ICML
International Conference
On Machine Learning

Weisfeiler Leman for Euclidean Equivariant Machine Learning

By Snir Hordan, Dr. Tal Amir, and Dr. Nadav Dym

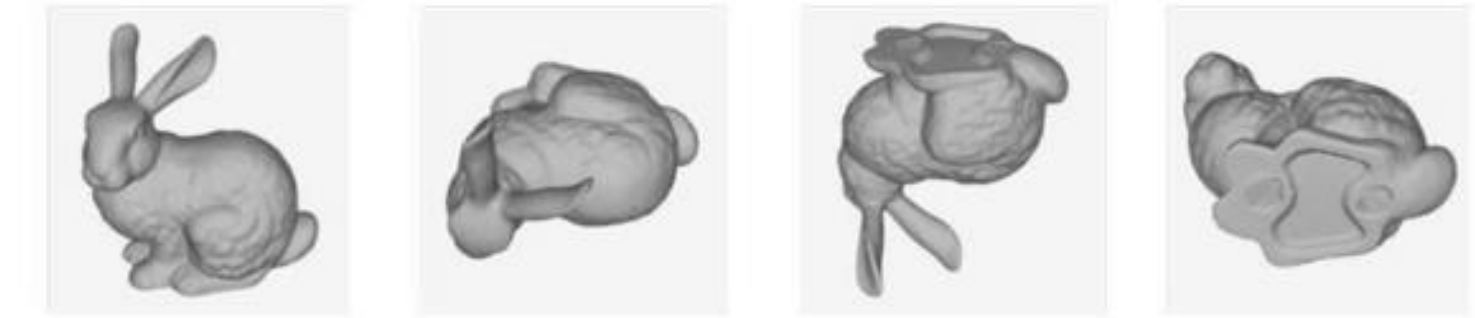
Technion

Preliminaries

Point Cloud – A collection of n points in Euclidean space considered up to permutation symmetries, that is $X \in \mathbb{R}^{3\times n}$ such that $\sigma \cdot X = X \quad , \forall \sigma \in S_n$, where σ permutes the columns of X.

Geometric Symmetries – We say that two point clouds, $X, Y \in \mathbb{R}^{3\times n}$, are *isometric* if there exist a permutation $P \in \mathbb{R}^{n\times n}$, a translation vector $t \in \mathbb{R}^3$, and an orthogonal action $O \in \mathbb{R}^{3\times 3}$, such that

$$X = OYP^T + t$$



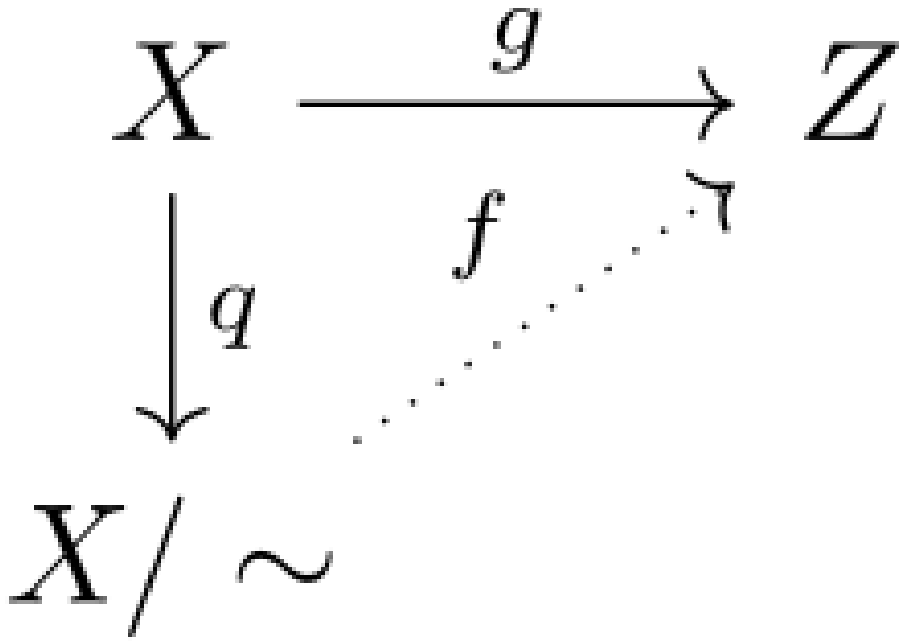
'A rotated bunny remains a bunny'

Preliminaries (continued)

The universal property: If $g: X \to Z$ is a continuous invariant function and $q: X \to X/\sim$ ($\sim$ is an equivalence relation) injectively embeds elements up to symmetries, then there exists a unique continuous

$$f: X/\sim \to Z$$

such that $g = f \circ q$.



How do we process point clouds?

Observe the mapping

$$\text{Dist}: \mathbb{R}^{3\times n} \to \mathbb{R}^{n\times n}$$

$$\text{Dist}(\text{X}) = \left(||\text{x}_\text{i} - \text{x}_j||\right)_{ij}$$

It is the distance matrix induced by an ordered list of points.

(Equivalently: centralize and $< \text{x}_i, \text{x}_j >$)

Satorras et al. (2021) have shown that the distance matrix is a complete descriptor of an ordered list of points with respect to joint Orthogonal and Translation group actions.

That is $N = \text{gM} + \text{t}, g \in O(3), t \in \mathbb{R}^3,$ if and only if $\text{Dist}(\text{M}) = \text{Dist}(\text{N})$

Bottleneck: Permutation symmetries imply the same point cloud can be identified with n! pairwise distance matrices.

Weisfeiler Leman Test

The Weisfeiler Leman (1968) graph isomorphism test is a longstanding relaxation of the task of graph isomorphism testing.

Let $G = (V, E)$ be a graph with $|V| = n, |E| = m$.

WL first initializes a uniform 'coloring' for each node,

denoted as $c_v^{(0)}, v \in V$, and iteratively updates this coloring via

$$c_v^{(t+1)} = \text{HASH}(c_v^{(t)}, \{(c_w^{(t)}, \delta_{w \in N(v)}) \mid w \in V\})$$

Then after T iterations, we apply a readout function

$$c_G^{(T)} = \text{HASH}(\{c_v^{(T)} \mid v \in V\})$$

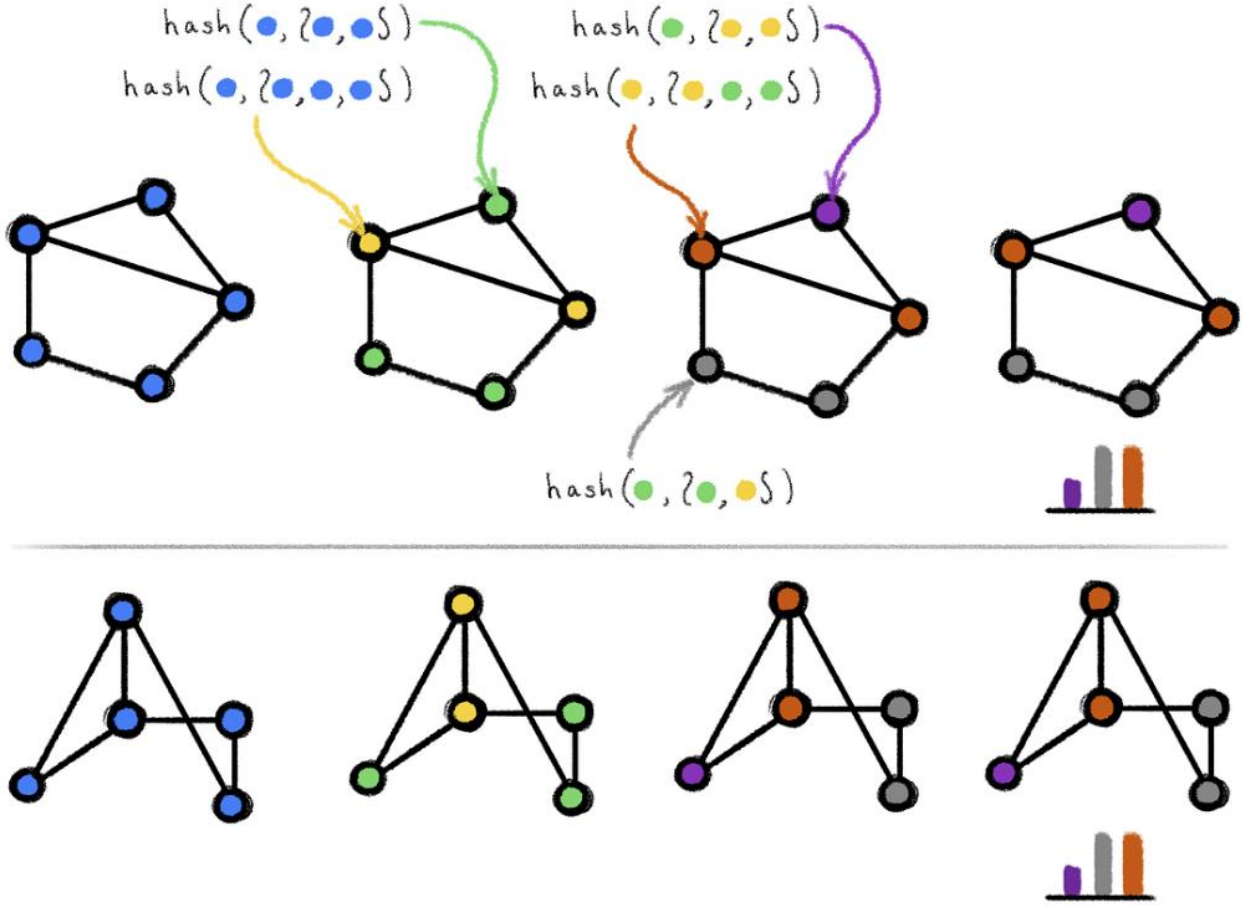
A. Leman B. Weisfeiler

Portraits: Ihor Gorsky (From Michael Bronstein's blog)

Podznyakov et al. defined a Euclidean variant of this test, named 1-EWL, where the update is

$$c_v^{(t+1)} = \text{HASH}(c_v^{(t)}, \{ (c_w^{(t)}, ||x_v - x_w||_2) \mid w \in V\})$$

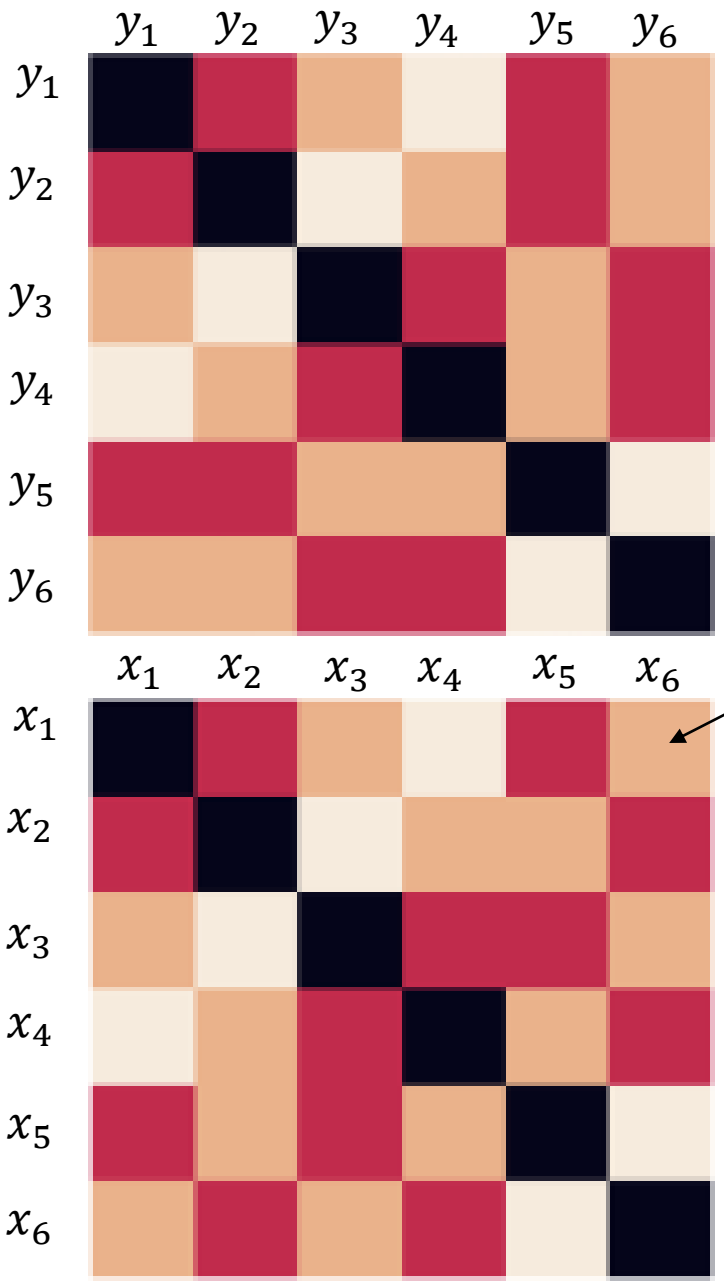
Weisfeiler-Leman Test Simulation



Example of execution of the Weisfeiler-Lehman test on two isomorphic graphs. Curled brackets denote multisets. The algorithm stops after the colouring does not change and produces an output (histogram of colours). Equal outputs for the two graphs suggest that they are possibly isomorphic.

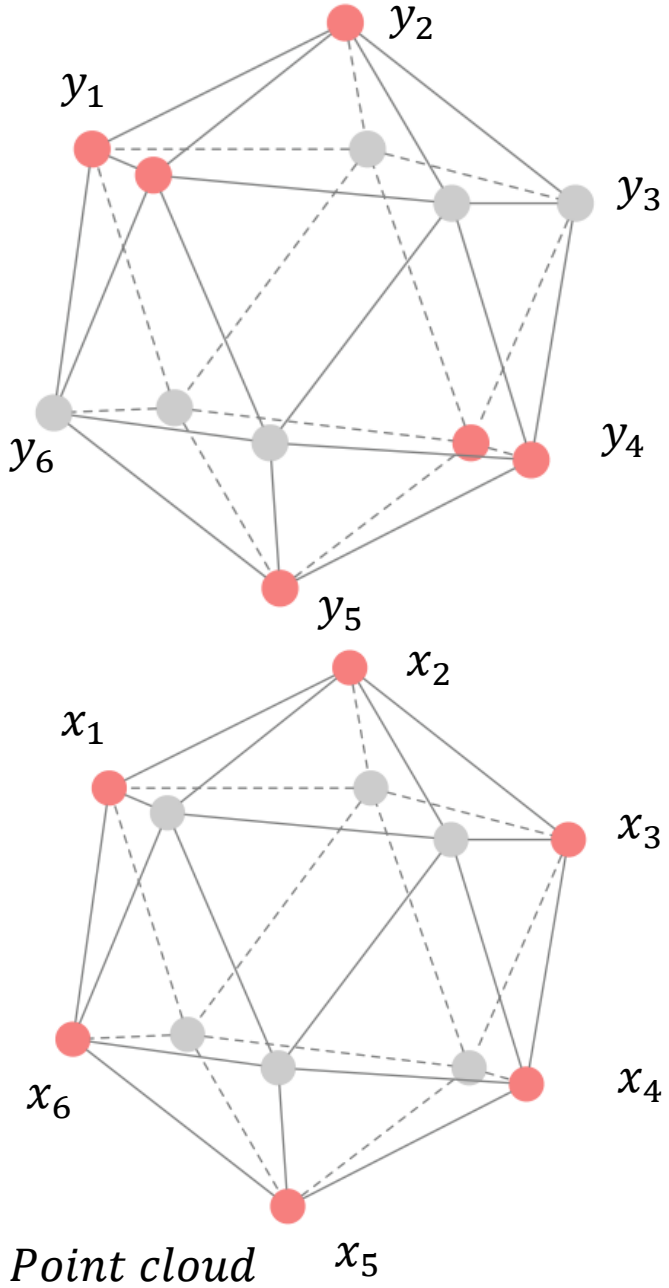
Credit : Michael Bronstein

Counterexamples



Distance graph of the point cloud

$||x_1 - x_6||$



Point cloud

Li et al 23′

2-(F)WL

The WL test is naturally extended to high-order variations, we focus on the *Folklore* variant. Many works denote 2-WL as 2-*F*WL, I will refer to the Folklore variant simply as 2-WL.

We assign an initial (uniform) coloring to each 2-tuple (edge) $\boldsymbol{C_0}(\mathbf{i})$, where $\mathbf{i} = (i_1, i_2) \in [n]^2$, of nodes in the graph.

Then we iteratively update this coloring via

$$\boldsymbol{C_{t+1}}((i_1, i_2)) = \mathbf{HASH}(\boldsymbol{C_t}((i_1, i_2)), \{ (\boldsymbol{C_t}(i_1, j), \boldsymbol{C_t}(j, i_2)) : j \in [n] \})$$

This process is repeated T times (predetermined or histogram of colors stays constant), then we apply a readout function

$$\boldsymbol{C_G}(\mathbf{i}) = \mathbf{HASH}(\{\boldsymbol{C_T}(\mathbf{i}) \ : \ \mathbf{i} \in [n]^2\})$$

To obtain a permutation invariant global graph feature.

We can *simulate* these tests via Graph Neural Networks by replacing the **HASH** functions with neural-network based aggregation schemes. *2-WL, when applied to the distance matrix induced by a point cloud, is complete (Delle Rose et al. NeurIPS 2023).*

Equivariant Functions on Point Clouds

We wish to define joint permutation, orthogonal, and translation equivariant functions on point clouds.

Formally,

$$f: \mathbb{R}^{3\times n} \to \mathbb{R}^{3\times n}$$

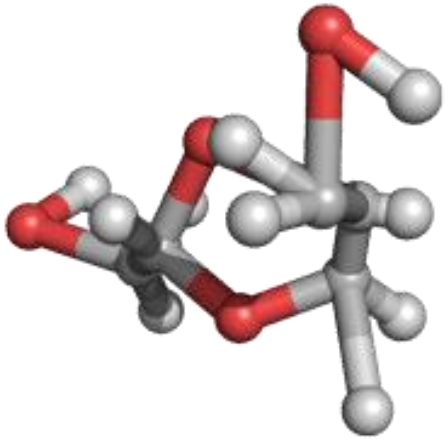
$$f(RXP^T + t)_i = Rf(X)_{\sigma^{-1}(i)} + t$$

Where $P \in \mathbb{R}^{n\times n}$ is a permutation, $t \in \mathbb{R}^3$ is a translation vector and $R \in \mathbb{R}^{3\times 3}$ is an orthogonal action.

This comes up in the N-body problem, chemical simulations and more.

ICML
International Conference
On Machine Learning

Weisfeiler Leman for Euclidean Equivariant Machine Learning



Animation by Shi et al. 2021

Weisfeiler Leman for Euclidean Equivariant Machine Learning

Main contributions:

- An *instantiation* of PPGN (Maron et al.) achieves a *uniformly injective* simulation of 2-WL on point clouds with O(n^3) running time.
- 2-WL achieves equivariant universality with a simple pooling operator.
- WeLNet - Model with competitive performance in generative and dynamic modeling (N-Body) tasks.

1st Result: Simulation of 2-WL via PPGN

We aim to simulate the aggregation $\{\big(C(i,k), C(k,j)\big) | \; k \in [n]\}$

Maron et al. (2019) made the connection to matrix multiplication:

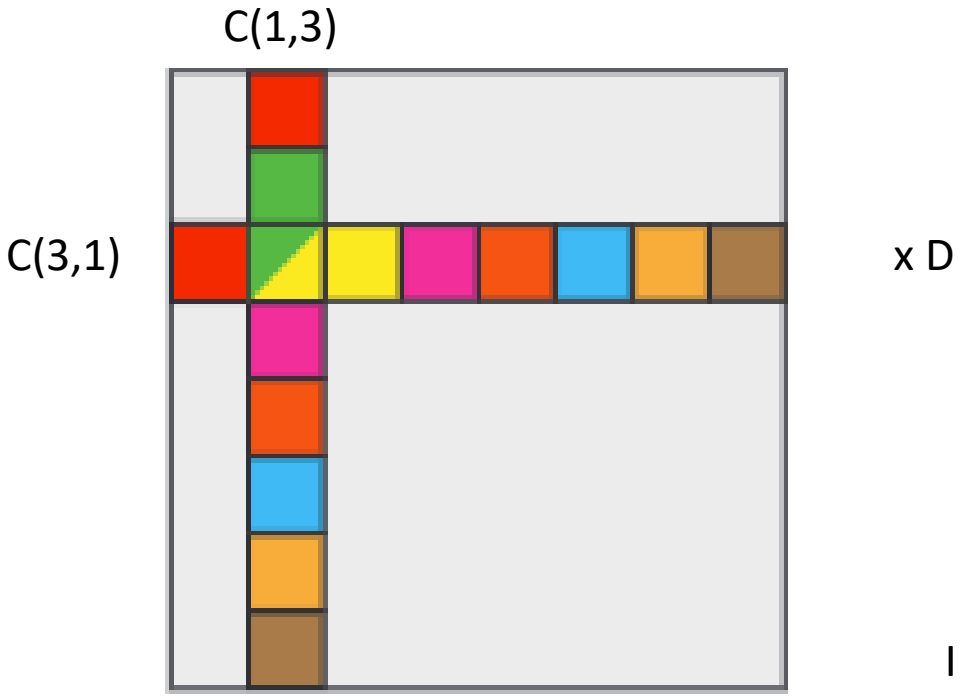


Image by Maron et al. 2019

How do we usually embed multi-sets?

Simulation of WL tests requires injectively embedding the neighborhood of each tuple considered.

For a given set of vectors $\{x_1, \dots, x_n\} \subset \mathbb{R}^D$, we can use MLPs to injectively embed this set:

$$f(\{x_1, \dots, x_n\}) = \sum_{k=1}^{n} \sigma(Ax_i) \in \mathbb{R}^{2D+1}$$

Where A is a matrix and σ is a *non-polynomial analytic* activation.

f is an injective (multi-)set function for almost any choice of A (Amir et al. NeurIPS '23).

For ReLU activation this result does not hold (Xu et al.)

PPGN

$$X^{(1)} = \phi_1\left(X^{in}\right) \in \mathbb{R}^{n\times n\times D}, \qquad X^{(2)} = \phi_2(X^{in}) \in \mathbb{R}^{n\times n\times D}$$

$$X_{ij}^{out} = \sum_{k=1}^{n} (X_{ik}^{(1)} \odot X_{kj}^{(2)}) \in \mathbb{R}^{D}$$

We prove for almost any choice of the parameters of ϕ_1, ϕ_2 (weights and biases) via neural injective moments (Amir et al.) and functional analysis:

X_{ij}^{out} is an injective embedding of $\left\{\left(X_{ik}, X_{kj}\right)\middle| k \in [n]\right\}$.

Any *instantiation* of PPGN *with analytic activations* simulates 2-WL!

Comparison: Simulation via transformed graph (Jogl et al.) requires $O(n^4)$ vs PPGN's $O(n^3)$.

2nd Result:
Weighted Summation is All You Need

Equivariant GNNs in literature often update positions via

$$f(X)_i = x_i + \sum_{j \neq i} g(X_{i,j})(x_i - x_j)$$

where g is a continuous, *rotation and translation invariant* vector-multiset scalar-valued function, $\mathrm{X}_{i,j} \stackrel{\text{def}}{=} \left(x_i, x_j, \{\, x_k \mid k \in [n] \backslash \{i,j\} \,\}\right)$ and $\sum_{j \neq i} g(\mathrm{X}_{i,j}) = 1$.

We show this is all that is required when simulating 2-WL.

Relation to Existing Literature

An abundance of equivariant GNNs use this aggregation procedure with various choices of the weighing function g.

Examples:

$g = f(h_i, h_j, ||x_i - x_j||_2)$ for a continuous f:

1. EGNN (ICML 2021)
2. PaiNN (ICML 2021)
3. DGSM (NeurIPS 2021)
4. ConfGF (ICML 2021)

$g = f(\text{Transformer} + \text{dist}(X))$:

1. UniMol (ICLR 2023)

Yet none are known to be universal!

Relation to 2-WL

Let $X \in \mathbb{R}^{3\times n}$ and $\mathrm{C}^{(5)}(i,j) \in \mathbb{R}^{D}$ be the pairwise coloring induced by 5 iterations of 2-WL update steps applied to the distance matrix induced by X.

Theorem $\mathrm{C}^{(5)}(i,j)$ constitutes a translation and orthogonal invariant injective embedding of $\mathrm{X}_{i,j} = (x_i, x_j, \{x_k \mid k \neq i,j\})$.

Using the Universal Property and UAT of MLPs:

$$f(X)_i = x_i + \sum_{j\neq i} h(\mathrm{C}^{(5)}(i,j))(x_i - x_j)$$

Is universal!

General framework: *any* injective embedding of $\mathrm{X}_{i,j}$ guarantees universality.

What about velocities?

Physical systems, such as electrically charged particles and planetary systems, have the additional information of instantaneous velocity.

We can attain completeness via 2-WL in this setting as well:

We initialize each edge feature as

$$C(i,j) = (||x_i^c - x_j^c||, ||x_i^c - v_j\ ||, ||v_i\ - x_j^c||, ||v_i\ - v_j\ ||)$$

Where x_j^c denotes the centralized x_j position.

That is, including the distances from the velocities to positions is required.

3rd Result: WeLNet

WeLNet is defined as a successive application of Euclidean equivariant convolution layers, which involve a parameter-sharing scheme.

Initialize:

1. Node-wise hidden states $h \stackrel{\text{def}}{=} \{h_i\}_{i=1}^{n}$
2. Shared $2-$WL-equivalent analytic PPGN architecture, PPGN_{an}

Convolution layer acts as

$$(h^{out}, X^{out}, V^{out}) = WeLConv(h, E, \mathrm{PPGN}_{an}, X, V)$$

WeLConv

We define the Convolution Layer of WeLNet, WeLConv, which is essentially an infusion of PPGN (Maron et al.) and EGNN (Satorras et al. 2021), defined as

$$c(i,j) = \mathrm{PPGN}_{\mathrm{an}}(X,V)_{i,j}$$

$$m_{ij} = \phi_e(h_i, h_j, e_{ij}, c(i,j))$$

$$m_i = \sum_j m_{ij}$$

$$x_i^{out} = x_i + \sum_j \psi_x(m_{ij})(x_i - x_j)$$

$$h_i = \phi_h(h_i, m_i)$$

Universality is guaranteed!

Experiments – N-Body Problem

METHOD	MSE
LINEAR	0.0819
SE(3) TRANSFORMER	0.0244
TFN	0.0155
GNN	0.0107
RADIAL FIELD	0.0104
EGNN	0.0071
CLOFNET	0.0065
FA-GNN	0.0057 ± 0.0002
CN-GNN	0.0043 ± 0.0001
SEGNN	0.0043 ± 0.0002
MC-EGNN-2	0.0041 ± 0.0006
TRANSFORMER-PS	0.0040 ± 0.00001
WELNET (OURS)	$\mathbf{0.0036 \pm 0.0002}$

Very popular task introduced by Sattoras et al. 2021.

We (WeLNet) reach state of the art on the N-Body simulation task (Satorras et al 2021).

N-Body with External Forces

Our construction outperforms other GNNs with a gravity force component, yet under external Lorentz force (magnetic field) WeLNet performs only as well as 1-WL equivalent models, despite higher computational cost. **Open problem.**

Table 2. Test MSE on the N-Body dynamic prediction task with Gravitational force and Lorentz-like force. We compare to ClofNet (Du et al., 2022), MC-EGNN (Levy et al., 2023) and EGNN (Victor Garcia Satorras, 2021).

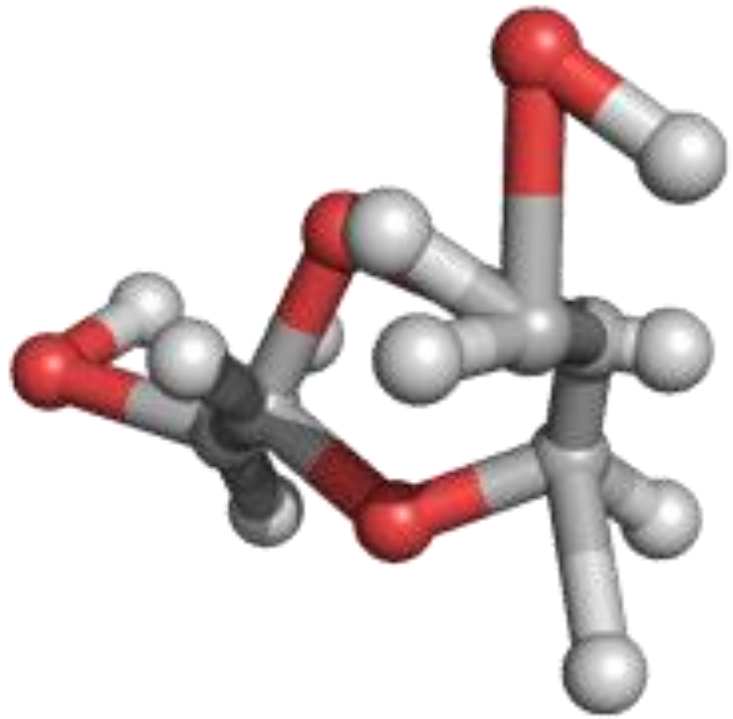
METHOD	GRAVITY	LORENTZ
GNN	0.0121	0.02755
EGNN	0.0906	0.032
CLOFNET	0.0082 ± 0.0003	0.0265± 0.0004
MC-EGNN	0.0073± 0.0002	**0.0240± 0.0010**
WELNET (OURS)	**0.0054 ± 0.0001**	**0.0238± 0.0002**

Experiment – Conformation Generation

The setup of the problem is as follows:

Given a molecular graph, that is a pair (V,E) with atomic numbers decorating V and edges with chemical bond information.

Recover a point cloud $X \in \mathbb{R}^{n\times 3}$ that represents a stable molecular conformation.



Visualization by Shi, Chence et al.

Experiments – GEOM-QM9

Model	COV (%) ↑		MAT (Å) ↓	
	Mean	Median	Mean	Median
RDKit	83.26	90.78	0.3447	0.2935
CVGAE	0.09	0	1.6713	1.6088
GraphDG	73.33	84.21	0.4245	0.3973
CGCF	78.05	82.48	0.4219	0.39
ConfVAE	80.42	85.31	0.4066	0.3891
ConfGF	88.49	94.13	0.2673	0.2685
GeoMol	71.26	72	0.3731	0.3731
DGSM	91.49	95.92	0.2139	0.2137
ClofNet	90.21	93.14	0.2430	0.2457
GeoDiff	92.65	95.75	0.2016	0.2006
DMCG	94.98	98.47	0.2365	0.2312
UniMol	**97.00**	**100.00**	0.1907	0.1754
WeLNet (Ours)	92.66	95.29	**0.1614**	**0.1566**

We (WeLNet) achieve a new state of the art result on MAT target in GEOM-QM9 molecular conformation generation task.

Future Research

- Scalability - simulate 2-WL much more efficiently using a sparse variation with distance cutoffs.
- Integration into diffusion/flow matching training. How can we use the symmetries of 2-WL for novel flow matching?

Thank you!