# **View Space:** Learning Representation across Arbitrary Graphs

**Dooho Lee** [1] **Myeong Kong** [1] **Minho Jeong** [2] **Jaemin Yoo** [2]

[1]School of Electrical Engineering, KAIST, Daejeon, Republic of Korea
[2]Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea

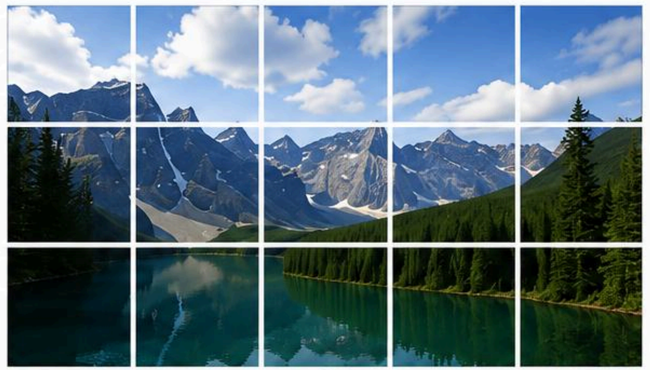


# Text vs. Image vs. Graph

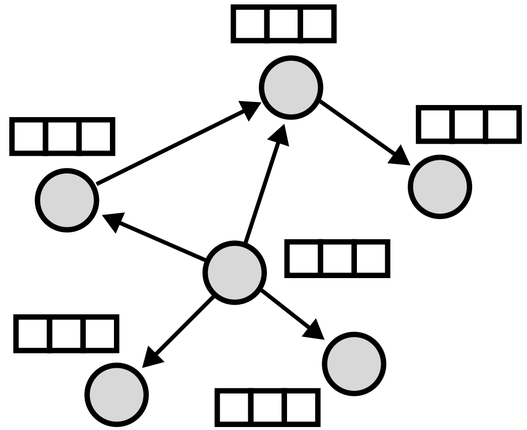
## Text

Large Language Models (LLMs), such as GPT-3 and GPT-4, utilize a process called tokenization. Tokenization involves breaking down text into smaller units, known as tokens, which the model can process and understand. These tokens can range from individual characters to entire words or even larger chunks, depending on the model. For GPT-3 and GPT-4, a Byte Pair Encoding (BPE) tokenizer is used. BPE is a subword tokenization technique that allows the model to dynamically build a vocabulary during training, efficiently representing common words and word fragments. Although the core tokenization process remains similar across different versions of these models, the specific implementation can vary based on the model's architecture and training objectives

## Image

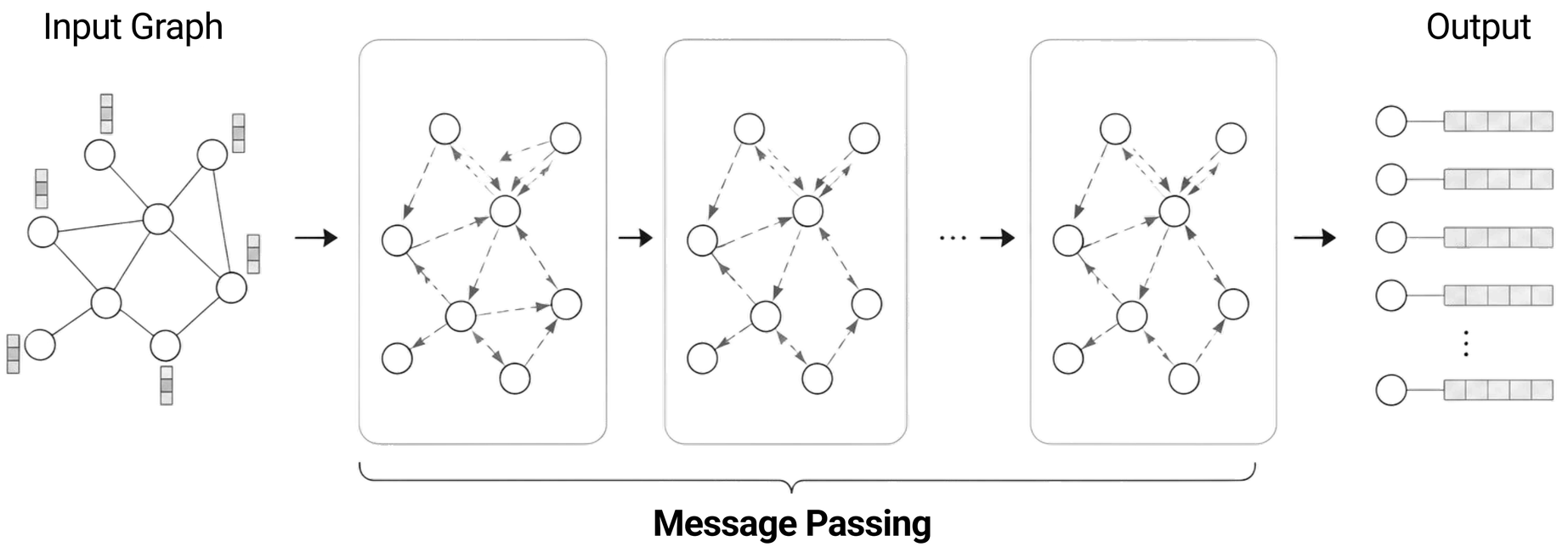


## Graph



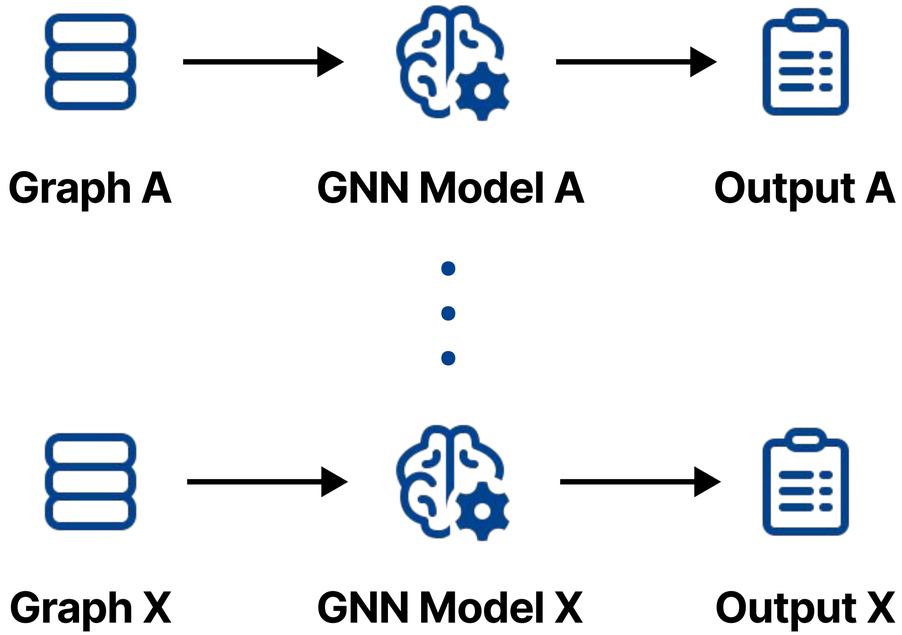
Graphs resist standardization: they vary in the number of nodes, graph topology, and feature specifications.

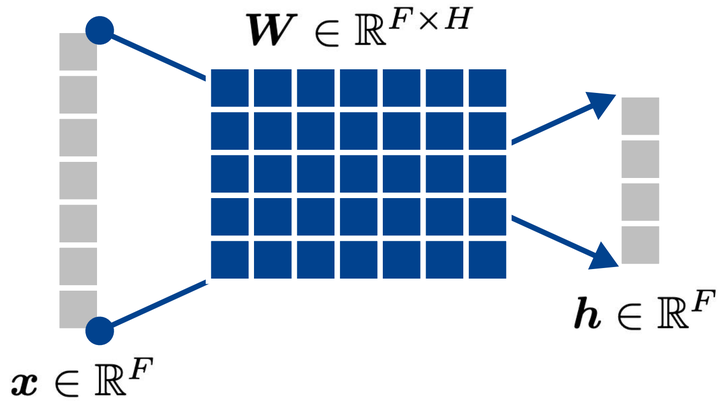
# Graph Neural Networks




Graph Neural Networks process graphs with arbitrary numbers of nodes and structures through message passing.

# Per-Dataset Nature of Graph Neural Networks




Feature transformation layers in GNNs are tied to the input feature space, necessitating retraining for each dataset.

# Fully Inductive Node Representation Learning (FINRL)

A fully inductive representation function $\Psi$ maps ***any unseen graph into latent node representations*** that jointly captures node features and graph structure.

$$\Psi : \mathbb{R}^{N \times F} \times \{0,1\}^{N \times N} \to \mathbb{R}^{N \times H} \quad \text{for all } N, F \geq 1$$

We introduce two requirements that guarantee a representation function is fully inductive (Lemma 3.1).

**R1.** The function $\Psi$ must be **equivariant to node permutations** to ensure it respects graph isomorphisms. For any permutation matrix $\boldsymbol{P} \in \{0,1\}^{N \times N}$:

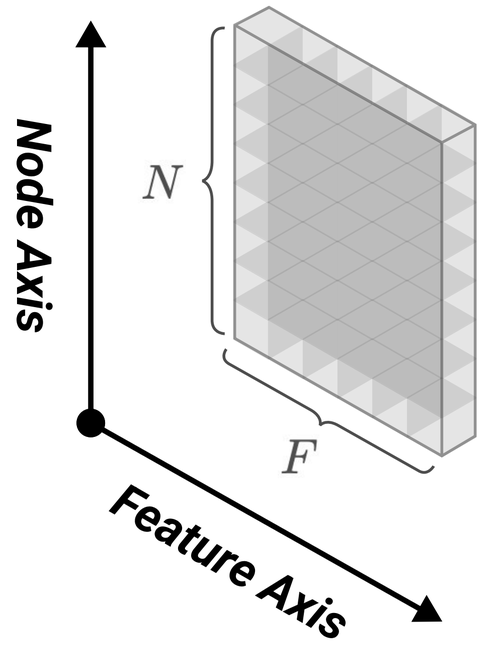
$$\Psi(\boldsymbol{P}\boldsymbol{X}, \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^\top) = \boldsymbol{P}\,\Psi(\boldsymbol{X}, \boldsymbol{A}).$$

**R2.** To support heterogenous feature sets across graphs, $\Psi$ must also be **equivariant to feature permutations**. For any permutation matrix $\boldsymbol{Q} \in \{0,1\}^{F \times F}$:

$$\Psi(\boldsymbol{X}\boldsymbol{Q}, \boldsymbol{A}) = \Psi(\boldsymbol{X}, \boldsymbol{A})\boldsymbol{Q}.$$

We formulate FINRL through two equivariance conditions that guarantee the full inductiveness of the function.

# Uncovering the Hidden Space of Graphs (1)




- Transformations along the node axis break R1
- Transformations along the feature axis break R2.

**How can we satisfy both simultaneously?**

**Our solution.**
Introduce a third axis of graph representation: the view space.

# Uncovering the Hidden Space of Graphs (2)

**Definition 4.1** (View Finder)**.** Given an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, a view finder is a matrix-valued function $\nu : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ that is node-permutation equivariant:

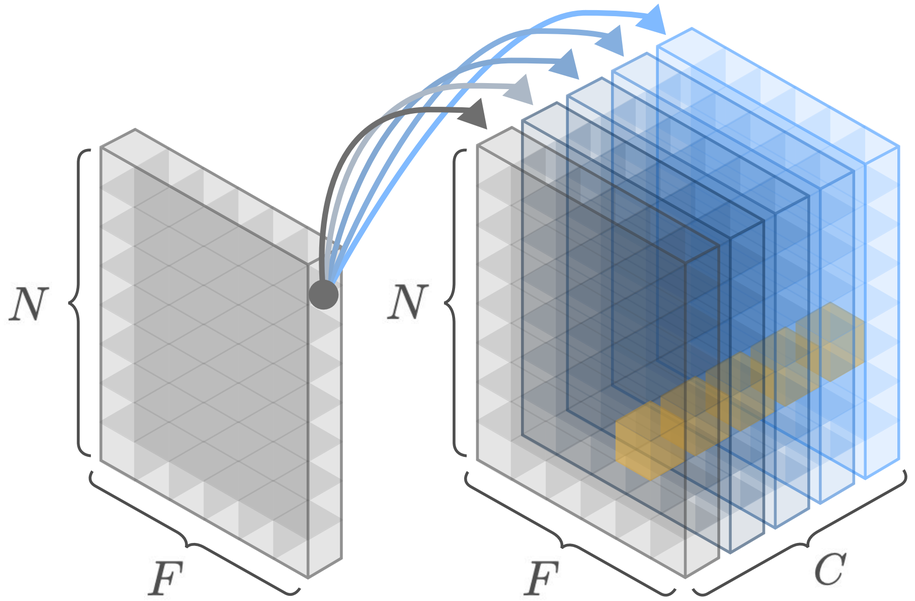
$$\nu(P\boldsymbol{A}P^\top) = P\,\nu(\boldsymbol{A})\,P^\top, \quad \forall P \in \{0,1\}^{N\times N}.$$

Applying $\nu(\boldsymbol{A})$ to $\boldsymbol{X} \in \mathbb{R}^{N\times F}$ produces a propagated node-feature matrix $\nu(\boldsymbol{A})\boldsymbol{X} \in \mathbb{R}^{N\times F}$.

**Definition 4.3** (View Stacking)**.** Given an adjacency matrix $\boldsymbol{A}$, a node feature matrix $\boldsymbol{X}$, and view finders $\{\nu_c\}_{c=1}^{C}$, the view stacking operator $\mathcal{V}$ is defined as

$$\mathbf{X} := \mathcal{V}(\boldsymbol{X}, \boldsymbol{A}) = [\,\nu_c(\boldsymbol{A})\boldsymbol{X}\,]_{c=1}^{C} \in \mathbb{R}^{N\times F\times C},$$

where $[\cdot]_{c=1}^{C}$ denotes the operation that stacks the $C$ propagated node-feature matrices along a new third dimension, forming the node-feature-view tensor $\mathbf{X}$.



# Uncovering the Hidden Space of Graphs (3)

**Definition 4.4** (View Vector)**.** For each node-feature pair $(n, f)$ in a graph, the view vector $\boldsymbol{v}_{n,f} \in \mathbb{R}^C$ is the vector obtained by slicing the node-feature-view tensor $\mathbf{X}$ along its third dimension, i.e., $\boldsymbol{v}_{n,f} = \mathbf{X}_{n,f,:}$.

**Definition 4.5** (View Space)**.** The view space is the vector space $\mathbb{R}^C$ in which the set of all view vectors $\{\boldsymbol{v}_{n,f}\}$ resides. Each coordinate axis corresponds to a specific view finder.

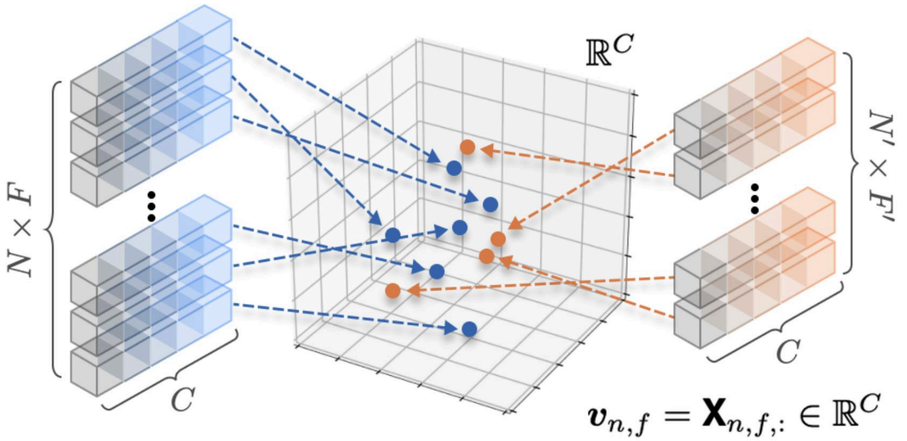



*Figure 1.* Graphs of varying sizes and features can be mapped into the view space as $N \times F$ view vectors $\boldsymbol{v}_{n,f} = \mathbf{X}_{n,f,:} \in \mathbb{R}^C$.

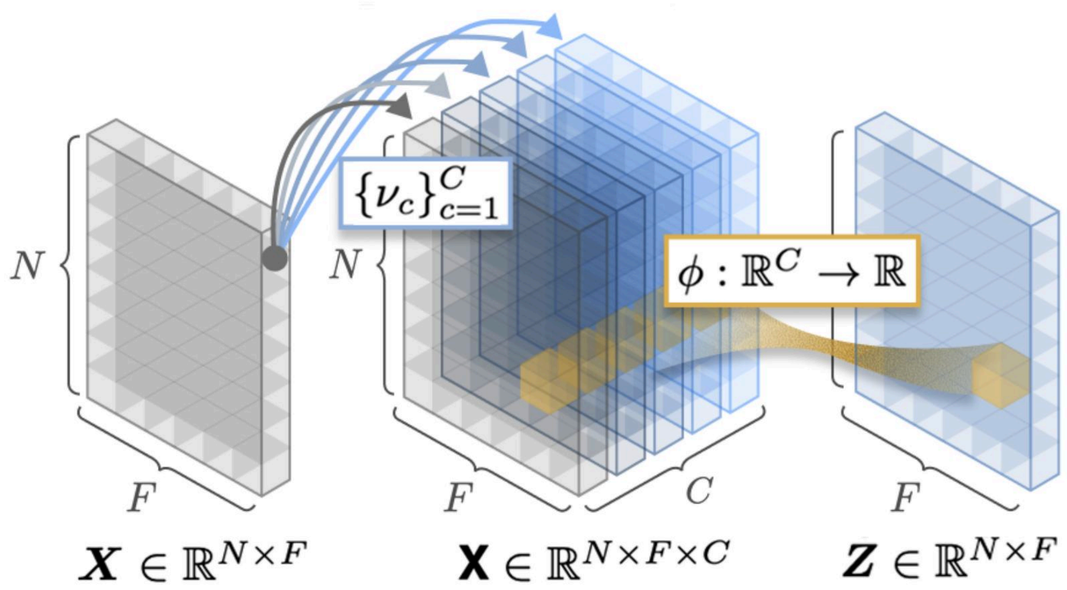
The third axis defines a shared view space,
in which each node-feature pair from an arbitrary dataset is represented as a view vector.

# Learning Representations via View Space

**Definition 4.6** (Graph View Transformation)**.** Given an adjacency matrix $\boldsymbol{A}$ and node features $\boldsymbol{X}$, the *Graph View Transformation* $\Psi$ maps to node representations $\boldsymbol{Z}$ as

$$\Psi(\boldsymbol{X}, \boldsymbol{A}) \;=\; \big[\, \phi(\mathbf{X}_{n,f,:} \mid \theta) \,\big]_{n\in[N],\, f\in[F]} \in \mathbb{R}^{N\times F},$$

where $\mathbf{X} = \mathcal{V}(\boldsymbol{X}, \boldsymbol{A})$ is the node-feature-view tensor obtained via view stacking (Definition 4.3), and $\phi : \mathbb{R}^C \to \mathbb{R}$ is a learnable dimension-collapsing function with parameters $\theta$, shared across all node-feature entries $(n, f)$.




GVT satisfies both R1 and R2 simultaneously; for simplicity, we implement it with an MLP in this work.

# RGVT: A Fully Inductive Graph Encoder

By decoupling parameterization from depth, as in RNNs, the model can operate at arbitrary depth $L$ , enabling dataset-specific depth selection without retraining.

$$\boldsymbol{Z}_k = \Psi(\boldsymbol{X}_k, \boldsymbol{A}_k \mid \theta)^{L_k}.$$

**Training of RGVT.** A shared RGVT encoder produces transferable representations for arbitrary graphs, while lightweight predictors specialize them to each dataset. Appropriate recurrent depth $L_k$ enables adaptation to label-scarce and unseen graphs.

$$\arg\min_{\hat{\theta}} \mathcal{L}\big(f_k(\boldsymbol{Z}_k), \boldsymbol{Y}_k\big), \quad f_k : \mathbb{R}^{N_k \times F_k} \to \mathbb{R}^{N_k \times D_k}.$$

We propose applying GVT recurrently, enabling the model to choose the appropriate depth on a per-dataset basis.

# Experiment Settings

**Pre-training & Transfer.** RGVT is pretrained on OGBN-Arxiv (Hu et al., 2020a) using its public split, and downstream evaluation is conducted on 27 node classification benchmarks.

**Splits.** Following prior work (Zhao et al., 2025), we use public splits when available; otherwise, we sample 20 nodes per class for training, split the remainder evenly for validation and testing.

**View Finder.** For RGVT, we use $\{\boldsymbol{I}\} \cup \{(\boldsymbol{D}^{-1}\boldsymbol{A})^k, (\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})^k\}_{k=1}^{K}$ as the set of view finders, where is $\boldsymbol{D}$ degree matrix, and consider $K \in \{1, 2, 3\}$ as a hyperparameter.

**RGVT Details.** Linear/MLP predictors are trained on top of RGVT representations to map them to the target label space, while the recurrent depth is selected for per-dataset based on validation.

RGVT requires training only L lightweight predictors for depth selection,
making it substantially more efficient than conventional per-dataset GNN training and hyperparameter tuning.

# Validation of FI-NRL

*Table 1.* Performance (%) comparison against predictor architectures and GraphAny across datasets grouped by feature type. The numbers in parentheses represent the number of datasets in each category; for example, we include 5 graphs with signed dense features. **1st** and 2nd best results are highlighted. RGVT consistently and significantly outperforms all baselines.

| **Method** | OGBN-Arxiv (1) | Signed Dense (5) | Unsigned Dense (4) | Sparse (4) | Binary Dense (3) | Binary Sparse (8) | One-hot (4) | **Total Avg. (28)** |
|---|---|---|---|---|---|---|---|---|
| Linear | $52.44_{\pm 0.04}$ | $53.29_{\pm 0.08}$ | $75.67_{\pm 0.64}$ | $66.41_{\pm 0.57}$ | $72.18_{\pm 0.69}$ | $57.11_{\pm 0.77}$ | $38.86_{\pm 2.52}$ | $59.41_{\pm 0.84}$ |
| MLP | $53.80_{\pm 0.14}$ | $55.08_{\pm 0.31}$ | $75.86_{\pm 0.68}$ | $69.02_{\pm 0.77}$ | $72.88_{\pm 0.64}$ | $57.65_{\pm 1.84}$ | $39.34_{\pm 2.43}$ | $60.43_{\pm 1.20}$ |
| GraphAny (Wisconsin) | $57.77_{\pm 0.45}$ | $59.12_{\pm 0.46}$ | $71.78_{\pm 1.22}$ | $81.61_{\pm 0.20}$ | $83.44_{\pm 0.24}$ | $55.25_{\pm 2.62}$ | $52.68_{\pm 0.40}$ | $64.72_{\pm 0.96}$ |
| GraphAny (Cora) | $58.58_{\pm 0.10}$ | $59.38_{\pm 0.42}$ | $71.76_{\pm 1.60}$ | $81.49_{\pm 0.10}$ | $83.35_{\pm 0.09}$ | $53.40_{\pm 2.44}$ | $53.30_{\pm 0.93}$ | $64.30_{\pm 0.94}$ |
| GraphAny (Arxiv) | $58.63_{\pm 0.14}$ | $59.70_{\pm 0.36}$ | $72.62_{\pm 1.32}$ | $81.68_{\pm 0.23}$ | $83.56_{\pm 0.08}$ | $54.18_{\pm 2.49}$ | $53.02_{\pm 0.35}$ | $64.71_{\pm 0.89}$ |
| GraphAny (Best) | $58.63_{\pm 0.14}$ | $59.74_{\pm 0.32}$ | $72.64_{\pm 1.02}$ | $81.89_{\pm 0.12}$ | $83.92_{\pm 0.13}$ | $55.58_{\pm 2.47}$ | $53.81_{\pm 0.77}$ | $65.30_{\pm 0.93}$ |
| **RGVT**+ Linear (Arxiv) | $\underline{70.14}_{\pm 0.28}$ | $\underline{64.95}_{\pm 0.44}$ | $\underline{76.44}_{\pm 0.39}$ | $\mathbf{84.33}_{\pm \mathbf{0.45}}$ | $\mathbf{85.11}_{\pm \mathbf{0.54}}$ | $\underline{62.77}_{\pm 1.93}$ | $\underline{58.85}_{\pm 3.14}$ | $\underline{70.03}_{\pm 1.26}$ |
| **RGVT**+ MLP (Arxiv) | $\mathbf{71.11}_{\pm \mathbf{0.28}}$ | $\mathbf{66.37}_{\pm \mathbf{0.90}}$ | $\mathbf{77.12}_{\pm \mathbf{0.45}}$ | $\underline{83.98}_{\pm 0.81}$ | $\underline{84.86}_{\pm 0.51}$ | $\mathbf{63.87}_{\pm \mathbf{1.58}}$ | $\mathbf{62.48}_{\pm \mathbf{3.95}}$ | $\mathbf{71.13}_{\pm 1.41}$ |

RGVT effectively encodes graph structure, producing representations that are both more predictive and robust than those of GraphAny.

# Power of FI-NRL

*Table 2.* Performance (%) comparison against 12 dataset-specialized GNNs across datasets grouped by feature type. 1st, 2nd, 3rd best results are highlighted. RGVT achieves the best average performance with the MLP predictor and ranks 2nd with the linear predictor.

| **Method** | OGBN-Arxiv (1) | Signed Dense (5) | Unsigned Dense (4) | Sparse (4) | Binary Dense (3) | Binary Sparse (8) | One-hot (4) | **Total Avg. (28)** |
|---|---|---|---|---|---|---|---|---|
| GIN | $65.77_{\pm 1.08}$ | $55.57_{\pm 3.94}$ | $67.95_{\pm 3.90}$ | $71.93_{\pm 4.66}$ | $44.83_{\pm 4.72}$ | $43.30_{\pm 2.49}$ | $55.31_{\pm 3.88}$ | $54.98_{\pm 3.70}$ |
| GAT | $71.89_{\pm 0.24}$ | $60.89_{\pm 0.31}$ | $73.24_{\pm 0.62}$ | $82.50_{\pm 1.26}$ | $83.36_{\pm 0.90}$ | $49.20_{\pm 2.58}$ | $54.39_{\pm 4.98}$ | $63.88_{\pm 1.87}$ |
| GATv2 | $72.13_{\pm 0.11}$ | $64.45_{\pm 0.45}$ | $72.00_{\pm 1.22}$ | $81.68_{\pm 1.44}$ | $83.20_{\pm 0.80}$ | $49.61_{\pm 2.80}$ | $56.12_{\pm 3.41}$ | $64.57_{\pm 1.83}$ |
| $S^2GC$ | $69.17_{\pm 0.02}$ | $59.94_{\pm 0.08}$ | $75.28_{\pm 0.35}$ | $82.04_{\pm 0.29}$ | $84.48_{\pm 0.14}$ | $52.28_{\pm 1.15}$ | $56.97_{\pm 1.35}$ | $65.31_{\pm 0.64}$ |
| SGC | $68.69_{\pm 0.03}$ | $58.95_{\pm 0.05}$ | $75.18_{\pm 0.26}$ | $82.36_{\pm 0.37}$ | $84.73_{\pm 0.10}$ | $51.15_{\pm 0.85}$ | $62.54_{\pm 3.24}$ | $65.66_{\pm 0.82}$ |
| JKNet | $71.73_{\pm 0.24}$ | $63.18_{\pm 1.19}$ | $76.12_{\pm 0.42}$ | $83.30_{\pm 0.71}$ | $84.02_{\pm 0.49}$ | $51.72_{\pm 1.60}$ | $58.46_{\pm 4.93}$ | $66.19_{\pm 1.59}$ |
| APPNP | $70.85_{\pm 0.15}$ | $64.86_{\pm 0.21}$ | $76.27_{\pm 0.36}$ | $83.66_{\pm 0.38}$ | $84.11_{\pm 0.21}$ | $55.21_{\pm 1.47}$ | $49.29_{\pm 1.26}$ | $66.26_{\pm 0.77}$ |
| GCN | $71.19_{\pm 0.30}$ | $61.49_{\pm 0.27}$ | $76.00_{\pm 0.38}$ | $83.48_{\pm 0.43}$ | $84.43_{\pm 0.26}$ | $53.03_{\pm 1.49}$ | $59.51_{\pm 1.43}$ | $66.46_{\pm 0.82}$ |
| GCNII | $72.05_{\pm 0.08}$ | $67.11_{\pm 0.25}$ | $77.54_{\pm 0.45}$ | $81.00_{\pm 0.97}$ | $83.65_{\pm 1.15}$ | $54.90_{\pm 1.73}$ | $56.11_{\pm 4.25}$ | $67.30_{\pm 1.47}$ |
| SAGE | $71.22_{\pm 0.24}$ | $67.98_{\pm 0.34}$ | $76.63_{\pm 0.65}$ | $82.14_{\pm 0.90}$ | $83.69_{\pm 0.51}$ | $60.07_{\pm 2.57}$ | $51.84_{\pm 3.39}$ | $68.36_{\pm 1.55}$ |
| GPRGNN | $69.45_{\pm 0.41}$ | $67.24_{\pm 0.28}$ | $76.78_{\pm 0.18}$ | $84.35_{\pm 0.40}$ | $85.26_{\pm 0.31}$ | $60.61_{\pm 1.89}$ | $50.40_{\pm 1.65}$ | $68.68_{\pm 0.94}$ |
| UniMP | $71.78_{\pm 0.12}$ | $69.09_{\pm 0.34}$ | $76.97_{\pm 0.60}$ | $82.28_{\pm 0.62}$ | $84.22_{\pm 0.54}$ | $59.15_{\pm 3.46}$ | $54.94_{\pm 3.61}$ | $68.86_{\pm 1.80}$ |
| **RGVT** + Linear | $70.14_{\pm 0.28}$ | $64.95_{\pm 0.44}$ | $76.44_{\pm 0.39}$ | $84.33_{\pm 0.45}$ | $85.11_{\pm 0.54}$ | $62.77_{\pm 1.93}$ | $58.85_{\pm 3.14}$ | $70.03_{\pm 1.26}$ |
| **RGVT** + MLP | $71.11_{\pm 0.28}$ | $66.37_{\pm 0.90}$ | $77.12_{\pm 0.45}$ | $83.98_{\pm 0.81}$ | $84.86_{\pm 0.51}$ | $63.87_{\pm 1.58}$ | $62.48_{\pm 3.95}$ | $71.13_{\pm 1.41}$ |

RGVT even surpasses 12 per-dataset trained and tuned GNNs,
suggesting that GVT can serve as a compelling alternative to conventional feature transformation.

# Conclusion

1. Introduces the **view space**, a graph-centric representation that enables arbitrary graphs with heterogeneous feature spaces to be processed in a unified manner.
2. Proposes **Graph View Transformation (GVT)**, a parametric mapping in the view space that enables node–feature dynamic aggregation beyond conventional GNNs.
3. Demonstrates strong cross-graph transfer with Recurrent GVT (RGVT), **outperforming 12 per-dataset optimized GNNs** despite using a single transferable model.

*Full Paper*

*Project Page*