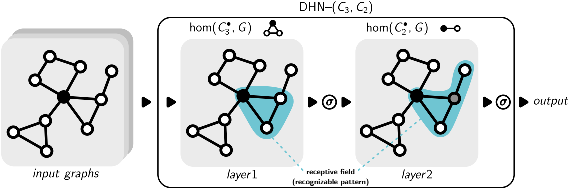
Deep Homomorphism Network

Expressivity through stacking homomorphism layers



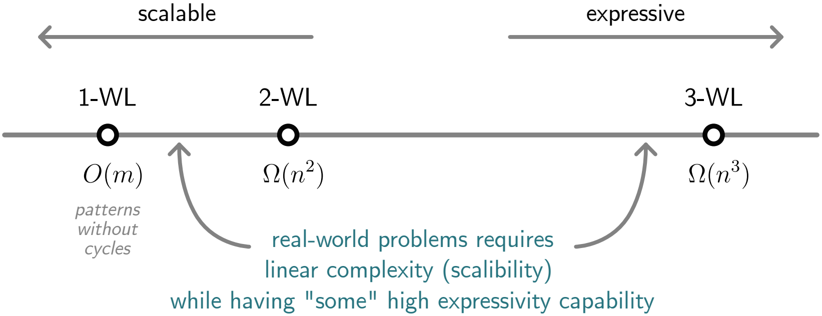
Takanori Maehara & Hoang NT

Roku & UTokyo

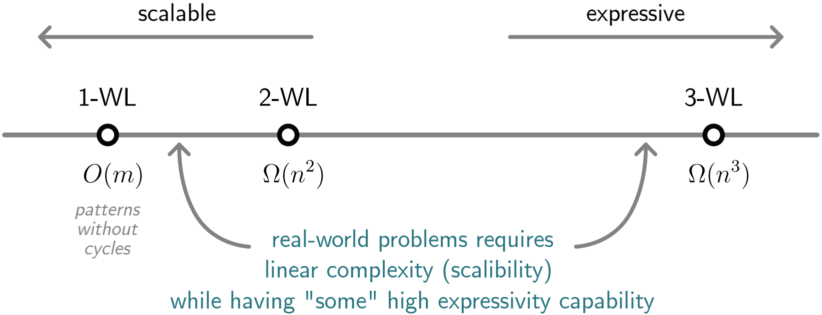
NeurIPS 2024

Introduction

Introduction

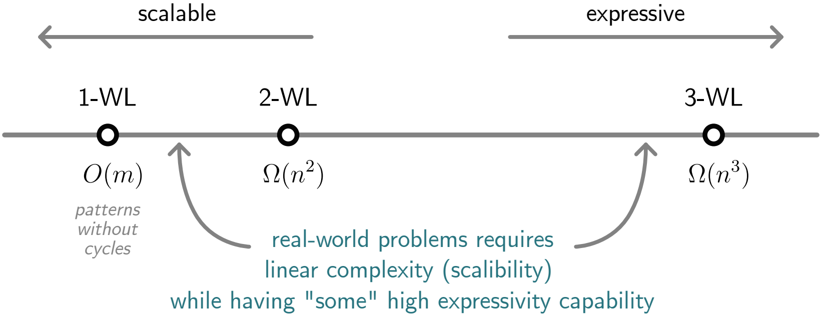


Introduction



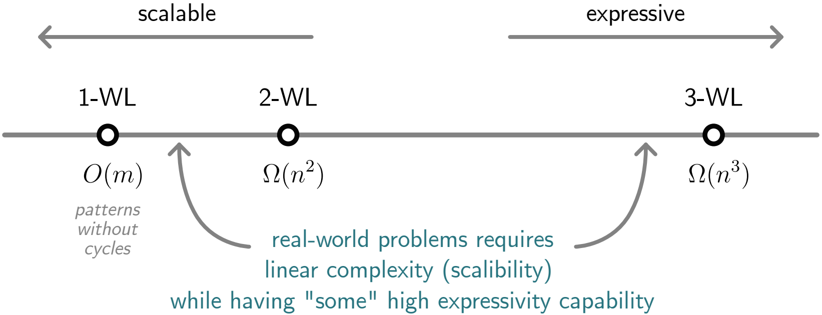
- Our work is motivated by the complexity-expressivity trade-offs:

Introduction



- Our work is motivated by the complexity-expressivity trade-offs:
  - Develop models that can detect patterns with cycles, but run in $O(n)$ time on sparse graphs.

Introduction



Our work is motivated by the complexity-expressivity trade-offs:

- Develop models that can detect patterns with cycles, but run in $O(n)$ time on sparse graphs.
- Analyze the expressivity of the multi-layers version of the proposed model.

Homomorphism Convolution Layer

Homomorphism Convolution Layer

- Basic definitions and ideas:

Homomorphism Convolution Layer

- Basic definitions and ideas:
  - A graph homomorphism from F to G is a mapping from $V(F)$ to $V(G)$ that preseves their structures.

Homomorphism Convolution Layer

- Basic definitions and ideas:
  - A graph homomorphism from F to G is a mapping from $V(F)$ to $V(G)$ that preseves their structures.
  - Graph homomorphism counts vectors from a collection of graphs F's to G and H can be used to distinguish G and H.

Homomorphism Convolution Layer

- Basic definitions and ideas:
  - A graph homomorphism from F to G is a mapping from $V(F)$ to $V(G)$ that preseves their structures.
  - Graph homomorphism counts vectors from a collection of graphs F's to G and H can be used to distinguish G and H.
- Our proposal:

Homomorphism Convolution Layer

- Basic definitions and ideas:
  - A graph homomorphism from F to G is a mapping from $V(F)$ to $V(G)$ that preseves their structures.
  - Graph homomorphism counts vectors from a collection of graphs F's to G and H can be used to distinguish G and H.
- Our proposal:
  - Specify F's collection of interest (cycles of lengths up to 6, cliques, etc.), then enumerate all these homorphism mappings.

Homomorphism Convolution Layer

- Basic definitions and ideas:
  - A graph homomorphism from F to G is a mapping from $V(F)$ to $V(G)$ that preseves their structures.
  - Graph homomorphism counts vectors from a collection of graphs F's to G and H can be used to distinguish G and H.
- Our proposal:
  - Specify F's collection of interest (cycles of lengths up to 6, cliques, etc.), then enumerate all these homorphism mappings.
  - Aggregate the transformed features along the mapping to get a single homomorphism convolution layer:

$$\hom((F^\bullet, \mu), (G^\bullet, x)) = \sum_{\pi \in \mathrm{Hom}(F^\bullet, G^\bullet)} \prod_{p \in V(F^\bullet)} \mu_p(x_{\pi(p)})$$

Expressivity of Deep Homomorphism Network (DHN)

Let P^ℓ be a series of patterns, and $(G, x), (G', x')$ be inputs.

Theorem 1 (Main Theorem)

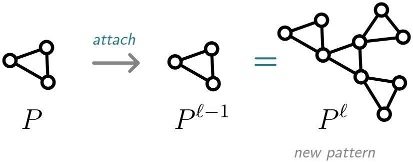
DHN(x) =*DHN*(x') *iff* $\hom(P^\ell, G) \neq \hom(P^\ell, G')$, *where* P^0 *is a singleton and* P^ℓ *are patterns obtained by attaching* P *to* $P^{\ell-1}$.

Expressivity of Deep Homomorphism Network (DHN)

Let P^ℓ be a series of patterns, and $(G, x), (G', x')$ be inputs.

Theorem 1 (Main Theorem)

DHN$(x) =$*DHN*(x') *iff* $\hom(P^\ell, G) \neq \hom(P^\ell, G')$, *where* P^0 *is a singleton and* P^ℓ *are patterns obtained by attaching* P *to* $P^{\ell-1}$.

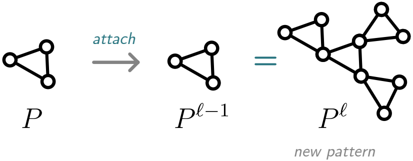


Expressivity of Deep Homomorphism Network (DHN)

Let P^ℓ be a series of patterns, and $(G, x), (G', x')$ be inputs.

Theorem 1 (Main Theorem)

DHN(x) =*DHN*(x') *iff* $\hom(P^\ell, G) \neq \hom(P^\ell, G')$, *where* P^0 *is a singleton and* P^ℓ *are patterns obtained by attaching* P *to* $P^{\ell-1}$.



- Stacking layers make the model *exponentially* expressive.
- DHN is a generalization of 1-WL when P is the single-edge.

Complexity, Implementation, and Experiments

Complexity, Implementation, and Experiments

Complexity: DHN runs in the same time complexity as computing $\hom(P, G)$, i.e., $O(n)$ if G is sparse and P is tree-like. This is true for real-world applications.

Complexity, Implementation, and Experiments

- Complexity: DHN runs in the same time complexity as computing $\hom(P, G)$, i.e., $O(n)$ if G is sparse and P is tree-like. This is true for real-world applications.
- Implementation: Homomorphism mappings for each unit pattern can be precomputed and stored similar to edge list.

Complexity, Implementation, and Experiments

- Complexity: DHN runs in the same time complexity as computing $\hom(P, G)$, i.e., $O(n)$ if G is sparse and P is tree-like. This is true for real-world applications.
- Implementation: Homomorphism mappings for each unit pattern can be precomputed and stored similar to edge list.

	#params	CSL	EXP	SR25	ENZYMES	PROTEINS
MPNN (4 layers)	27k	0	0	0	54.6 ± 4.5	72.0 ± 4.0
PPGN (4 layers)	96k	100	100	0	58.2 ± 5.7	77.2 ± 3.7
I^2-GNN (4 layers)	143k	100	100	100	-	-
N^2-GNN (4 layers)	355k	100	100	100	-	-
DHN–$(C_{2:4})$	5k	100	50	0	64.3 ± 5.5	76.5 ± 3.0
DHN–$(C_{2:5})$	7k	100	81	0	63.7 ± 5.4	77.0 ± 3.2
DHN–$(C_{2:10})$	27k	100	98	0	58.0 ± 5.3	78.5 ± 2.5
DHN–$(C_2K_{3:5})$	7k	100	50	53	63.3 ± 5.5	76.0 ± 2.7
DHN–$(C_{2:4}, C_2)$	8k	100	50	0	64.4 ± 5.9	77.1 ± 2.8
DHN–$(C_{2:5}, C_2)$	11k	100	99	0	62.0 ± 5.5	77.0 ± 2.5
DHN–$(C_{2:5}, C_{2:5})$	36k	100	99	0	59.9 ± 5.2	76.7 ± 3.3
DHN–$(C_{5:10}, C_2)$	27k	100	100	0	63.5 ± 6.1	78.2 ± 3.3
DHN–$(C_2K_{3:5}, C_2K_{3:5})$	36k	100	100	100	57.5 ± 6.6	77.4 ± 3.4

Conclusion

Implication of our main theorem: Let k be the tree-width of pattern P. Then DHN is:

- Strictly more expressive than 1-WL if P contains a single-edge pattern,
- Incomparable with k'-WL for $k' < k$,
- Less expressive than k-WL
- See our manuscript for comparison with other GNN models.

Conclusion and future work:

- Stacking homomorphism layers leads to powerful models
- Future work will study how graph pooling and attention can help realizing the potential of DHN.

Thank you for listening!