# Wasserstein Weisfeiler-Lehman

# Graph Kernels

Matteo Togninalli, Felipe Llinares-López, Elisabetta Ghisu, Bastian Rieck, Karsten Borgwardt – ETH ZURICH

Fabrice Beaumont

Rheinische Friedrich-Wilhelms-Universität Bonn

01.07.2020

# Motivation

**TASK:** Graph classification (via graph kernels as similarity measure)

**Solution:** $\mathfrak{R}$-Convolution kernels

**Problem:** Naive aggregation of substructures may disregard valuable information

Only a few approaches extendible for continuously attributed graphs

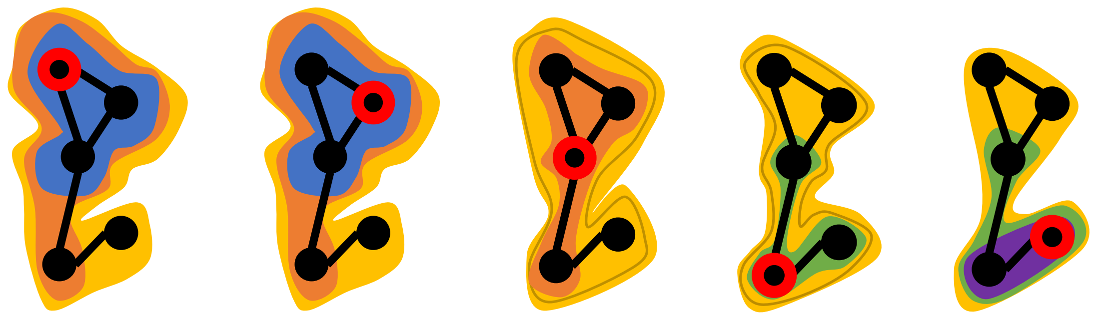
# Method Overview

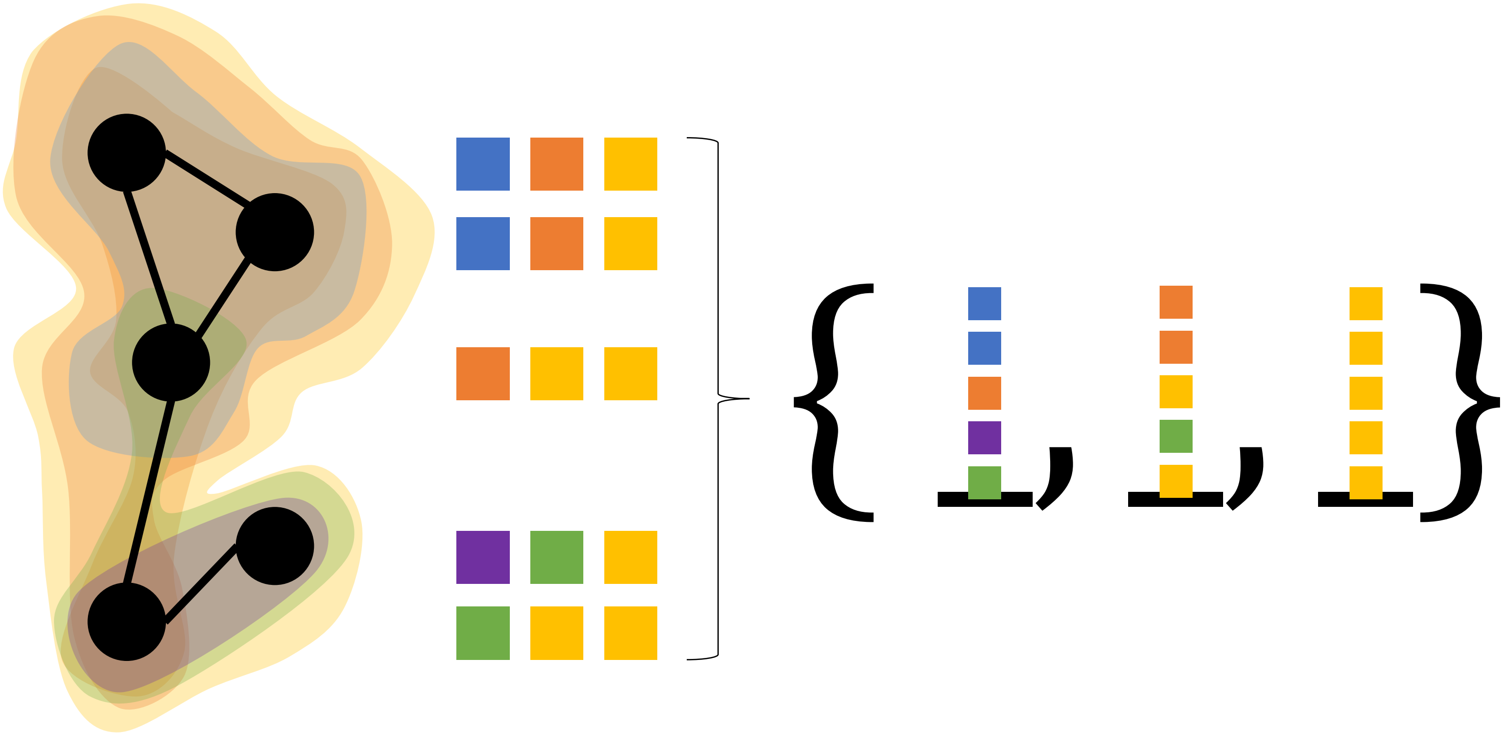
1. Transform each graph into a set of node embeddings
   1. Weisfeiler-Lehman node embedding scheme
   2. WL-features
2. Measure the Wasserstein distance between each pair of graphs
   1. Ground distance (similarity matrix)
   2. Wasserstein distance
3. Wasserstein Weisfeiler-Lehman kernel definition

**Optimal transport**

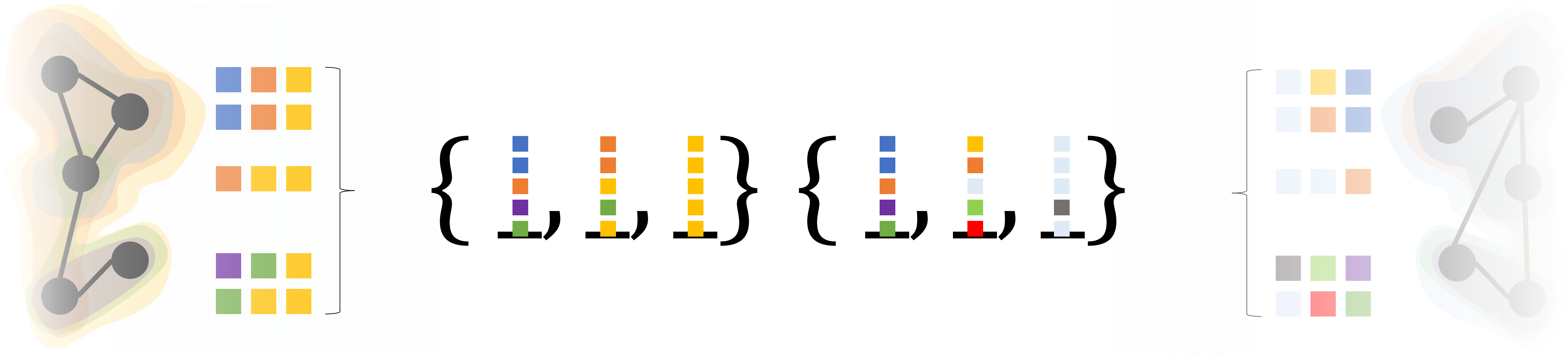
(Differentiate between a finite set of categorical **node labels** and real valued **node attributes**)

# 1.1 Weisfeiler-Lehman node embedding scheme
## Idea





# 1.1 Weisfeiler-Lehman node embedding scheme
# Idea



# 1.1 Weisfeiler-Lehman node embedding scheme Theory

$m$: dimensionality of the node labels

$n_G$: cardinality of the vertex set

$N(v)$: neighbourhood of $v$

$$N^h(v) = \{l^h(u_0), \dots, l^h(u_{\deg(v)-1})\}$$

| NODE LABELS | NODE ATTRIBUTES |
|---|---|
| $l^{h+1}(v) = hash(l^h(v), N^h(v))$ | $a^{h+1}(v) = \frac{1}{2}\left(a^h(v) + \frac{1}{\deg(v)} \sum_{u \in N(v)} w((u,v))\, a^h(u)\right)$ $\in \mathbb{R}^m$ |

**WL-features** $X_G^h = \left[x^h(v_1), \dots, x^h(v_{n_G})\right]^T$ $\in \mathbb{R}^{n_G \times m}$

**Node embedding** $f_{WL}^H: G \longrightarrow \left(X_G^0, \dots, X_G^H\right)$ $\in \mathbb{R}^{n_G \times m(H+1)}$

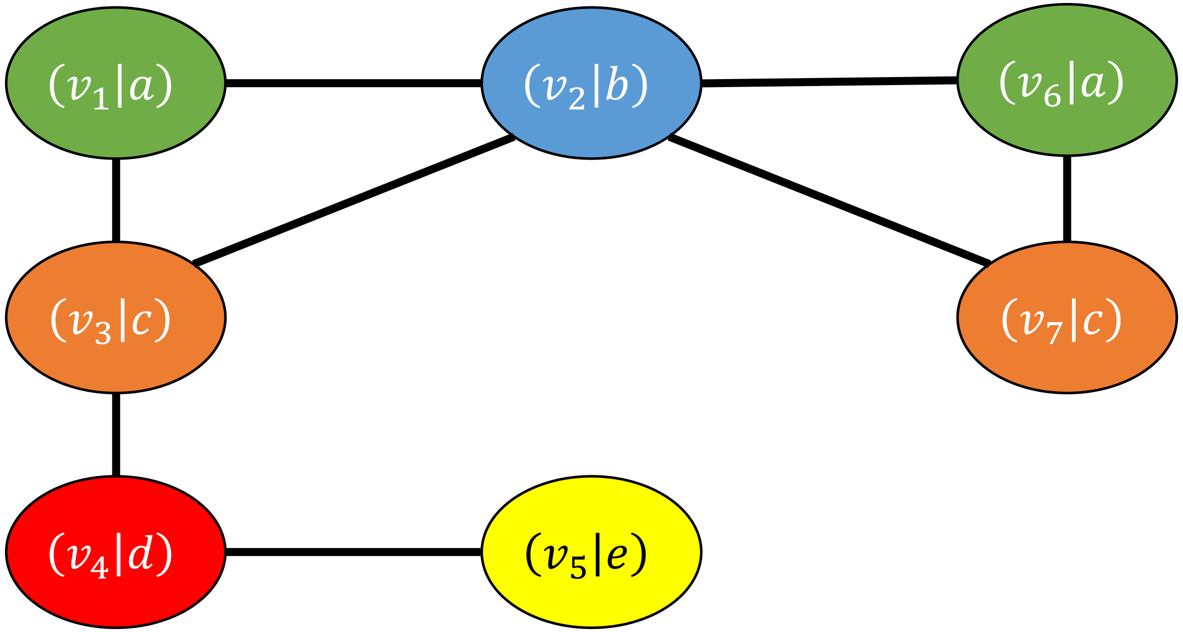
# 1.1 Weisfeiler-Lehman node embedding scheme
# Example – Node labels

$m = 1$: dimensionality of the node labels

$n_G = 7$: cardinality of the vertex set

$N(v)$: neighbourhood of $v$

$$l^{h+1}(v) = hash(l^h(v), N^h(v))$$



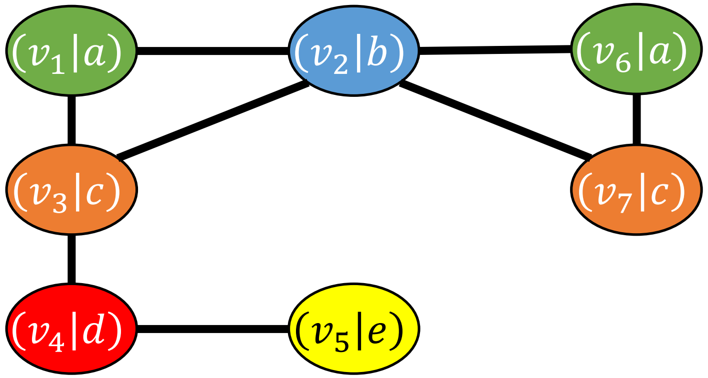
# 1.1 Weisfeiler-Lehman node embedding scheme
# Example – Node labels

$m = 1$: dimensionality of the node labels

$n_G = 7$: cardinality of the vertex set

$N(v)$: neighbourhood of $v$

$$l^{h+1}(v) = hash(l^h(v), N^h(v))$$

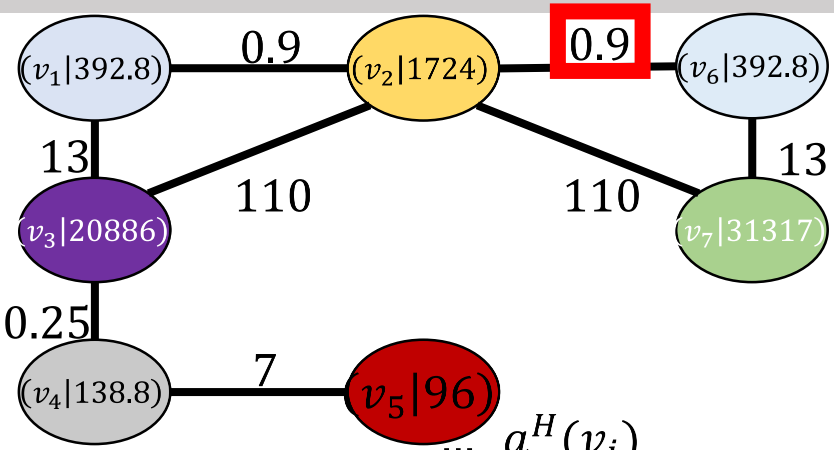



| $i$ | $l^0(v_i)$ | $l^1(v_i)$ | $l^2(v_i)$ | ... | $l^H(v_i)$ |
|---|---|---|---|---|---|
| 1 | $a$ | hash$(a, \{b, c\}) = x$ | hash$(x, \{q, p\}) = k$ | ... | ... |
| 2 | $b$ | hash$(b, \{a, c\}) = q$ | hash$(q, \{x, p\}) = o$ | ... | ... |
| 3 | $c$ | hash$(c, \{a, b, d\}) = p$ | hash$(p, \{x, q, h\}) = r$ | ... | ... |
| 4 | $d$ | hash$(d, \{c, e\}) = h$ | hash$(h, \{p, j\}) = w$ | ... | ... |
| 5 | $e$ | hash$(e, \{d\}) = j$ | hash$(j, \{h\}) = f$ | ... | ... |
| 6 | $a$ | hash$(a, \{b, c\}) = x$ | hash$(x, \{q, r\}) = g$ | ... | ... |
| 7 | $c$ | hash$(c, \{a, b\}) = r$ | hash$(r, \{x, q\}) = s$ | ... | ... |
| $f_{WL}^H : G \longrightarrow ($ | $\boldsymbol{X_G^0}$ | $\boldsymbol{X_G^1}$ | $\boldsymbol{X_G^2}$ | ... | $\boldsymbol{X_G^H}$ $)$ |

# 1.1 Weisfeiler-Lehman node embedding scheme
# Example – Attribute labels

UNIVERSITÄT BONN

$$a^{h+1}(v) = \frac{1}{2}\left( a^h(v) + \frac{1}{\deg(v)} \sum_{u \in N(v)} w((u,v))\, a^h(u) \right)$$




| $i$ | $a^0(v_i)$ | $a^1(v_i)$ | ... | $a^H(v_i)$ |
|---|---|---|---|---|
| 1 | 0.5 | $^1/_2\,(0.5 + {^1/_2}\,[0.9 * 1138 + 13 * 42]) = 392.8$ | ... | ... |
| 2 | 42 | $^1/_2\,(42 + {^1/_3}\,[13 * 0.5 + 110 * 1138 + 0.25 * 17]) = 20886.125$ | ... | ... |
| 3 | 1138 | $^1/_2\,(1138 + {^1/_4}\,[0.9 * 0.5 + 110 * 42 + \boxed{0.9} * 0.5 + 110 * 42]) = 1724.1125$ | ... | ... |
| 4 | 17 | $^1/_2\,(17 + {^1/_2}\,[0.25 * 42 + 7 * 73]) = 138.875$ | ... | ... |
| 5 | 73 | $^1/_2\,(73 + {^1/_1}\,[7 * 17]) = 96$ | ... | ... |
| 6 | 0.5 | $^1/_2\,(0.5 + {^1/_2}\,[\boxed{0.9} * 1138 + 13 * 42]) = 392.8$ | ... | ... |
| 7 | 42 | $^1/_2\,(42 + {^1/_2}\,[13 * 0.5 + 110 * 1138]) = 31317.625$ | ... | ... |

# 2.1 Ground disance between nodes

$$\rho(x, y) = \begin{cases} 1 \text{ if } x \neq y \\ 0 \text{ if } x = y \end{cases}$$

| NODE LABELS | NODE ATTRIBUTES |
|---|---|
| Normalised Hamming distance: $d_{Ham}(v, v') = \frac{1}{H+1} \sum_{i=1}^{H+1} \rho(v_i, v'_i)$ | Euclidean distance: $d_E(v, v') = \lVert v - v' \rVert_2$ |

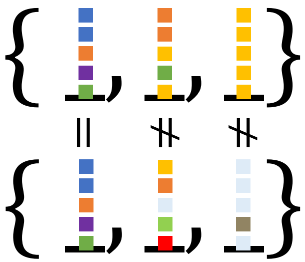
$\neq$ 1
$\neq$ 2
$=$
$\neq$ 3 $\Longrightarrow d_{Ham} = 3$
$=$

# 2.1 Ground disance between nodes

UNIVERSITÄT BONN

$$\rho(x, y) = \begin{cases} 1 \text{ if } x \neq y \\ 0 \text{ if } x = y \end{cases}$$

| NODE LABELS | NODE ATTRIBUTES |
|---|---|
| Normalised Hamming distance: $$d_{Ham}(v, v') = \frac{1}{H+1} \sum_{i=1}^{H+1} \rho(v_i, {v'}_i)$$ | Euclidean distance: $$d_E(v, v') = \lVert v - v' \rVert_2$$ |



1 2

$$\Longrightarrow d_{Ham} = 2$$

# 2.2 Wasserstein distance - $L^p$

$p \in [1, \infty)$: parameter for the $\boldsymbol{L^p}$ Wasserstein distance

$\sigma, \mu$: probability distributions on a metric space $\boldsymbol{M}$

$d: \mathbb{R}^m x \mathbb{R}^m \to \mathbb{R}^m$: ground distance (on $M$)

$\Gamma(\sigma, \mu)$: set of all transportaion plans over MxM (with marginals $\sigma$ and $\mu$ on the first and second factor)

$$W_p(\sigma, \mu) := \left( \lim_{\gamma \in \Gamma(\sigma,\mu)} \int_{MxM} d(x,y)^p d\gamma(x,y) \right)^{\frac{1}{p}}$$

# 2.2 Graph Wasserstein distance

UNIVERSITÄT BONN

$p \in [1, \infty)$: parameter for the $\boldsymbol{L^p}$ Wasserstein distance

$X \in \mathbb{R}^{n \times m}$

$X' \in \mathbb{R}^{n' \times m}$: set of vectors

$M = [d(x, y)]_{x \in X, y \in X'}$: distance matrix on a ground distance

$\Gamma(X, X')$: set of all transportaion plans

$\langle ., . \rangle$: Frobenius dot product

$$W_1(X, X') := \min_{P \in \Gamma(X, X')} \langle P, M \rangle$$

$$G, G' \quad \rightarrowtail f_{WL}^H : G \to \mathbb{R}^{n_G \times m(H+1)} \quad \rightarrowtail \quad D_W^f(G, G') := W_1(f(G), f(G')) = \min_{P \in \Gamma(f(G), f(G'))} \langle P, M \rangle$$

# 3 Kernel definition

$$G, G' \quad \rightarrowtail \quad f_{WL}^{H}: G \rightarrow \mathbb{R}^{n_G \times m(H+1)} \quad \rightarrowtail \quad D_W^f(G, G') := W_1(f(G), f(G')) = \min_{P \in \Gamma(f(G), f(G'))} \langle P, M \rangle \quad \rightarrowtail \quad K_{WWL} = e^{-\lambda D_W^{f_{WL}}(G, G')}$$

# Experimental evaluation

UNIVERSITÄT BONN

| NODE LABELS | NODE ATTRIBUTES |
|---|---|
| • Competitive with best graph kernel | • Outperforms all state-of-the art graph kernels |

TODO: summarize tables in pile chart with slack

# Sumary

TODO: the whole method only in 3-4 graphics
(graphics from above)

# Thank you

# for your attention!