Can We Go Lower Order by Going Higher Order?

August 12, 2025

Marek Černý

University of Antwerp
marek.cerny@uantwerpen.be
https://marekcerny.com

Overview

- **What** is “lower-order”?

  …and why “lower-order”?

- **How** to use higher-order structures?

- Experiments and Future Work

Graphs and Homomorphisms

Let $G = (V(G), E(G))$ be a (simple, undirected) graph, and $F = (V(F), E(F))$ other graph.

A map

$$\varphi \colon V(F) \to V(G)$$

is a *homomorphism* $F \to G$ if

$$uv \in E(F) \implies \varphi(u)\varphi(v) \in E(G).$$

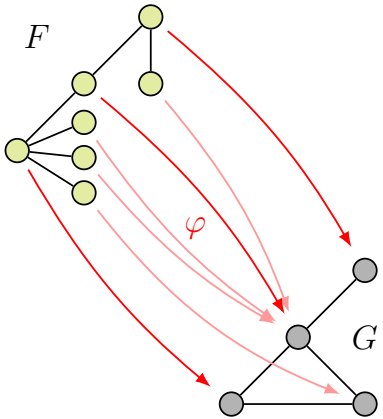


Figure: Example $\varphi \colon F \to G$.

Homomorphisms and Graph Neural Networks

For input G, homomorphism counts

$$\#\mathsf{hom}(F, G) = |\{\varphi \colon F \to G\}|$$

over all graphs F determine G (Lovász, 1967)[a].

Restricting

$$F \in \mathcal{F} \quad \text{(a subclass of all graphs)}$$

provides *practical bounds* on the expressivity of graph neural networks (GNNs).

[a] up to isomorphism

Homomorphisms and Graph Neural Networks

For input G, homomorphism counts

$$\#\mathsf{hom}(F, G) = |\{\varphi \colon F \to G\}|$$

over all graphs F determine G (Lovász, 1967)[a].

Restricting

$$F \in \mathcal{F} \quad \text{(a subclass of all graphs)}$$

provides *practical bounds* on the expressivity of graph neural networks (GNNs).

[a] up to isomorphism

GNN Type	**Class of graphs $\mathcal{F}$**
Message-Passing	trees[4]
Higher-Order (k-WL)[1]	treewidth[4] $\leqslant k$
Ordered Subgraph[2]	apex trees[5]
Spectral Invariant[3]	parallel trees[6]
Caterpillar **(ours)**	caterpillars

Table: Related work: [1](Morris et al., 2019), [2](Qian et al., 2022; Frasca et al., 2022), [3](Zhang et al., 2024). [4](Dvořák, 2010), [5](Rattan and Seppelt, 2023), [6](Gai et al., 2025), ours: (Černý, 2025).

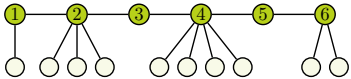
Caterpillar Graphs

Caterpillar graphs of height h,

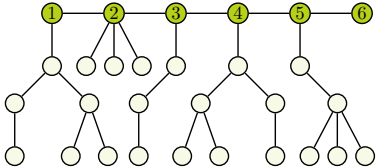
$$\mathcal{C}_h$$

consist of a (long) *spine*
with (short) *legs* of height h attached.

Caterpillars are a **strict subset** of trees,
and therefore of **lower order**
than Message-Passing GNNs.



$F \in \mathcal{C}_1$



$F \in \mathcal{C}_3$

Figure: Example of caterpillar graphs.

Motivation for Lower Order

Lower order imposes **stronger** inductive biases for information aggregation on graphs.

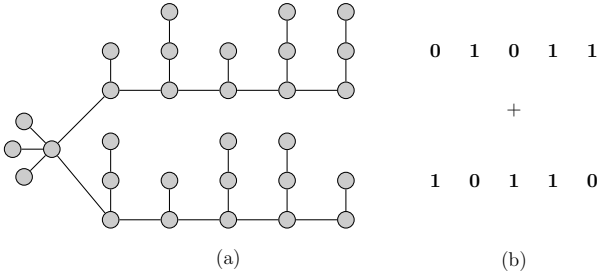
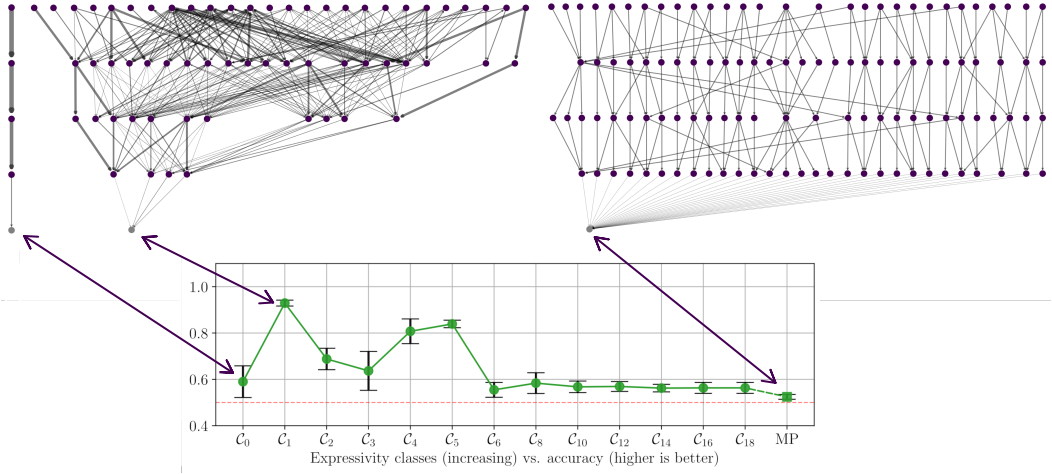


Figure: Binary encoding of $11 + 22$ as a graph G. *Classification problem*: does G represent the sum 33?

Computational Graphs (Caterpillar GNN vs. Message-Passing GNN)



Going Higher Order

Colored walks

Graph $G = (V, E)$ with vertices *colored by*

$$\chi\colon V \to \Sigma.$$

A *walk* in G is a sequence of vertices

$$(u_1, u_2, \ldots, u_t) \quad \text{s.t. } u_i u_{i+1} \in E.$$

A *walk color* is a *word*

$$c_1 c_2 \cdots c_t \in \Sigma^* \quad \text{s.t. } c_i = \chi(u_i).$$

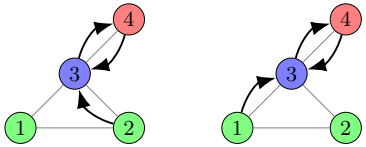


Figure: Graph $G = (V, E)$ colored by $\chi\colon V \to \{r, g, b\}$. Two walks $(1, 3, 4, 3)$ and $(2, 3, 4, 3)$ of the same color *gbrb*.

Colored walks and Caterpillars 

Theorem (Exact correspondence)

For all graphs G and H, and h in $\mathbb{N}$, the following is equivalent

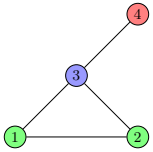
(i) G and H have the same numbers of homomorphisms from caterpillars of height h,

$$\#\mathsf{hom}(F, G) = \#\mathsf{hom}(F, H) \quad \text{for all } F \text{ in } \mathcal{C}_h.$$

(ii) G and H colored by $\chi^{(h)}$ contain the same numbers of colored walks,

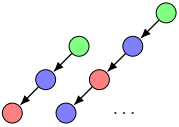
*where $\chi^{(h)}$ denotes the vertex colors of the h-**th** step of **color refinement**.*

Incidence matrix of vertices and colored walks



graph $G = (V, E)$,
colored by $\chi\colon V \to \Sigma$

colored walks Σ^*



vertices V	λ						⋯			⋯
①	**1**	**1**	0	0	**1**	0	⋯	0	0	⋯
②	**1**	**1**	0	0	**1**	0	⋯	0	0	⋯
③	**1**	0	0	**1**	0	**2**	⋯	0	**2**	⋯
④	**1**	0	**1**	0	0	0	⋯	**2**	0	⋯

incidence matrix $W \in \mathbb{N}^{V \times \Sigma^*}$

(Inefficient) Graph Matrices

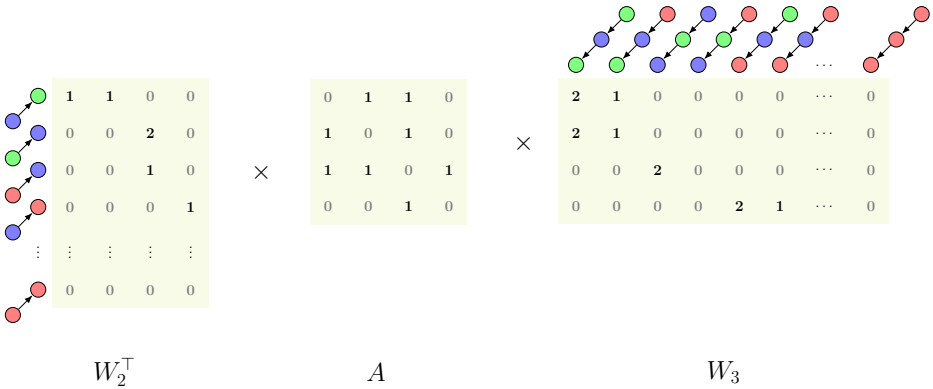


Figure: "Locking spaces" of adjacency matrix by $W_t^T A W_{t+1}$ for $t := 2$.

Efficient Graph Matrices

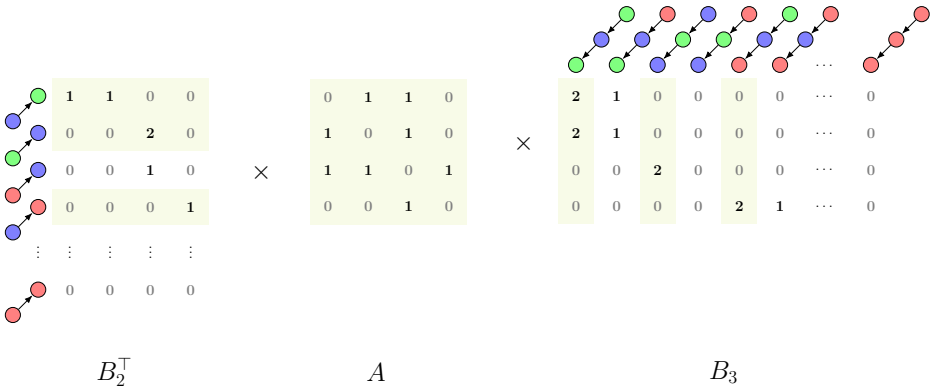


Figure: "Locking spaces" of adjacency matrix by $B_t^T A B_{t+1}$ for $t := 2$.

Replacing Message Passing

Standard Message-Passing — **fixed** vertices V **copied** to all layers.

Efficient aggregation — **multisets** of vertices in $\mathbb{N}^V$ **specific** for each layer.

[1] non-strict inequality, but typically smaller

Replacing Message Passing

Standard Message-Passing — **fixed** vertices V **copied** to all layers.

Efficient aggregation — **multisets** of vertices in $\mathbb{N}^V$ **specific** for each layer.

- Selected multisets avoid *linear dependencies* (as columns of incidence matrix W in $\mathbb{N}^{V \times \Sigma^*}$).

[1] non-strict inequality, but typically smaller

Replacing Message Passing

Standard Message-Passing — **fixed** vertices V **copied** to all layers.

Efficient aggregation — **multisets** of vertices in $\mathbb{N}^V$ **specific** for each layer.

- Selected multisets avoid *linear dependencies* (as columns of incidence matrix W in $\mathbb{N}^{V \times \Sigma^*}$).
- In every layer, we replace *adjacency* and *identity* matrices by smaller[1] ones

$$A \mapsto C_t^A \coloneqq B_t^+ A B_{t+1}, \qquad I \mapsto C_t^I \coloneqq B_t^+ A B_{t+1}.$$

[1] non-strict inequality, but typically smaller

Replacing Message Passing

Standard Message-Passing — **fixed** vertices V **copied** to all layers.

Efficient aggregation — **multisets** of vertices in $\mathbb{N}^V$ **specific** for each layer.

- Selected multisets avoid *linear dependencies* (as columns of incidence matrix W in $\mathbb{N}^{V \times \Sigma^*}$).
- In every layer, we replace *adjacency* and *identity* matrices by smaller[1] ones

$$A \mapsto C_t^A \coloneqq B_t^+ A B_{t+1}, \qquad I \mapsto C_t^I \coloneqq B_t^+ A B_{t+1}.$$

- Fully preserving *equivariance* and *determinism*.

[1] non-strict inequality, but typically smaller

Computational Graphs: Explained

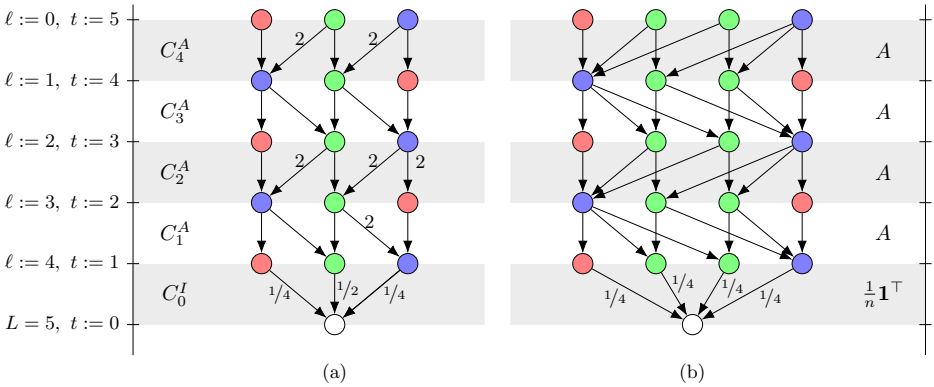


Figure: Comparison of computational graphs (without self-loops): (a) *efficient aggregation* (ours), and (b) *standard message-passing*. Each consists of L layers for the same input graph G as in fig. 7.

Experiments

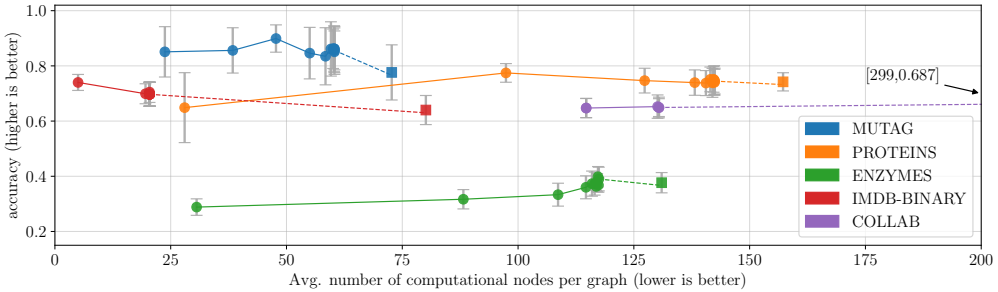


Figure: **Number of Computational Nodes vs. Accuracy.** Caterpillar GNN: circles starting from $\mathcal{C}_0$, $\mathcal{C}_1$ up to $\mathcal{C}_{10}$. Message-Passing GCN: square.

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).
- Lifting Caterpillars to Higher Order?

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).
- Lifting Caterpillars to Higher Order? (that Went Lower Order

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).
- Lifting Caterpillars to Higher Order? (that Went Lower Order by Going Higher Order)

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).
- Lifting Caterpillars to Higher Order? (that Went Lower Order by Going Higher Order) Yes!

Future Work

- Almost any Message-Passing GNN can be “caterpillarized”.
- More efficient precomputation (now $O(L|\Sigma|n^3)$).
- Lifting Caterpillars to Higher Order? (that Went Lower Order by Going Higher Order) Yes!
- Invitation to **structural graph learning**!

References I

Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64, 2010.

Fabrizio Frasca, Beatrice Bevilacqua, Michael M. Bronstein, and Haggai Maron. Understanding and extending subgraph GNNs by rethinking their symmetries. *CoRR*, 2022.

Jingchu Gai, Yiheng Du, Bohang Zhang, Haggai Maron, and Liwei Wang. Homomorphism expressivity of spectral invariant graph neural networks, 2025. URL https://arxiv.org/abs/2503.00485.

L. Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18 (3):321–328, sep 1967. ISSN 1588-2632. doi: 10.1007/BF02280291. URL https://doi.org/10.1007/BF02280291.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, pages 4602–4609, 2019.

References II

C. Qian, G. Rattan, F. Geerts, C. Morris, and M. Niepert. Ordered subgraph aggregation networks. In *Advances in Neural Information Processing Systems*, 2022.

Gaurav Rattan and Tim Seppelt. Weisfeiler-leman and graph spectra. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2268–2285. SIAM, 2023. URL https://doi.org/10.1137/1.9781611977554.ch87.

Marek Černý. Caterpillar gnn: Replacing message passing with efficient aggregation, 2025. URL https://arxiv.org/abs/2506.06784.

Bohang Zhang, Lingxiao Zhao, and Haggai Maron. On the expressive power of spectral invariant graph neural networks, 2024. URL https://arxiv.org/abs/2406.04336.

Multisets and vectors

$\Sigma = \{c, m, y, k\}$ is a *finite set*,

$M = \{\!\{c, m, m, m, m, y, y\}\!\}$ is a *multiset*,

with multiplicities

$$M[c] = 1, M[m] = 4, M[y] = 2, M[k] = 0.$$

Multisets can be embedded as vectors

$$M \text{ in } \mathbb{N}^{\Sigma} \subset \mathbb{R}^{\Sigma}.$$

Results Table I

type	MUTAG			PROTEINS			ENZYMES			IMDB-BINARY			COLLAB		
	mean ± std	#n.	%s.	mean ± std	#n.	%s.	mean ± std	#n.	%s.	mean ± std	#n.	%s.	mean ± std	#n.	%s.
$\mathcal{C}_0$	0.851 ± 0.091	23.7	67.4	0.649 ± 0.127	28.0	82.2	0.288 ± 0.030	30.6	76.6	**0.740 ± 0.029**	5.0	93.8	0.647 ± 0.035	114.7	61.6
$\mathcal{C}_1$	0.856 ± 0.082	38.4	47.2	**0.774 ± 0.034**	97.4	38.1	0.317 ± 0.035	88.2	32.7	0.699 ± 0.036	19.6	75.6	0.647 ± 0.035	114.7	61.6
$\mathcal{C}_2$	**0.899 ± 0.050**	47.7	34.4	0.747 ± 0.045	127.3	19.0	0.333 ± 0.042	108.6	17.1	0.705 ± 0.037	20.4	74.5	0.652 ± 0.042	130.1	56.5
$\mathcal{C}_3$	0.846 ± 0.093	55.0	24.4	0.739 ± 0.046	138.2	12.1	0.360 ± 0.042	114.7	12.5	0.697 ± 0.043	20.5	74.5	0.649 ± 0.032	130.4	56.4
$\mathcal{C}_4$	0.835 ± 0.103	58.4	19.7	0.738 ± 0.045	140.6	10.6	0.373 ± 0.045	116.0	11.5	0.697 ± 0.043	20.5	74.5	0.649 ± 0.037	130.4	56.4
$\mathcal{C}_5$	0.861 ± 0.098	59.6	18.1	0.751 ± 0.045	141.5	10.0	0.373 ± 0.042	116.5	11.0	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
$\mathcal{C}_6$	0.862 ± 0.071	60.1	17.4	0.739 ± 0.052	142.0	9.7	0.363 ± 0.015	116.9	10.8	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
$\mathcal{C}_7$	0.862 ± 0.083	60.3	17.1	0.751 ± 0.047	142.2	9.5	0.377 ± 0.033	117.1	10.6	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
$\mathcal{C}_8$	0.851 ± 0.076	60.3	17.0	0.748 ± 0.052	142.3	9.5	**0.398 ± 0.037**	117.2	10.5	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
$\mathcal{C}_9$	0.862 ± 0.076	60.4	17.0	0.740 ± 0.047	142.4	9.4	0.368 ± 0.026	117.4	10.4	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
$\mathcal{C}_{10}$	0.856 ± 0.087	60.4	17.0	0.745 ± 0.047	142.5	9.4	0.390 ± 0.043	117.4	10.4	0.697 ± 0.043	20.5	74.5	0.651 ± 0.035	130.4	56.4
MP	0.776 ± 0.100	72.7	0.0	0.742 ± 0.033	157.2	0.0	0.377 ± 0.037	131.0	0.0	0.640 ± 0.053	80.1	0.0	**0.687 ± 0.043**	299.0	0.0

Table: Results for *graph-level classification* datasets. By $\mathcal{C}_h$ is denoted the (caterpillar height) parameter h in $\mathbb{N}$ of efficient aggregation (ours), while MP denotes the full message-passing GCN. We report mean validation accuracy with standard deviation of different 10 splits. Columns "#n." denotes number of *nodes* of the computation graph, and columns "%s." percent of nodes of the computation graph *saved* comparing to message-passing (MP).

Results Table II

type	FreeSolv			Lipo			ESOL			ZINC		
	mean ± std	#n.	%s.	mean ± std	#n.	%s.	mean ± std	#n.	%s.	mean ± std	#n.	%s.
$\mathcal{C}_0$	2.693 ± 0.223	24.3	32.5	1.116 ± 0.108	77.3	29.1	1.664 ± 0.106	35.8	34.0	0.570 ± 0.012	65.7	29.8
$\mathcal{C}_1$	2.693 ± 0.223	24.3	32.5	1.116 ± 0.108	77.3	29.1	1.664 ± 0.106	35.8	34.0	0.571 ± 0.006	65.7	29.8
$\mathcal{C}_2$	2.704 ± 0.237	26.5	26.5	1.069 ± 0.043	87.0	20.3	1.666 ± 0.212	40.1	26.0	0.584 ± 0.007	75.9	18.9
$\mathcal{C}_3$	2.576 ± 0.283	27.5	23.7	1.105 ± 0.098	93.7	14.1	1.516 ± 0.100	42.6	21.4	0.602 ± 0.003	82.5	11.9
$\mathcal{C}_4$	2.630 ± 0.350	27.8	22.8	1.051 ± 0.079	95.8	12.3	1.410 ± 0.152	43.4	20.0	0.570 ± 0.012	84.6	9.6
$\mathcal{C}_5$	2.597 ± 0.178	27.9	22.6	1.097 ± 0.067	96.3	11.8	1.458 ± 0.102	43.6	19.6	0.551 ± 0.012	85.1	9.1
$\mathcal{C}_6$	2.650 ± 0.266	27.9	22.6	1.070 ± 0.077	96.5	11.6	1.528 ± 0.207	43.6	19.5	0.546 ± 0.013	85.2	9.0
$\mathcal{C}_7$	2.719 ± 0.206	27.9	22.6	1.072 ± 0.111	96.5	11.6	1.446 ± 0.101	43.7	19.4	0.539 ± 0.012	85.2	9.0
$\mathcal{C}_8$	2.676 ± 0.196	27.9	22.6	1.108 ± 0.104	96.5	11.6	**1.392 ± 0.129**	43.7	19.4	0.543 ± 0.006	85.2	9.0
$\mathcal{C}_9$	2.652 ± 0.284	27.9	22.6	1.094 ± 0.069	96.5	11.6	1.416 ± 0.148	43.7	19.4	0.538 ± 0.003	85.2	9.0
$\mathcal{C}_{10}$	2.675 ± 0.276	27.9	22.6	1.063 ± 0.091	96.5	11.6	1.514 ± 0.142	43.7	19.4	0.528 ± 0.004	85.2	9.0
MP	**2.550 ± 0.374**	36.0	0.0	**1.031 ± 0.059**	109.2	0.0	1.449 ± 0.140	54.2	0.0	**0.477 ± 0.020**	93.6	0.0

Table: Results for *graph-level regression* datasets. By $\mathcal{C}_h$ is denoted the (caterpillar height) parameter h in $\mathbb{N}$ of efficient aggregation (ours), while MP denotes the full message-passing GCN. We report validation mean absolute error (MAE) with standard deviation of different 10 splits and random seeds, in the case of ZINC of distinct 5 random seed repetitions. Columns "#n." denotes number of *nodes* of the computation graph, and columns "%s." percent of nodes of the computation graph *saved* comparing to message-passing (MP).