideal for large-scale
- Interpretable → meaningful and human-understandable node features

https://github.com/LUOyk1999/NodeID

Conclusions

Node IDs without reconstruction task:

- No performance drop → seamless integration with existing GNNs
- Compact (6-15 dim int4) → ideal for large-scale
- Interpretable → meaningful and human-understandable node features

Thanks for listening!

https://github.com/LUOyk1999/NodeID