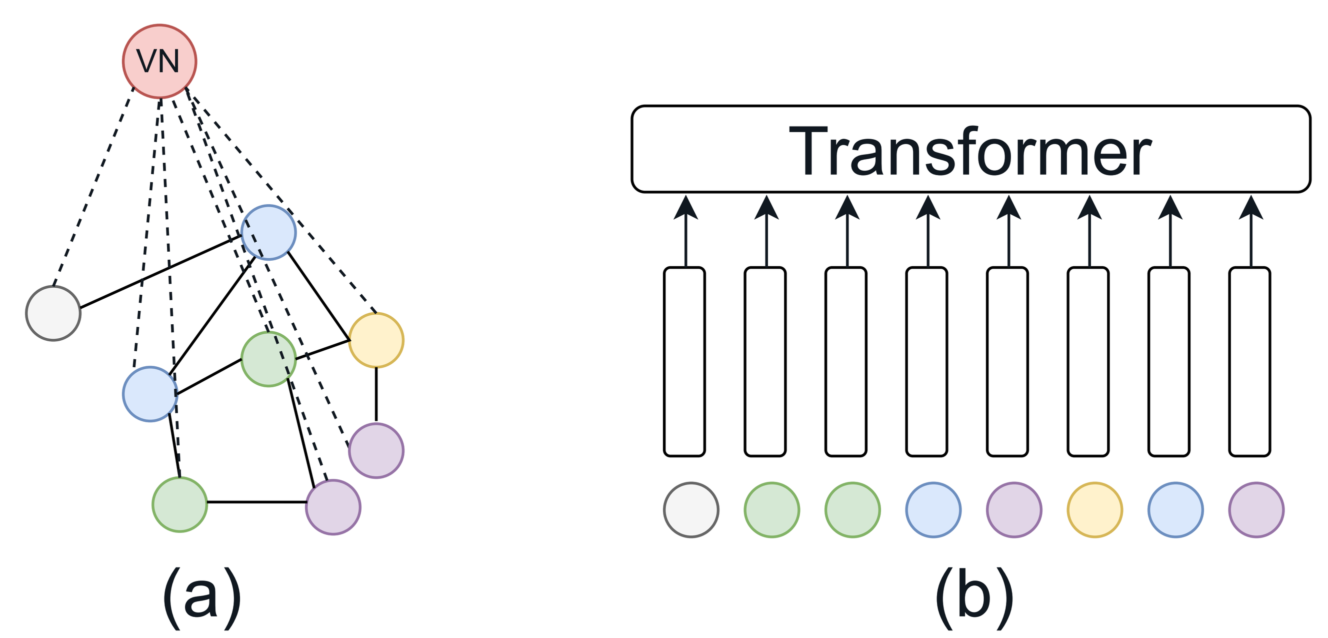
On the Connection Between MPNN and Graph Transformer

Chen Cai[1], Truong Son Hy[1], Rose Yu[1], and Yusu Wang[1]

University of California San Diego [1] Emails: {c1cai,tshy,roseyu,yusuwang}@ucsd.edu

Motivation & Main Results

- Message-passing neural networks (MPNN) have been the leading architecture for processing graph-structured data.
- Graph Transformer (GT) recently emerges as a new paradigm of graph learning algorithms.
- **GT → MPNN.** With proper position embedding, GT can approximate MPNN arbitrarily well [2]
- **MPNN → GT.** What about the other direction?
- We systematically study the representation power and limitation of MPNN + VN (virtual node), a widely used heuristics with little theoretical understanding.



(a) MPNN + VN = we augment the graph with a virtual node (VN) connecting to all other nodes. **(b)** Graph Transformer = we treat each node embedding as a token and apply a Transformer on the sequence of node embeddings/tokens.

Depth	Width	Self-Attention	Note
$\mathcal{O}(1)$	$\mathcal{O}(n^d)$	Full	Leverage the universality of equivariant DeepSets [4]
$\mathcal{O}(1)$	$\mathcal{O}(1)$	Approximate	Approximate self attention in Performer [1]
$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, strong assumption on $\mathcal{X}$
$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, relaxed assumption on $\mathcal{X}$

Summary of approximation result of MPNN + VN on self-attention layer. n is the number of nodes and d is the feature dimension of node features.

MPNN + VN with $\mathcal{O}(1)$ depth and $\mathcal{O}(1)$ width can approximate Performer

Rewrite self-attention in kernel form

$$\boldsymbol{x}_i^{(l+1)} = \sum_{j=1}^{n} \frac{\kappa\left(\boldsymbol{W}_Q^{(l)}\boldsymbol{x}_i^{(l)}, \boldsymbol{W}_K^{(l)}\boldsymbol{x}_j^{(l)}\right)}{\sum_{k=1}^{n}\kappa\left(\boldsymbol{W}_Q^{(l)}\boldsymbol{x}_i^{(l)}, \boldsymbol{W}_K^{(l)}\boldsymbol{x}_k^{(l)}\right)} \cdot \left(\boldsymbol{W}_V^{(l)}\boldsymbol{x}_j^{(l)}\right) \tag{1}$$

approximate kernel $\kappa(\boldsymbol{x},\boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})\rangle_{\mathcal{V}} \approx \phi(\boldsymbol{x})^T\phi(\boldsymbol{y})$

$$\boldsymbol{x}_i^{(l+1)} = \sum_{j=1}^{n} \frac{\phi(\boldsymbol{q}_i)^T\phi(\boldsymbol{k}_j)}{\sum_{k=1}^{n}\phi(\boldsymbol{q}_i)^T\phi(\boldsymbol{k}_k)} \cdot \boldsymbol{v}_j = \frac{\left(\phi(\boldsymbol{q}_i)^T\sum_{j=1}^{n}\phi(\boldsymbol{k}_j)\otimes\boldsymbol{v}_j\right)^T}{\phi(\boldsymbol{q}_i)^T\sum_{k=1}^{n}\phi(\boldsymbol{k}_k)}. \tag{2}$$

which can be approximated by MPNN+VN with constant depth and width!

- Of course Performer is just one of the efficient transformers. There are many other linear transformers that can not be expressed under MPNN+VN framework, such as Linformer and Sparse Transformer.
- Efficient transformer literature explores a larger model design space than MPNN+VN.

Wide MPNN + VN ($\mathcal{O}(1)$ depth, $\mathcal{O}(n^d)$ width)

Theorem 1. *MPNN + VN can simulate (not just approximate) equivariant DeepSets: $\mathbb{R}^{n\times d}\to\mathbb{R}^{n\times d}$. This implies that MPNN + VN of $\mathcal{O}(1)$ depth and $\mathcal{O}(n^d)$ width is permutation equivariant universal, and can approximate self-attention layer and transformers arbitrarily well.*

Main idea: show MPNN + VN can simulate DeepSets + leverage the universality of DeepSets to approximate permutation equivariant maps.

Deep MPNN + VN ($\mathcal{O}(n)$ depth, $\mathcal{O}(1)$ width)

Definition 1. *Self attention layer $\boldsymbol{L}:\mathbb{R}^{n\times d}\to\mathbb{R}^{n\times d}$ is of the following form: $\boldsymbol{L}(\boldsymbol{X}) = softmax(\boldsymbol{X}\boldsymbol{W}_Q(\boldsymbol{X}\boldsymbol{W}_K)^T)\boldsymbol{X}\boldsymbol{W}_V$.*

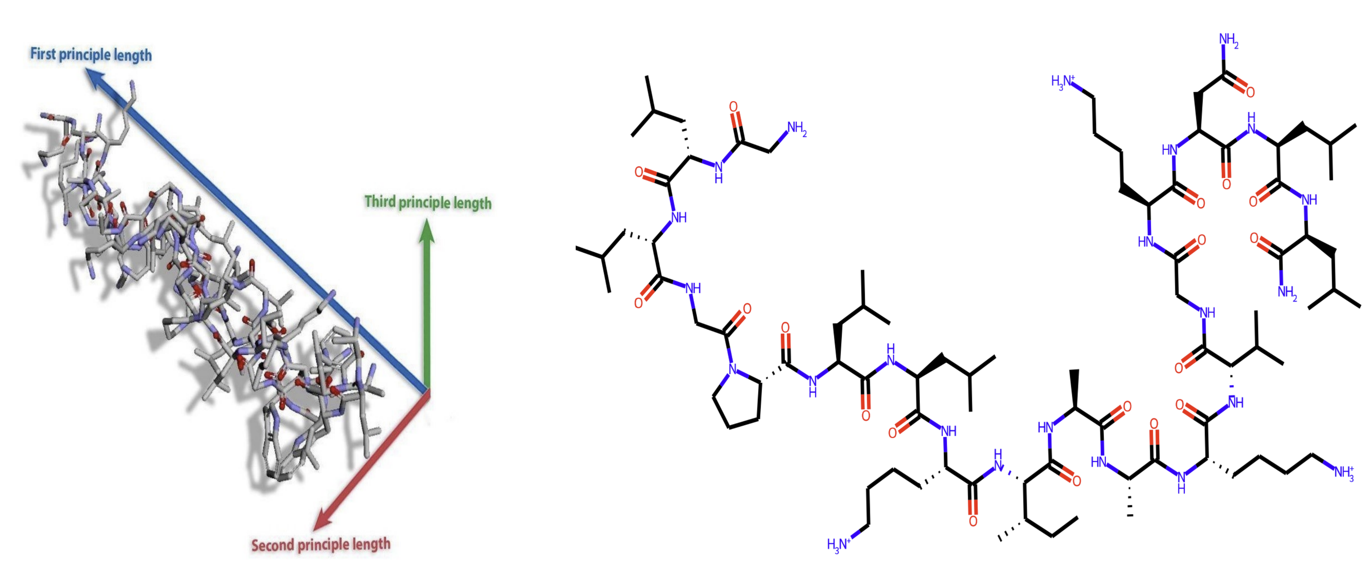
- AS1. $\mathcal{X}$ is $(\boldsymbol{V},\delta)$ separable by α for some fixed $\boldsymbol{V}\in\mathbb{R}^{n\times d}$ and $\delta > 0$.
- AS2. $\forall i\in[n], \boldsymbol{x}_i\in\mathcal{X}_i, \|\boldsymbol{x}_i\| < C_1$. This implies $\mathcal{X}$ is compact.
- AS3. $\|\boldsymbol{W}_Q\| < C_2, \|\boldsymbol{W}_K\| < C_2, \|\boldsymbol{W}_V\| < C_2$ for target layer $\boldsymbol{L}$.

Theorem 2. *Assume AS 1-3 hold for the compact set $\mathcal{X}$ and $\boldsymbol{L}$. Given any graph G of size n with node features $\boldsymbol{X}\in\mathcal{X}$, and a self-attention layer $\boldsymbol{L}$ on G (fix $\boldsymbol{W}_K, \boldsymbol{W}_Q, \boldsymbol{W}_V$), there exists a $\mathcal{O}(n)$ layer of heterogeneous MPNN + VN with the specific aggregate/update/message function that can approximate $\boldsymbol{L}$ on $\mathcal{X}$ arbitrarily well.*

Main idea: use VN to select one node to process at each iteration. After $\mathcal{O}(n)$ rounds, we are able to approximate one self-attention layer.

MPNN + VN for Long Range Graph Benchmark (LRGB)

- Peptides-functional and Peptides-structural are two datasets of LRGB
- Previously GT shows a large margin over MPNN
- Simply adding VN is enough to make MPNN outperform GT



Model	# Params.	Peptides-functional		Peptides-structural	
		Test AP before VN	Test AP after VN ↑	Test MAE before VN	Test MAE after VN ↓
GCN	508k	0.5930±0.0023	0.6623±0.0038	0.3496±0.0013	**0.2488±0.0021**
GINE	476k	0.5498±0.0079	0.6346±0.0071	0.3547±0.0045	0.2584±0.0011
GatedGCN	509k	0.5864±0.0077	0.6635±0.0024	0.3420±0.0013	0.2523±0.0016
GatedGCN+RWSE	506k	0.6069±0.0035	**0.6685±0.0062**	0.3357±0.0006	0.2529±0.0009
Transformer+LapPE	488k	0.6326±0.0126	-	0.2529±0.0016	-
SAN+LapPE	493k	0.6384±0.0121	-	0.2683±0.0043	-
SAN+RWSE	500k	0.6439±0.0075	-	0.2545±0.0012	-

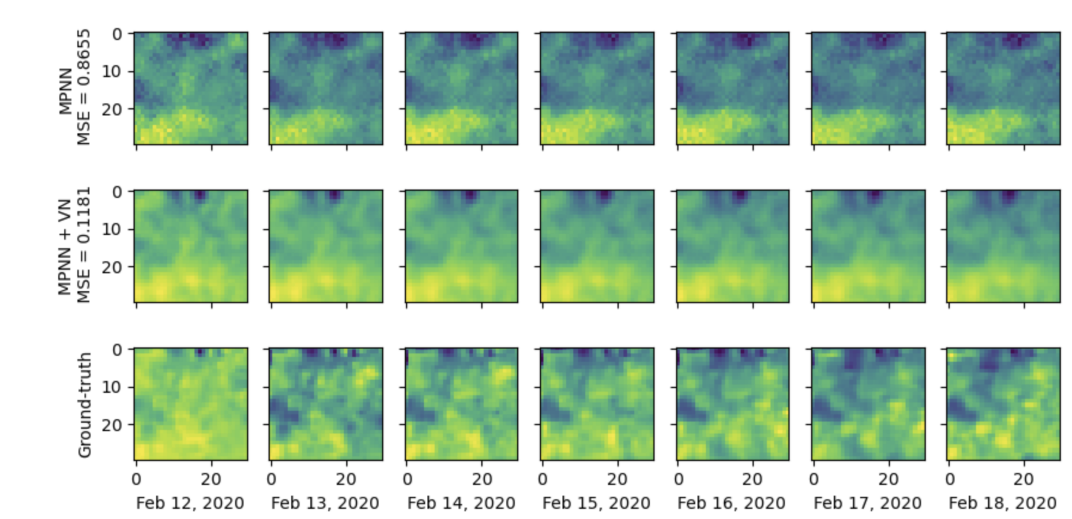
VN as a Global Module

- Replace Global Module (transformer) in GraphGPS [3] with VN module
- Comparable results with GraphGPS and much better than existing MPNN + VN

Model	ogbg-molhiv	ogbg-molpcba	ogbg-ppa	ogbg-code2
	AUROC ↑	Avg. Precision ↑	Accuracy ↑	F1 score ↑
GCN	0.7606 ± 0.0097	0.2020 ± 0.0024	0.6839 ± 0.0084	0.1507 ± 0.0018
GCN+virtual node	0.7599 ± 0.0119	0.2424 ± 0.0034	0.6857 ± 0.0061	0.1595 ± 0.0018
GIN	0.7558 ± 0.0140	0.2266 ± 0.0028	0.6892 ± 0.0100	0.1495 ± 0.0023
GIN+virtual node	0.7707 ± 0.0149	0.2703 ± 0.0023	0.7037 ± 0.0107	0.1581 ± 0.0026
SAN	0.7785 ± 0.2470	0.2765 ± 0.0042	–	–
GraphTrans (GCN-Virtual)	–	0.2761 ± 0.0029	–	0.1830 ± 0.0024
K-Subtree SAT	–	–	0.7522 ± 0.0056	0.1937 ± 0.0028
	0.7880 ± 0.0101	0.2907 ± 0.0028	0.8015 ± 0.0033	0.1894 ± 0.0024
MPNN + VN (ours)	0.7687 ± 0.0136	0.2848 ± 0.0026	0.8055 ± 0.0038	0.1727 ± 0.0017

MPNN + VN for climate prediction

We apply our MPNN + VN model to forecast daily **sea surface temperature** (SST) in the Pacific Ocean from 1982 to 2021, given 6 weeks of history to predict the next 1, 2 and 4 weeks of temperatures. The input is a grid graph of 30 longitudes and 30 latitudes at 0.5°-degree resolution. We report the error with Mean Square Error (MSE) metric.



Model	4 weeks	2 weeks	1 week
MLP	0.3302	0.2710	0.2121
TF-Net	0.2833	**0.2036**	**0.1462**
Linear Transformer + LapPE	0.2818	0.2191	0.1610
MPNN	0.2917	0.2281	0.1613
MPNN + VN	**0.2806**	0.2130	0.1540

Acknowledgements

This work was supported in part by the U.S. Department Of Energy, Office of Science, U. S. Army Research Office under Grant W911NF-20-1-0334, Google Faculty Award, Amazon Research Award, and NSF Grants #2134274, #2107256, #2134178, CCF-2217033, and CCF-2112665.

Reference

[1] Krzysztof et al., *Rethinking a en on with performers*, ICLR 2021.
[2] Kim et al., *Pure transformers are powerful graph learners*, NeurIPS 2022.
[3] Rampášek et al., *Recipe for a General, Powerful, Scalable Graph Transformer*, NeurIPS 2022.
[4] N. Segol and Y. Lipman, *On universal equivariant set networks*, ICLR 2020.