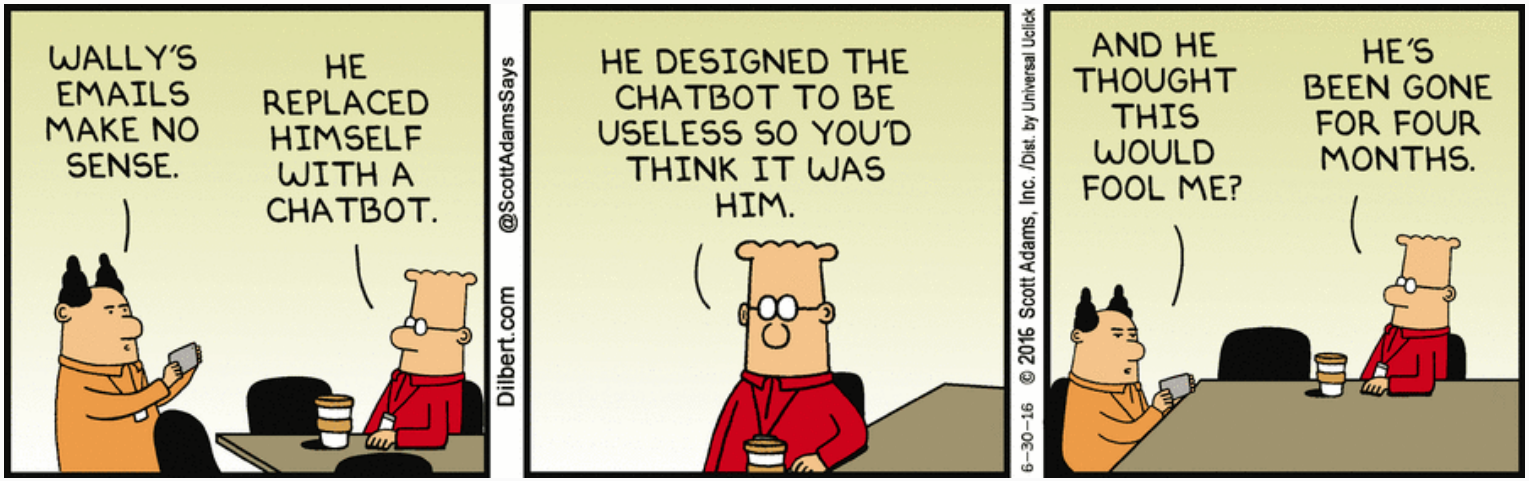
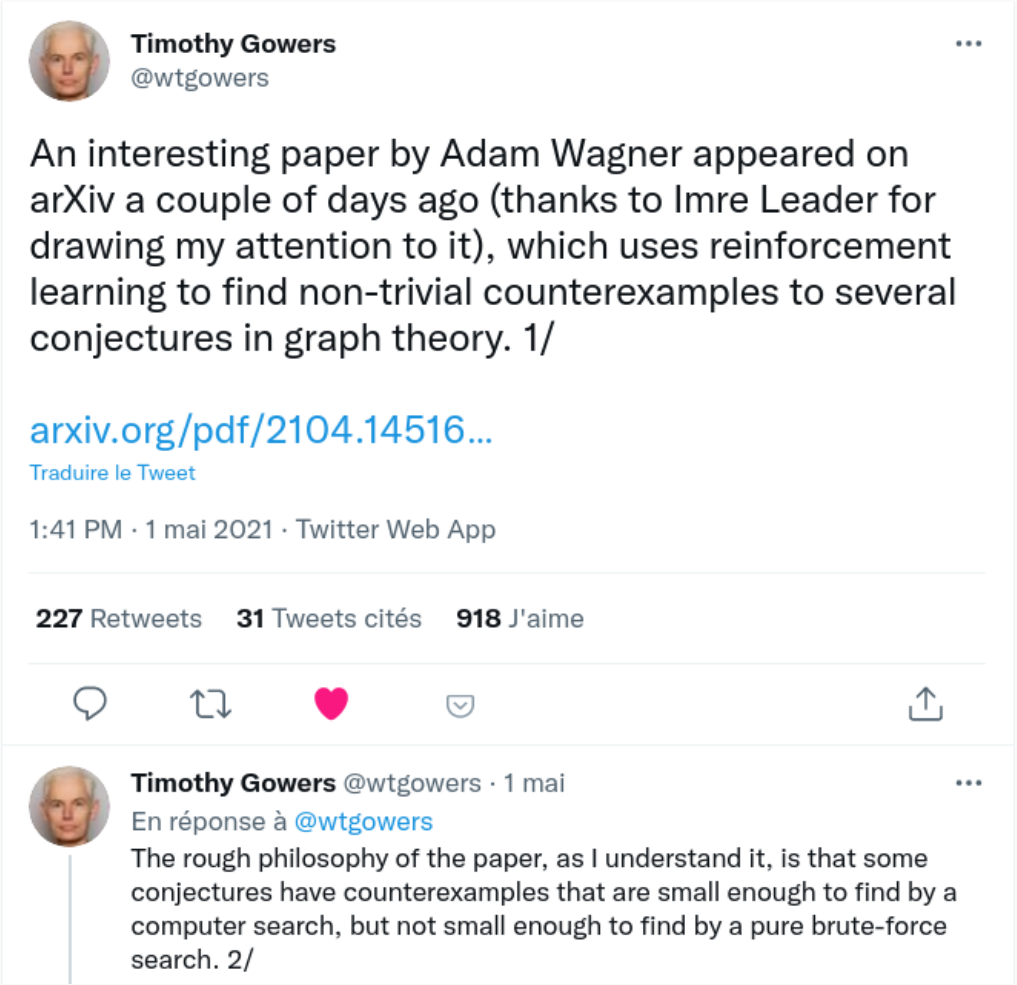
Deep Learning with Symmetries

Marc Lelarge
ENS & INRIA
marc.lelarge@ens.fr

AI in combinatorics ?



AI in combinatorics ?



Timothy Gowers
@wtgowers

An interesting paper by Adam Wagner appeared on arXiv a couple of days ago (thanks to Imre Leader for drawing my attention to it), which uses reinforcement learning to find non-trivial counterexamples to several conjectures in graph theory. 1/

arxiv.org/pdf/2104.14516...

Traduire le Tweet

1:41 PM · 1 mai 2021 · Twitter Web App

227 Retweets **31** Tweets cités **918** J'aime

Timothy Gowers @wtgowers · 1 mai

En réponse à @wtgowers

The rough philosophy of the paper, as I understand it, is that some conjectures have counterexamples that are small enough to find by a computer search, but not small enough to find by a pure brute-force search. 2/

Apetizer : convolutions as equivariant layers

Convolutions from first principles

Apetizer : convolutions as equivariant layers

Convolutions from first principles

From an arbitrary function f, an easy way to construct an invariant version :

$$\frac{1}{|T|} \sum_{t \in T} f(T(\mathcal{I}))$$

Apetizer : convolutions as equivariant layers

Convolutions from first principles

From an arbitrary function f, an easy way to construct an invariant version :

$$\frac{1}{|T|}\sum_{t\in T} f(T(\mathcal{I}))$$

In practice, data augmentation :

so that the learned function f_θ is such that

$$f_\theta(T(\mathcal{I})) \approx f_\theta(\mathcal{I})$$

Learning with graph symmetries

Why symmetries matter in machine learning ?

Start with a linear regression : your task is to estimate a linear model $\beta_1 x_1 + \cdots + \beta_n x_n$ from noisy observations $(\mathbf{x}, y)$.

Q : How many parameters do you need to estimate if you know in addition that the model is invariant to permutation of the input $(x_1, \ldots, x_n)$?

Why symmetries matter in machine learning ?

Start with a linear regression : your task is to estimate a linear model $\beta_1 x_1 + \cdots + \beta_n x_n$ from noisy observations $(\mathbf{x}, y)$.

Q : How many parameters do you need to estimate if you know in addition that the model is invariant to permutation of the input $(x_1, \ldots, x_n)$?

A : there is only one parameter to estimate because invariance implies $\beta_1 = \cdots = \beta_n$.

Why symmetries matter in machine learning ?

Start with a linear regression : your task is to estimate a linear model $\beta_1 x_1 + \cdots + \beta_n x_n$ from noisy observations $(\mathbf{x}, y)$.

Q : How many parameters do you need to estimate if you know in addition that the model is invariant to permutation of the input $(x_1, \ldots, x_n)$?

A : there is only one parameter to estimate because invariance implies $\beta_1 = \cdots = \beta_n$.

Q : a linear regression on graphs : estimate a linear function of the adjacency matrix in $\mathbb{R}^{n \times n}$, how many parameters to estimate ?

Why symmetries matter in machine learning?

Start with a linear regression : your task is to estimate a linear model $\beta_1 x_1 + \cdots + \beta_n x_n$ from noisy observations $(\mathbf{x}, y)$.

Q : How many parameters do you need to estimate if you know in addition that the model is invariant to permutation of the input $(x_1, \ldots, x_n)$?

A : there is only one parameter to estimate because invariance implies $\beta_1 = \cdots = \beta_n$.

Q : a linear regression on graphs : estimate a linear function of the adjacency matrix in $\mathbb{R}^{n \times n}$, how many parameters to estimate ?

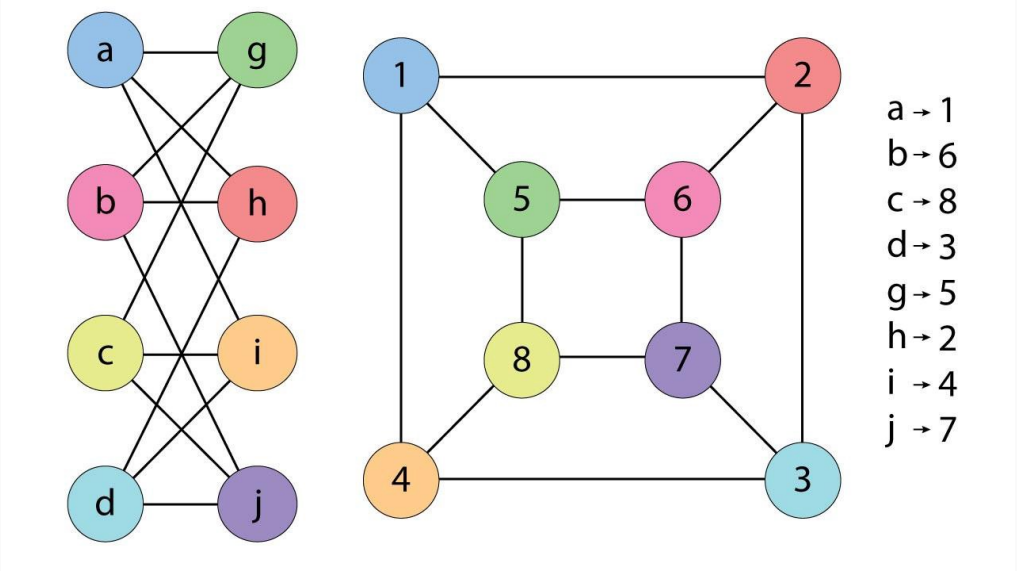
A : there are only two parameters to estimate for a linear function $f : \mathbb{R}^{n \times n} \to \mathbb{R}$ invariant to permutation of the rows and columns :

$$f(A) = \alpha \sum_{i=j} A_{i,j} + \beta \sum_{i \neq j} A_{i,j},$$

whatever the value of n !

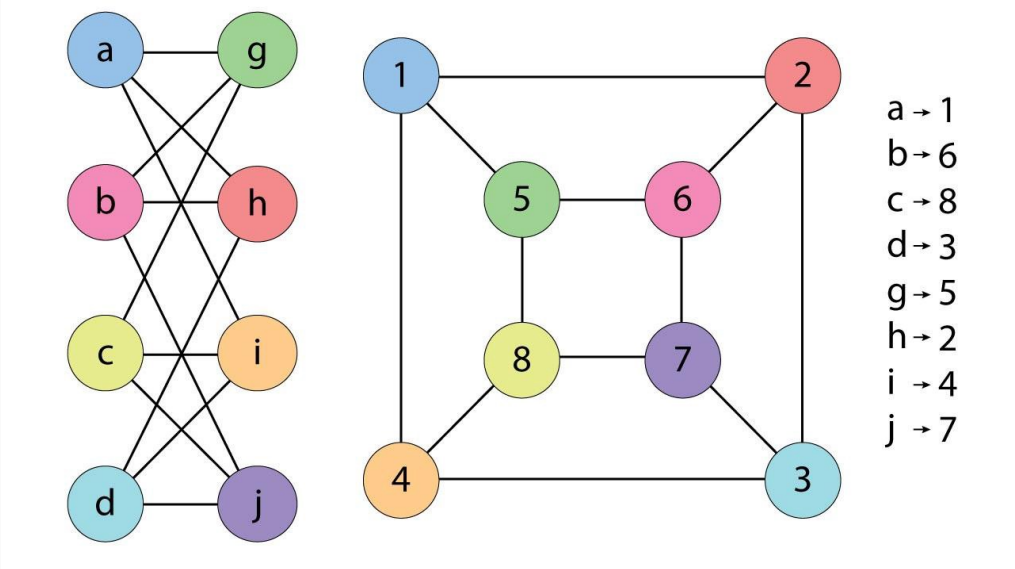
Graph isomorphism

$G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a bijection $V_1 \longrightarrow V_2$ which preserves edges.



Graph isomorphism

$G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a bijection $V_1 \longrightarrow V_2$ which preserves edges.

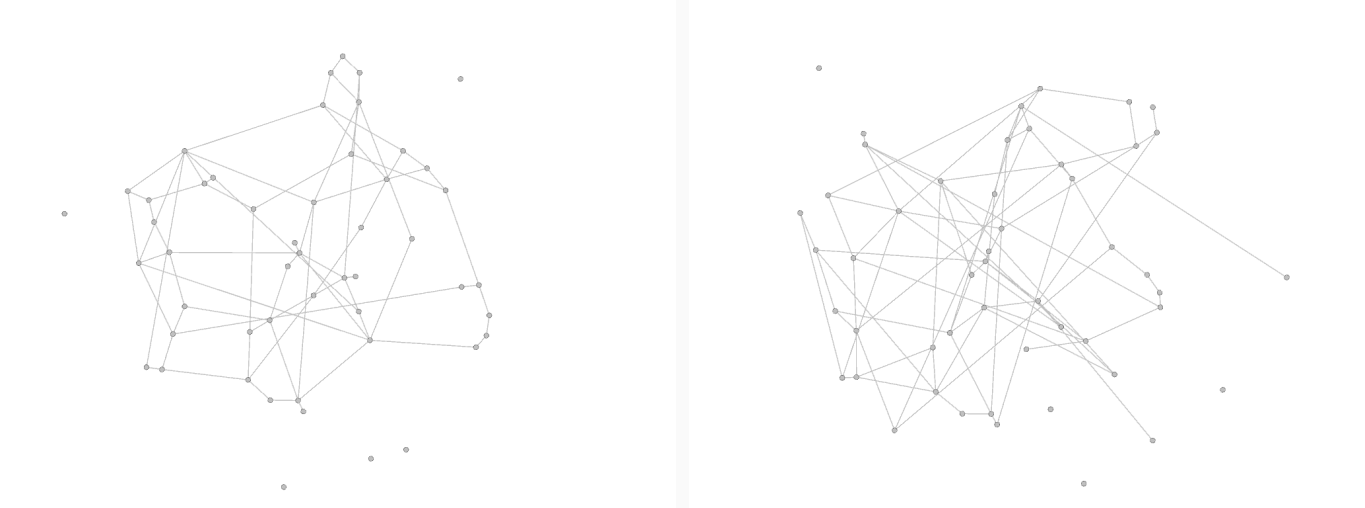


Idea : design a machine learning algorithm whose result does not depend on the representation of the input.

Alignment of Graphs

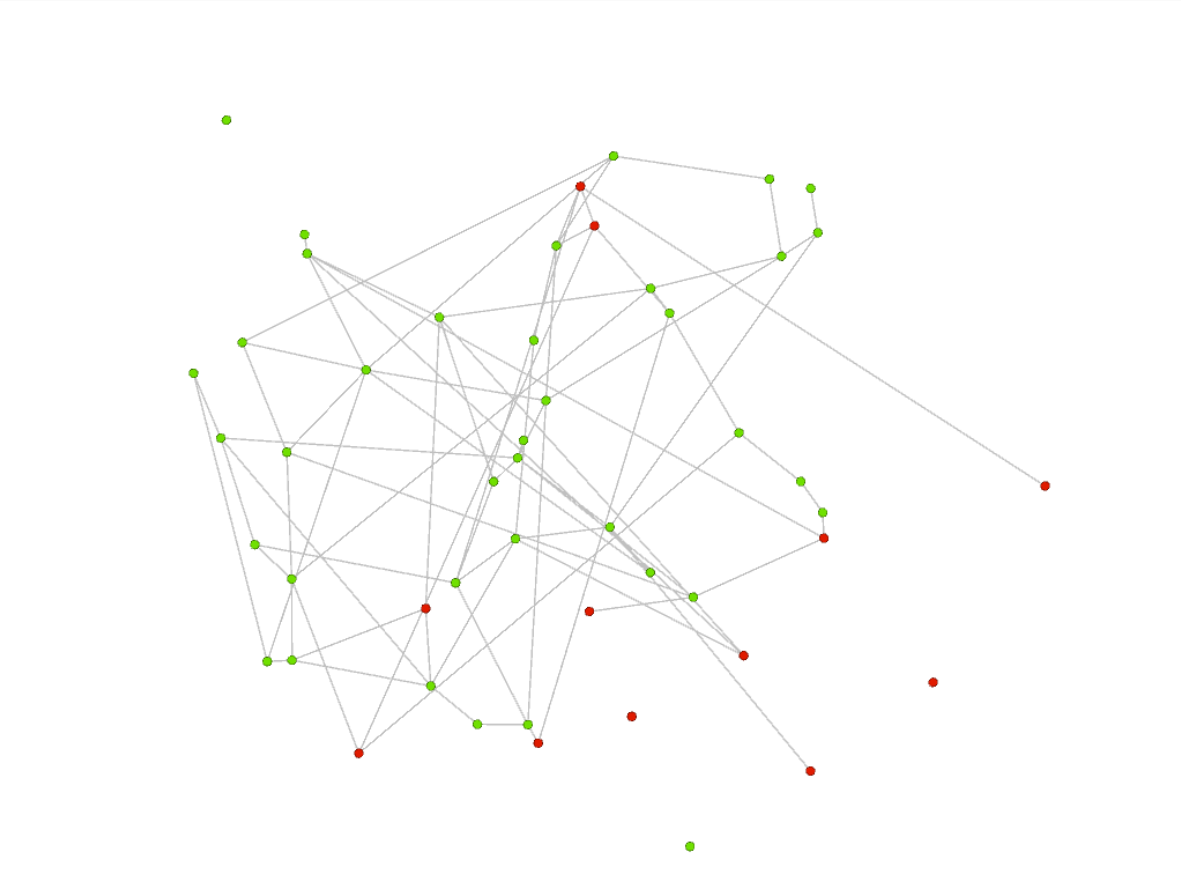
Problem : alignment of graphs

These 2 graphs are noisy versions of an original graph : can you align the vertices ?



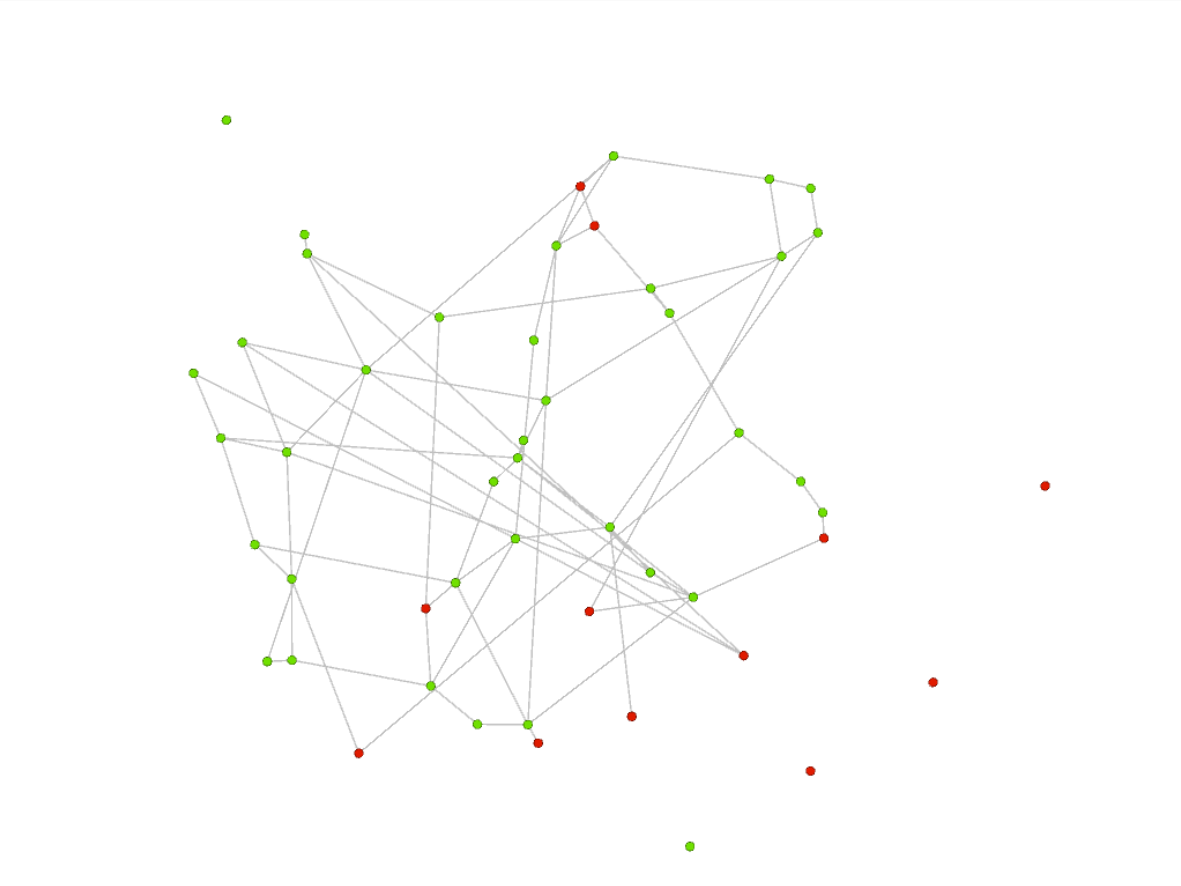
Result with our GNN

Green vertices are well paired vertices. Red vertices are errors (graph 1).



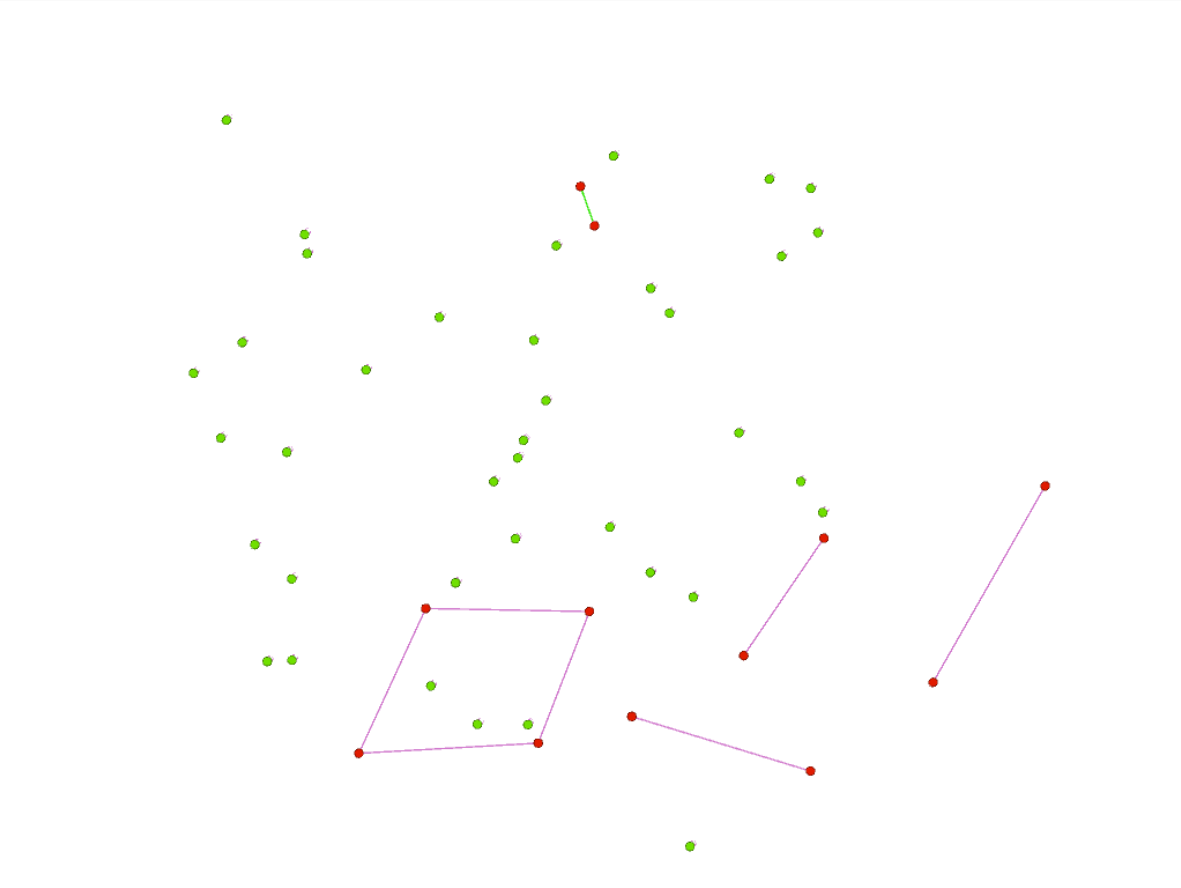
Result with our GNN

Green vertices are well paired vertices. Red vertices are errors (graph 2).



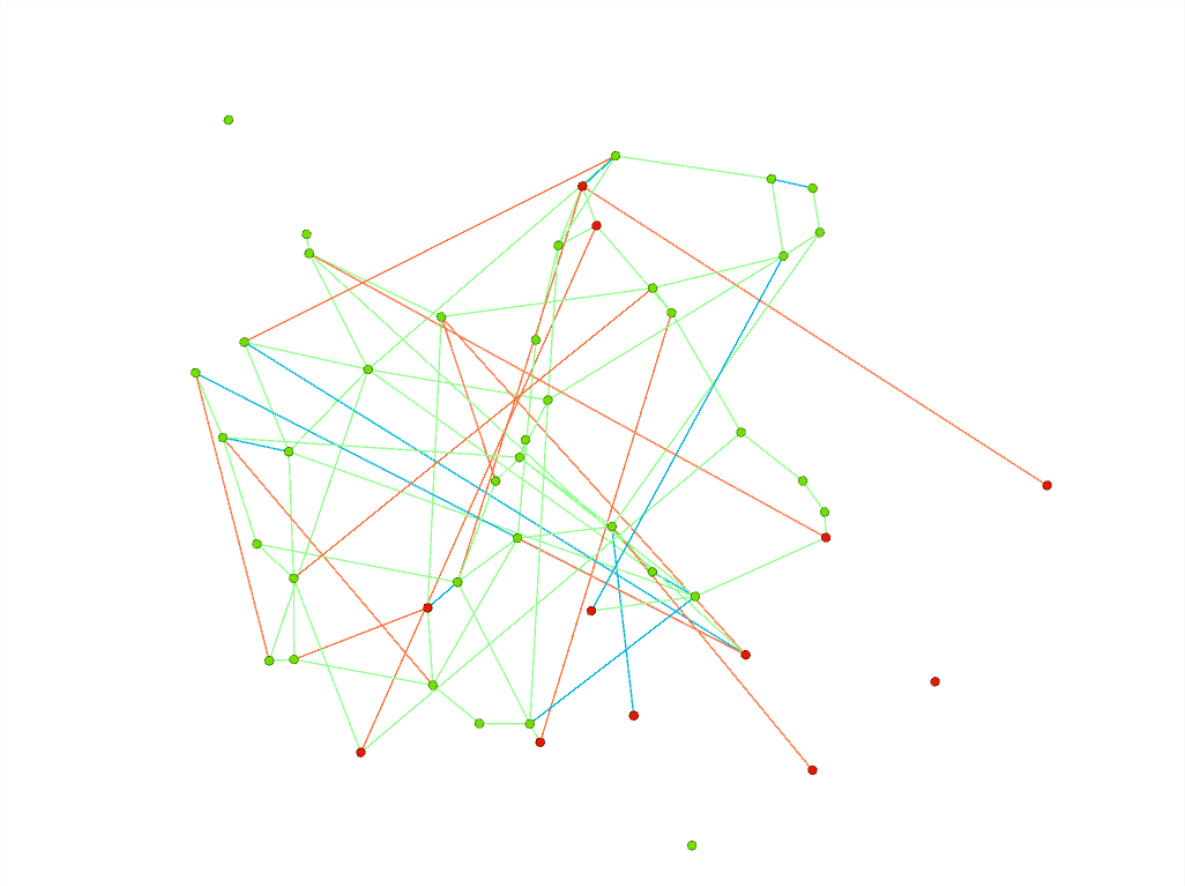
Result with our GNN

Here are the ’wrong’ matchings.



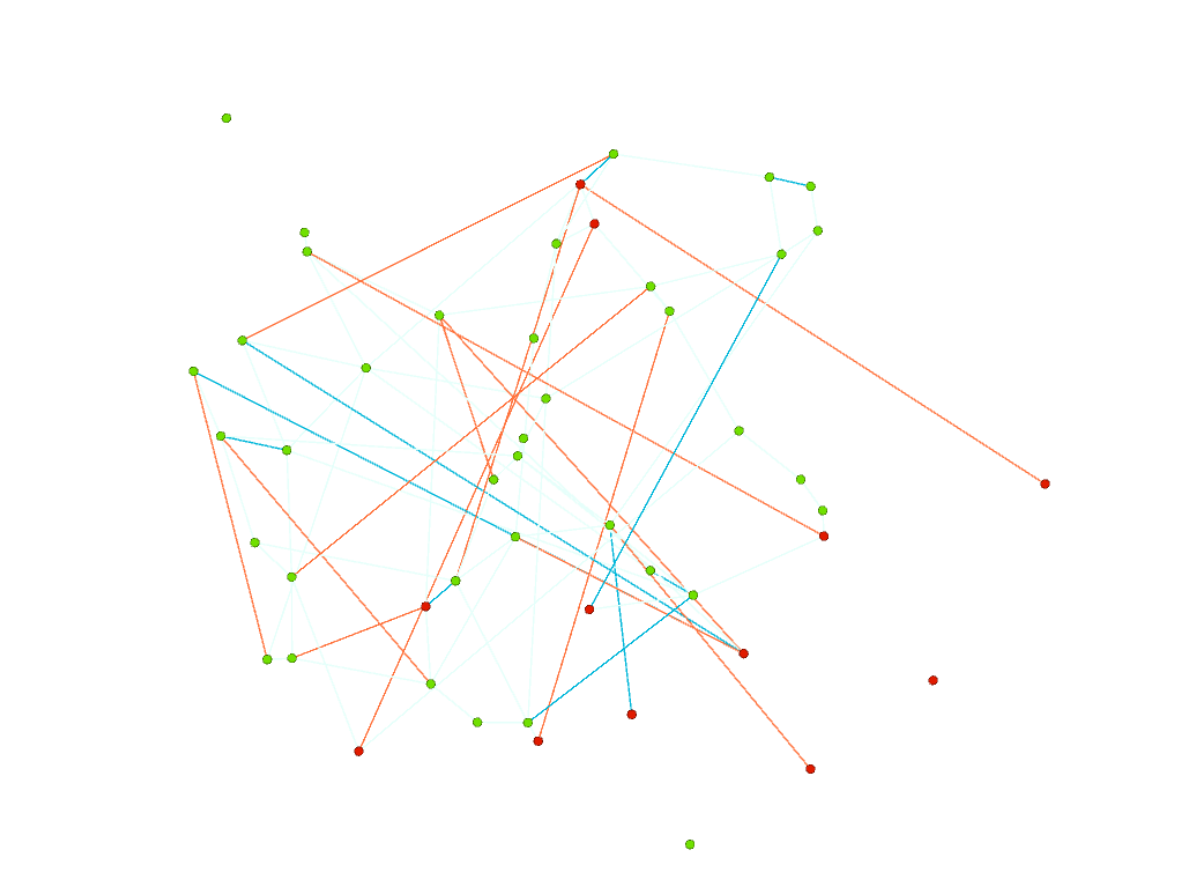
Result with our GNN

Superposing the 2 graphs : green edges in both, orange and blue edges in graph 1 and 2 resp.



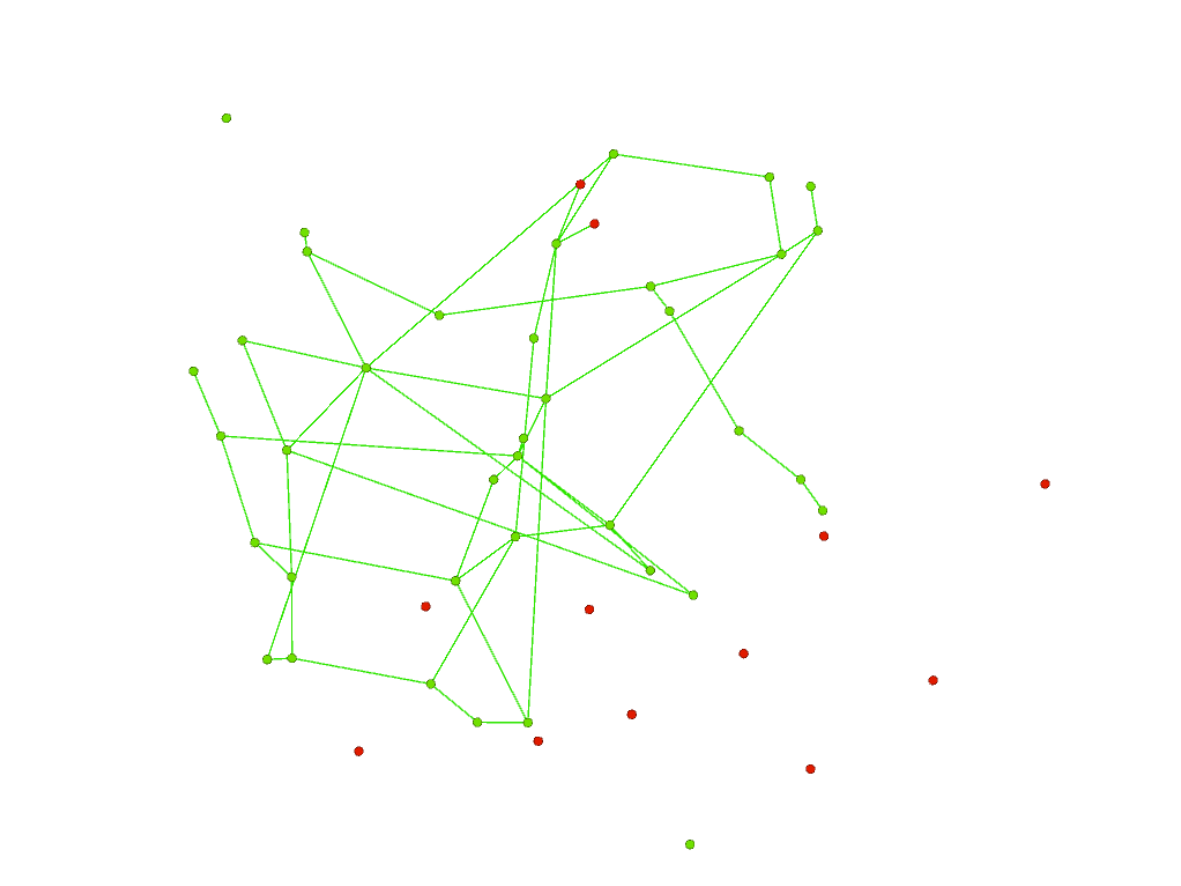
Result with our GNN

Green vertices are well paired vertices. Red vertices are errors.



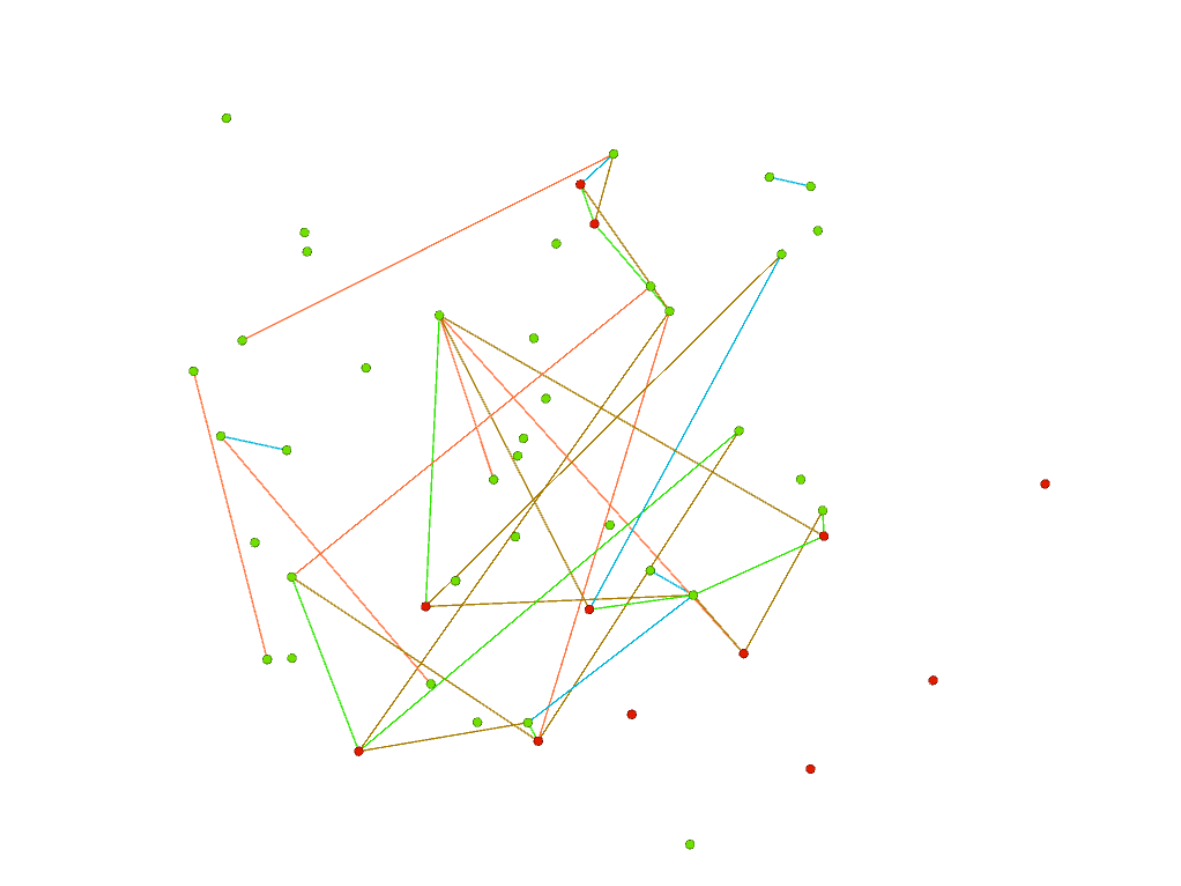
Result with our GNN

Matched edges.



Result with our GNN

Mismatched edges.



Invariant and Equivariant GNNs

Invariant and equivariant functions

For a permutation $\sigma \in \mathcal{S}_n$, we define ($\mathbb{F} = \mathbb{R}^p$ feature space) :

- for $X \in \mathbb{F}^n$, $(\sigma \star X)_{\sigma(i)} = X_i$
- for $G \in \mathbb{F}^{n \times n}$, $(\sigma \star G)_{\sigma(i_1),\sigma(i_2)} = G_{i_1,i_2}$

Invariant and equivariant functions

For a permutation $\sigma \in \mathcal{S}_n$, we define ($\mathbb{F} = \mathbb{R}^p$ feature space) :

- for $X \in \mathbb{F}^n$, $(\sigma \star X)_{\sigma(i)} = X_i$
- for $G \in \mathbb{F}^{n \times n}$, $(\sigma \star G)_{\sigma(i_1),\sigma(i_2)} = G_{i_1,i_2}$

G_1, G_2 are isomorphic iff $G_1 = \sigma \star G_2$.

Invariant and equivariant functions

For a permutation $\sigma \in \mathcal{S}_n$, we define ($\mathbb{F} = \mathbb{R}^p$ feature space) :

- for $X \in \mathbb{F}^n$, $(\sigma \star X)_{\sigma(i)} = X_i$
- for $G \in \mathbb{F}^{n \times n}$, $(\sigma \star G)_{\sigma(i_1),\sigma(i_2)} = G_{i_1,i_2}$

G_1, G_2 are isomorphic iff $G_1 = \sigma \star G_2$.

Definition
($k = 1$ or $k = 2$)

A function $f : \mathbb{F}^{n^k} \to \mathbb{F}$ is said to be invariant if $f(\sigma \star G) = f(G)$.

A function $f : \mathbb{F}^{n^k} \to \mathbb{F}^n$ is said to be equivariant if $f(\sigma \star G) = \sigma \star f(G)$.

Invariant and equivariant functions

For a permutation $\sigma \in \mathcal{S}_n$, we define ($\mathbb{F} = \mathbb{R}^p$ feature space) :

- for $X \in \mathbb{F}^n$, $(\sigma \star X)_{\sigma(i)} = X_i$
- for $G \in \mathbb{F}^{n \times n}$, $(\sigma \star G)_{\sigma(i_1),\sigma(i_2)} = G_{i_1,i_2}$

G_1, G_2 are isomorphic iff $G_1 = \sigma \star G_2$.

Definition
($k = 1$ or $k = 2$)

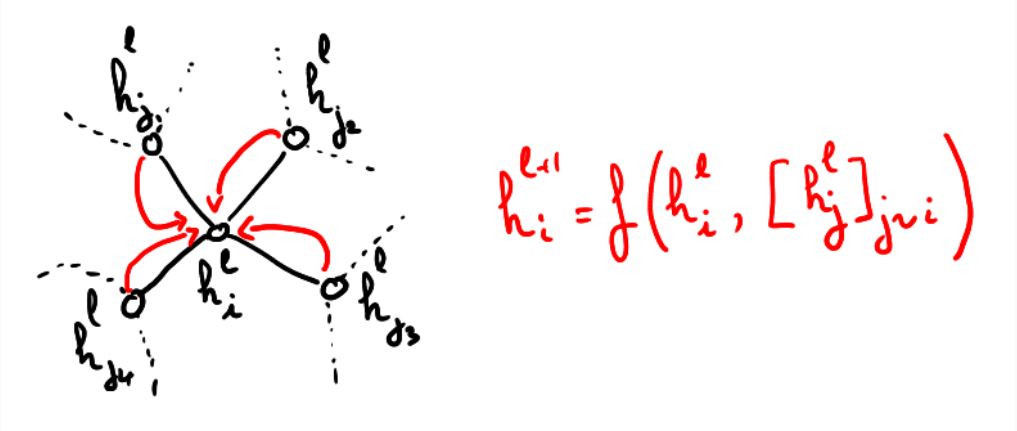
A function $f : \mathbb{F}^{n^k} \to \mathbb{F}$ is said to be invariant if $f(\sigma \star G) = f(G)$.

A function $f : \mathbb{F}^{n^k} \to \mathbb{F}^n$ is said to be equivariant if $f(\sigma \star G) = \sigma \star f(G)$.

For the graph alignment problem, we used an equivariant GNN from $\{0,1\}^{n \times n}$ to $\mathbb{F}^n$.

Practical GNNs are not universal

A first example : Message passing GNN (MGNN)

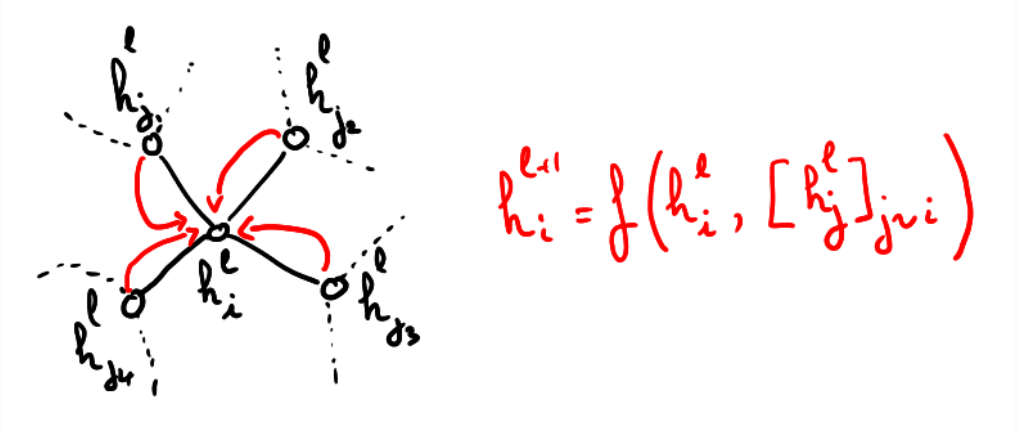


MGNN takes as input a discrete graph $G = (V, E)$ with n nodes and are defined inductively as : $h_i^\ell \in \mathbb{F}$ being the features at layer ℓ associated with node i, then

$$h_i^{\ell+1} = f\left(h_i^\ell, \left\{\left\{h_j^\ell\right\}\right\}_{j\sim i}\right) = f_0\left(h_i^\ell, \sum_{j\sim i} f_1\left(h_i^\ell, h_j^\ell\right)\right),$$

where f or f_0 and f_1 are learnable functions.

A first example : Message passing GNN (MGNN)



MGNN takes as input a discrete graph $G = (V, E)$ with n nodes and are defined inductively as : $h_i^\ell \in \mathbb{F}$ being the features at layer ℓ associated with node i, then

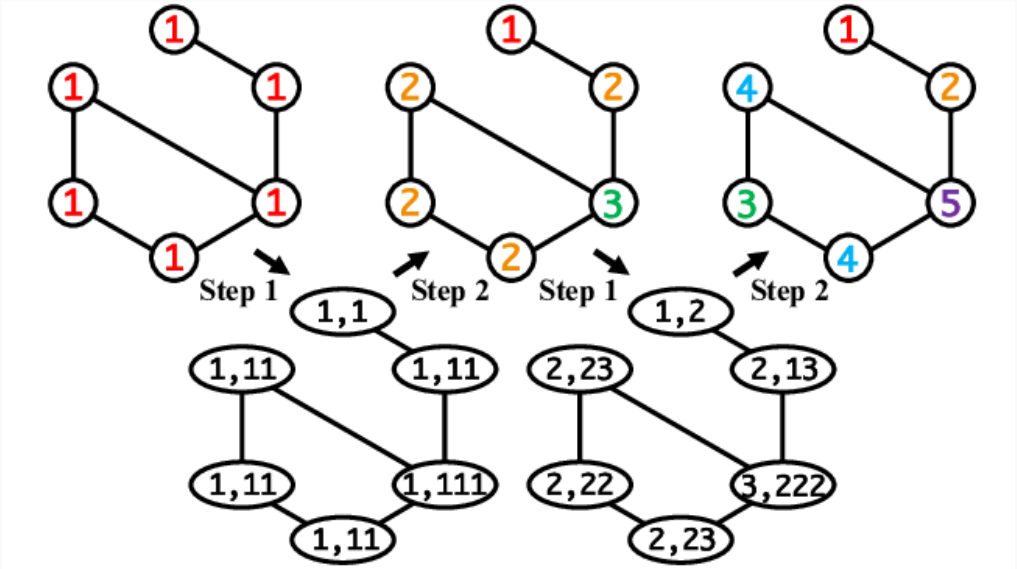
$$h_i^{\ell+1} = f\left(h_i^\ell, \left\{\left\{h_j^\ell\right\}\right\}_{j\sim i}\right) = f_0\left(h_i^\ell, \sum_{j\sim i} f_1\left(h_i^\ell, h_j^\ell\right)\right),$$

where f or f_0 and f_1 are learnable functions.

Prop : The message passing layer is equivariant and both expressions above are equivalent (i.e. for each f, there exists f_0 and f_1).

Xu et al. (2019) : MGNN are as powerful as 2-Weisfeiler-Lehman

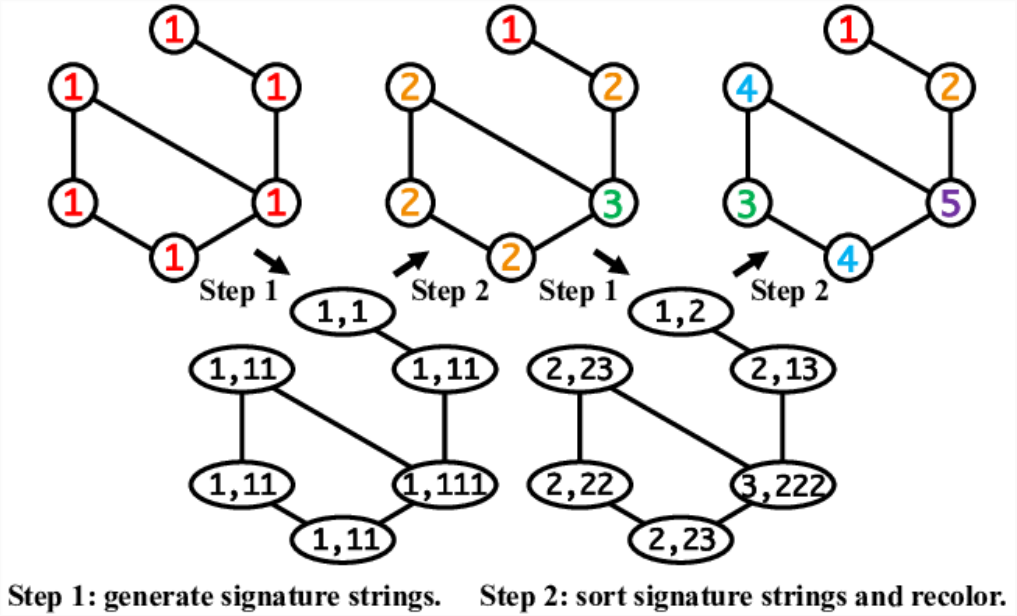
For $k \geq 2$, k-WL(G) are invariants based on the Weisfeiler-Lehman tests designed for the graph isomorphism problem.



Step 1: generate signature strings. Step 2: sort signature strings and recolor.

Xu et al. (2019) : MGNN are as powerful as 2-Weisfeiler-Lehman

For $k \geq 2$, k-WL(G) are invariants based on the Weisfeiler-Lehman tests designed for the graph isomorphism problem.



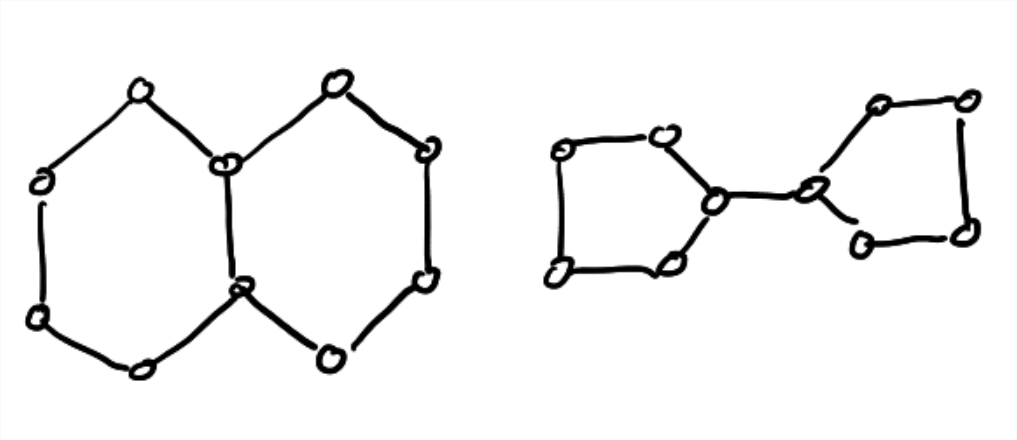
Step 1: generate signature strings. Step 2: sort signature strings and recolor.

Cor : MGNN are useless on d-regular graphs.

MGNN are not universal

Prop : MGNN are useless on d-regular graphs (without features).

Another example of a problematic pair for MGNN :

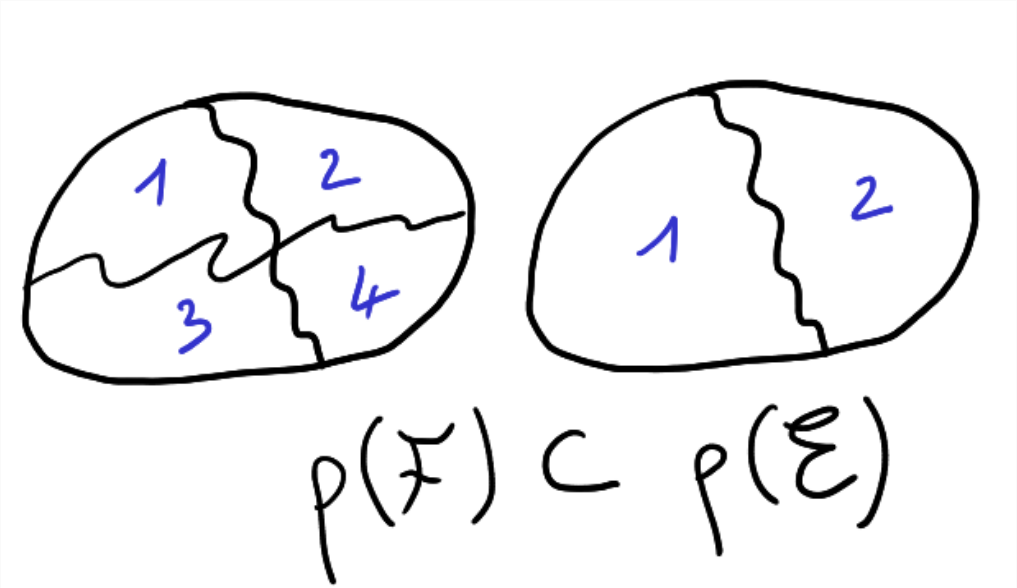


Separating power of MGNN

Separation : Let $\mathcal{F}$ be a set of functions f defined on a set X. The equivalence relation $\rho(\mathcal{F})$ defined by $\mathcal{F}$ on X is : for any $x, x' \in X$,

$$(x, x') \in \rho(\mathcal{F}) \iff \forall f \in \mathcal{F},\ f(x) = f(x')\,.$$

Given two sets of functions $\mathcal{F}$ and $\mathcal{E}$, we say that $\mathcal{F}$ is more separating than $\mathcal{E}$ if $\rho(\mathcal{F}) \subset \rho(\mathcal{E})$.

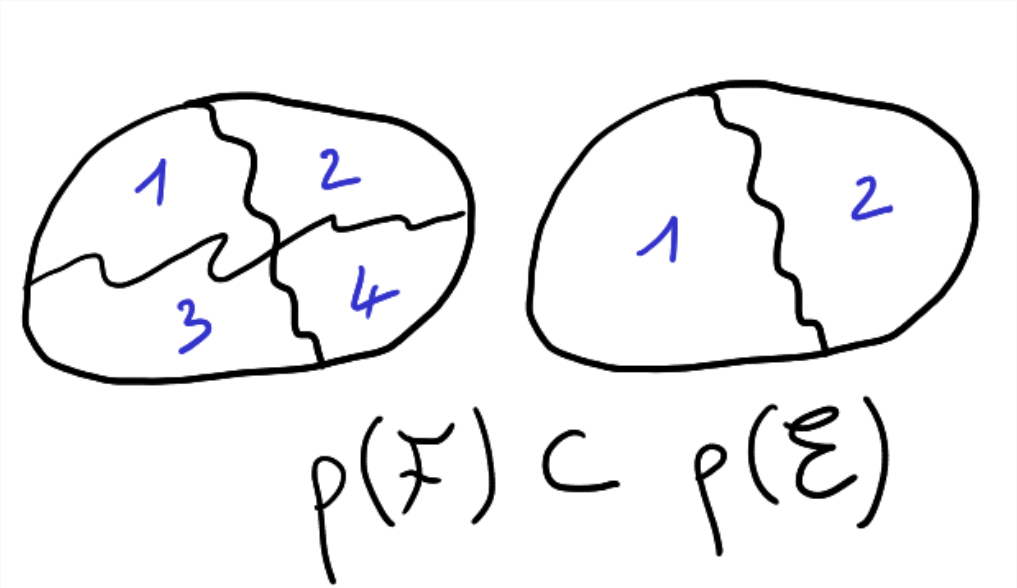


Separating power of MGNN

Separation : Let $\mathcal{F}$ be a set of functions f defined on a set X. The equivalence relation $\rho(\mathcal{F})$ defined by $\mathcal{F}$ on X is : for any $x, x' \in X$,

$$(x, x') \in \rho(\mathcal{F}) \iff \forall f \in \mathcal{F},\ f(x) = f(x')\,.$$

Given two sets of functions $\mathcal{F}$ and $\mathcal{E}$, we say that $\mathcal{F}$ is more separating than $\mathcal{E}$ if $\rho(\mathcal{F}) \subset \rho(\mathcal{E})$.



Xu et al. (2019) Prop : $\rho(\text{MGNN}) = \rho(2\text{-WL})$

Waïss Azizian (ICLR21) : from Separation to Approximation

Stone-Weierstrass theorem

An easy general fact :

If there exists $x \neq x'$ with $(x, x') \in \rho(\mathcal{F})$, all functions in $\mathcal{F}$ take the same values at x and x' and $\mathcal{F}$ cannot be dense.

Approximation $\Rightarrow$ Separation

Stone-Weierstrass theorem

An easy general fact :

If there exists $x \neq x'$ with $(x, x') \in \rho(\mathcal{F})$, all functions in $\mathcal{F}$ take the same values at x and x' and $\mathcal{F}$ cannot be dense.

Approximation $\Rightarrow$ Separation

If $\mathcal{F}$ is an algebra containing the constant function $\mathbf{1}$, i.e. vector space closed under pointwise multiplication then : **Separation $\Leftrightarrow$ Approximation**.

Stone-Weierstrass theorem

An easy general fact :

If there exists $x \neq x'$ with $(x, x') \in \rho(\mathcal{F})$, all functions in $\mathcal{F}$ take the same values at x and x' and $\mathcal{F}$ cannot be dense.

Approximation $\Rightarrow$ Separation

If $\mathcal{F}$ is an algebra containing the constant function $\mathbf{1}$, i.e. vector space closed under pointwise multiplication then : **Separation $\Leftrightarrow$ Approximation**.

Pb : we know MGNNs do not separate all graphs !

Stone-Weierstrass theorem

An easy general fact :

If there exists $x \neq x'$ with $(x, x') \in \rho(\mathcal{F})$, all functions in $\mathcal{F}$ take the same values at x and x' and $\mathcal{F}$ cannot be dense.

Approximation $\Rightarrow$ Separation

If $\mathcal{F}$ is an algebra containing the constant function $\mathbf{1}$, i.e. vector space closed under pointwise multiplication then : **Separation $\Leftrightarrow$ Approximation**.

Pb : we know MGNNs do not separate all graphs !

Sol : we need to relax the separation assumption... and consider vector-valued functions

Stone-Weierstrass theorem for vector-valued functions with symmetries

Building on the work of Timofte (2005), and we proved :

Theorem
Let $\mathcal{F} \subset \mathcal{C}_I(X, \mathbb{R}^p)$ be a sub-algebra of continuous invariant functions, (...).

If the set of functions $\mathcal{F}_{scal} \subset \mathcal{C}(X, \mathbb{R})$ defined by,

$$\mathcal{F}_{scal} = \{f \in \mathcal{C}(X, \mathbb{R}) : f\mathbf{1} \in \mathcal{F}\}$$

is more separating than $\mathcal{F}$, i.e. satisfies,

$$\rho(\mathcal{F}_{scal}) \subset \rho(\mathcal{F})\,.$$

Then any function less separating than $\mathcal{F}$ can be approximated, i.e.

$$\overline{\mathcal{F}} = \left\{f \in \mathcal{C}_I(X, \mathbb{R}^p) : \rho(\mathcal{F}) \subset \rho(f)\right\}\,.$$

Stone-Weierstrass theorem for vector-valued functions with symmetries

Building on the work of Timofte (2005), and we proved :

Theorem
Let $\mathcal{F} \subset \mathcal{C}_I(X, \mathbb{R}^p)$ be a sub-algebra of continuous invariant functions, (...).

If the set of functions $\mathcal{F}_{scal} \subset \mathcal{C}(X, \mathbb{R})$ defined by,

$$\mathcal{F}_{scal} = \{f \in \mathcal{C}(X, \mathbb{R}) : f\mathbf{1} \in \mathcal{F}\}$$

is more separating than $\mathcal{F}$, i.e. satisfies,

$$\rho(\mathcal{F}_{scal}) \subset \rho(\mathcal{F})\,.$$

Then any function less separating than $\mathcal{F}$ can be approximated, i.e.

$$\overline{\mathcal{F}} = \left\{f \in \mathcal{C}_I(X, \mathbb{R}^p) : \rho(\mathcal{F}) \subset \rho(f)\right\}\,.$$

See our paper for the equivariant version.

Application to GNNs

For all GNNs studied, the technical condition on $\mathcal{F}_{scal}$ is satisfied !

As a consequence, we show that :

$$\overline{\mathrm{GNN}} \quad = \quad \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(\mathrm{GNN}) \subset \rho(f)\} .$$

Application to GNNs

For all GNNs studied, the technical condition on $\mathcal{F}_{scal}$ is satisfied !

As a consequence, we show that :

$$\overline{\mathrm{GNN}} \quad = \quad \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(\mathrm{GNN}) \subset \rho(f)\}\,.$$

Recall : $\rho(\mathrm{MGNN}) = \rho(2\text{-WL})$

so that : $\overline{\mathrm{MGNN}} = \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(2\text{-WL}) \subset \rho(f)\}$

Application to GNNs

For all GNNs studied, the technical condition on $\mathcal{F}_{scal}$ is satisfied !

As a consequence, we show that :

$$\overline{\text{GNN}} \quad = \quad \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(\text{GNN}) \subset \rho(f)\}\,.$$

Recall : $\rho(\text{MGNN}) = \rho(2\text{-WL})$

so that : $\overline{\text{MGNN}} = \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(2\text{-WL}) \subset \rho(f)\}$

More generally, we obtain the expressive power of Linear GNN (k-LGNN) and Folklore GNN (k-FGNN) with tensors of order k :

$$\begin{aligned}
\overline{k\text{-LGNN}} \quad &= \quad \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(k\text{-WL}) \subset \rho(f)\} \\
\overline{k\text{-FGNN}} \quad &= \quad \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho((k+1)\text{-WL}) \subset \rho(f)\}
\end{aligned}$$

Learning with (practical i.e. $k = 2$) FGNN

Better expressive power with FGNN

(Maron et al., 2019) adapted the Folklore version of the Weisfeiler-Lehman test to propose the folklore graph layer (FGL) :

$$h_{i\to j}^{\ell+1} = f_0\left(h_{i\to j}^{\ell}, \sum_{k\in V} f_1\left(h_{i\to k}^{\ell}\right) f_2\left(h_{k\to j}^{\ell}\right)\right),$$

where f_0, f_1 and f_2 are learnable functions.

For FGNNs, messages are associated with pairs of vertices as opposed to MGNN where messages are associated with vertices.

FGNN : a FGNN is the composition of FGLs and a final invariant/equivariant reduction layer from $\mathbb{F}^{n^2}$ to $\mathbb{F}/\mathbb{F}^n$.

A small diversion

Prop : Let $f : [0,1]^{2\times n} \to \mathbb{R}$ be a continuous function. f is invariant if and only if there exist continuous functions $\rho : \mathbb{R}^M \to \mathbb{R}$, $\Phi_1 : [0,1] \to \mathbb{R}^M$ and $\Phi_2 : [0,1] \to \mathbb{R}^M$ such that

$$f\left((x_k, y_k)_{k=1}^n\right) = \rho\left(\sum_{k=1}^n \Phi_1(x_k)\Phi_2(y_k)\right),$$

where the product $\Phi_1(x_k)\Phi_2(y_k)$ is component-wise and $M = \binom{n+2}{2}$.

A small diversion

Prop : Let $f : [0,1]^{2\times n} \to \mathbb{R}$ be a continuous function. f is invariant if and only if there exist continuous functions $\rho : \mathbb{R}^M \to \mathbb{R}$, $\Phi_1 : [0,1] \to \mathbb{R}^M$ and $\Phi_2 : [0,1] \to \mathbb{R}^M$ such that

$$f\left((x_k, y_k)_{k=1}^n\right) = \rho\left(\sum_{k=1}^n \Phi_1(x_k)\Phi_2(y_k)\right),$$

where the product $\Phi_1(x_k)\Phi_2(y_k)$ is component-wise and $M = \binom{n+2}{2}$.

Proof : The ring of multisymmetric polynomials in n (vector-valued) variables is generated by the multisymmetric power sums of total degree $\leq n$: $p_\alpha(x_1, y_1, x_2, y_2, \ldots, y_n) = \sum_{i=1}^n x_i^{\alpha_1} y_i^{\alpha_2}$, with $\alpha_1 + \alpha_2 \leq n$.

Newton identities relating elementary symmetric functions and power sums.

A question...

What happens for polynomials $p \in \mathbb{R}[X_{11}, X_{12}, ..., X_{nn}]$ in n^2 variables which are invariant with respect of the action of the symmetric group as follows :

$$p(x_{11}, x_{12}, ..., x_{nn}) = p(x_{\sigma(1)\sigma(1)}, x_{\sigma(2)\sigma(2)}, ..., x_{\sigma(n)\sigma(n)}).$$

In words, the input of the polynomial is a $n \times n$ matrix and it should be invariant to the permutation of its rows and columns. I would like to know about explicit generators in this setting; a minimal set ?

A question...

What happens for polynomials $p \in \mathbb{R}[X_{11}, X_{12}, ..., X_{nn}]$ in n^2 variables which are invariant with respect of the action of the symmetric group as follows :

$$p(x_{11}, x_{12}, ..., x_{nn}) = p(x_{\sigma(1)\sigma(1)}, x_{\sigma(2)\sigma(2)}, ..., x_{\sigma(n)\sigma(n)}).$$

In words, the input of the polynomial is a $n \times n$ matrix and it should be invariant to the permutation of its rows and columns. I would like to know about explicit generators in this setting; a minimal set ?

A reference ?

Properties of Folklore GNN (FGNN)

(Maron et al., 2019) Prop : FGL is equivariant and $\rho(\text{FGNN}) = \rho(3\text{-WL})$.

Properties of Folklore GNN (FGNN)

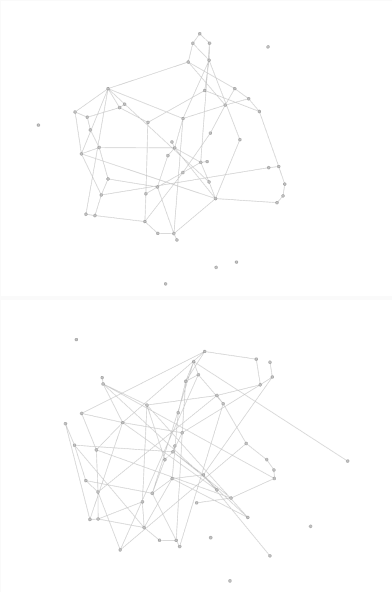
(Maron et al., 2019) Prop : FGL is equivariant and $\rho(\text{FGNN}) = \rho(3\text{-WL})$.

Approximation for FGNN :

$$\overline{\text{FGNN}} = \{f \in \mathcal{C}(X, \mathbb{F}) : \ \rho(3\text{-WL}) \subset \rho(f)\}$$

FGNN has the best power of approximation among all architectures working with tensors of order $\mathbf{2}$ presented so far.

Learning the graph alignment problem with Siamese FGNNs



$$G_1 \in \{0,1\}^{n^2} \xrightarrow{\text{FGNN}} E_1 \in \mathbb{R}^{n \times b}$$

$$G_2 \in \{0,1\}^{n^2} \xrightarrow{\text{FGNN}} E_2 \in \mathbb{R}^{n \times b}$$

$$E_1 E_2^T \in \mathbb{R}^{n^2}$$

From the node similarity matrix $E_1 E_2^T$, we extract a mapping from nodes of G_1 to nodes of G_2.

Graphs distributions

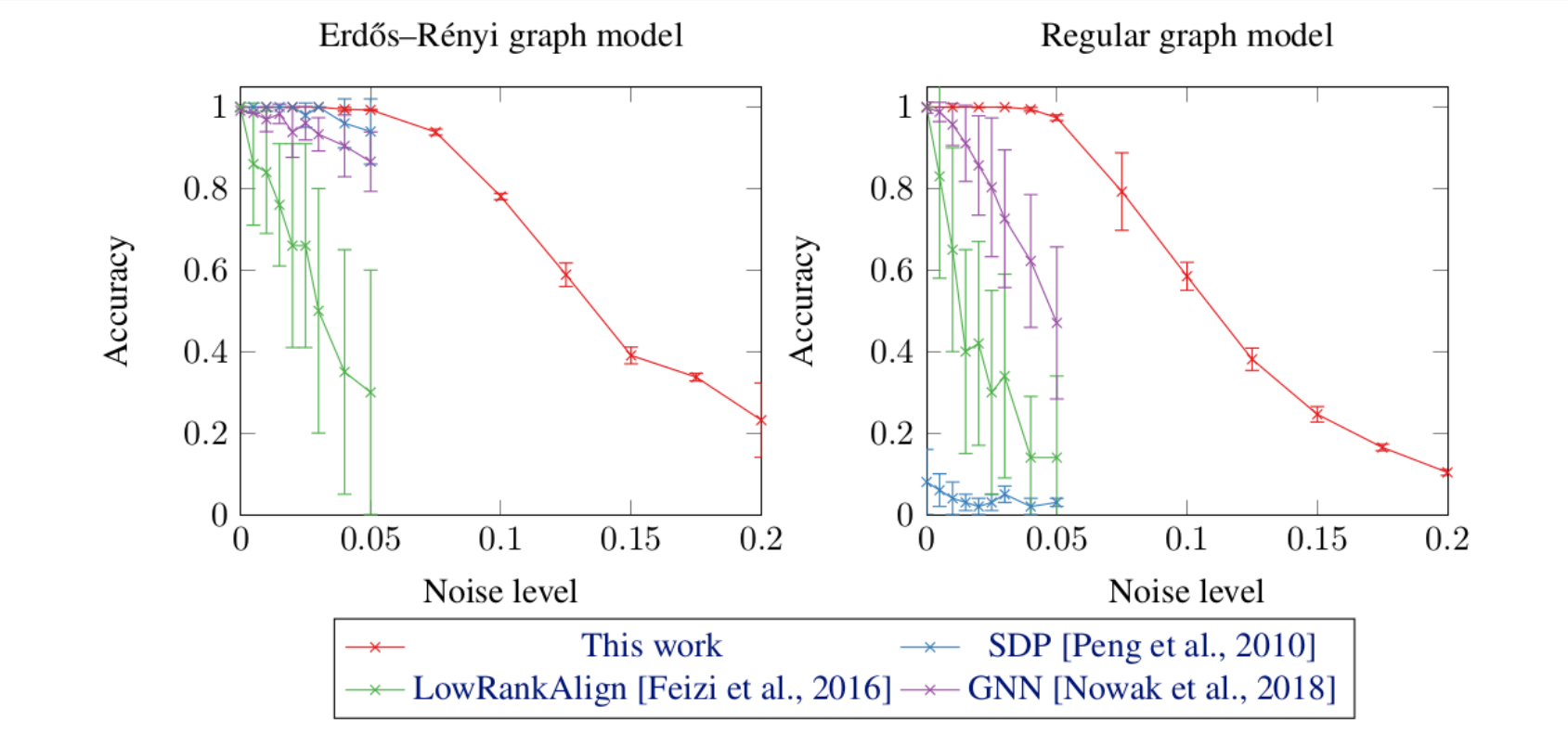
We tested on two graphs distributions :

Erdős–Rényi : each edge added with some probability p

d**-Regular :** each node has d neighbors

$\rightarrow$ considered as hard examples

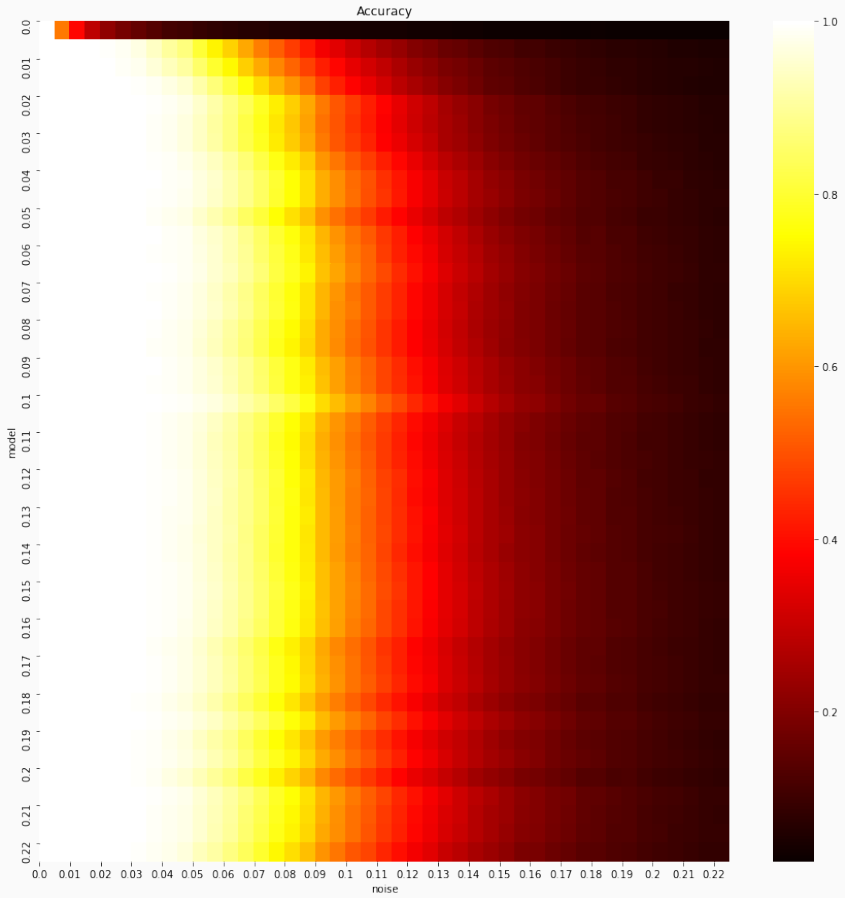
Results on synthetic data



- Graphs : $n = 50$, density = 0.2
- Training set : 20000 samples
- Validation and Test sets : 1000 samples

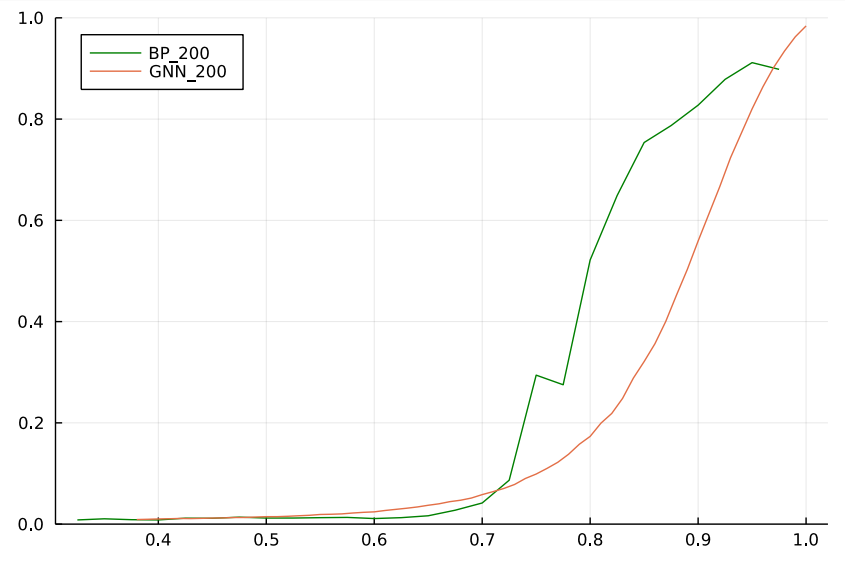
SDP from (Peng et al., 2010), message passing from (Nowak et al., 2018).

Generalization property



Each line corresponds to a model trained at a given noise level and shows its accuracy across all noise levels.

Comparison with BPAlign



Comparison with a recent smart algorithm designed with Luca Ganassali and Laurent Massoulié : Correlation detection in trees for partial graph alignment

Conclusion

The Bitter Lesson by Rich Sutton

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law (...) Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. (...) the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation.

Thank You!

Références

H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32 : Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 2153–2164, 2019. URL **http://papers.nips.cc/paper/8488-provably-powerful-graph-networks**.

A. Nowak, S. Villar, A. S. Bandeira, and J. Bruna. Revised note on learning quadratic assignment with graph neural networks. In *2018 IEEE Data Science Workshop, DSW 2018, Lausanne, Switzerland, June 4-6, 2018*, pages 229–233. IEEE, 2018. doi : 10.1109/DSW.2018.8439919. URL **https://doi.org/10.1109/DSW.2018.8439919**.

J. Peng, H. D. Mittelmann, and X. Li. A new relaxation framework for quadratic assignment problems based on matrix splitting. *Math. Program. Comput.*, 2 (1) :59–77, 2010. doi : 10.1007/s12532-010-0012-6. URL **https://doi.org/10.1007/s12532-010-0012-6**.

V. Timofte. Stone–weierstrass theorems revisited. *Journal of Approximation Theory*, 136(1) :45 – 59, 2005. ISSN 0021-9045. doi : https://doi.org/10.1016/j.jat.2005.05.004. URL **http:**