Recent Advances and Challenges in Machine Learning Representations of Molecules

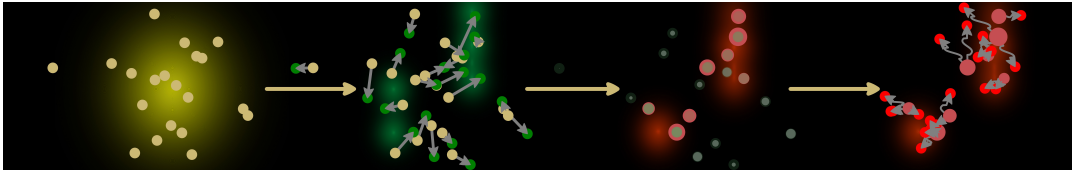
Julien Mairal

Inria Grenoble

My team at Inria: Who are we?

Some of my research topics

- Statistical machine learning and optimization

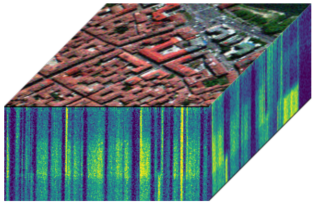


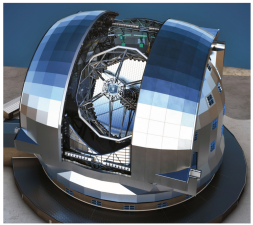
Some of my research topics

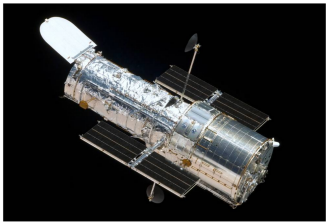
- Statistical machine learning and optimization
- Representation learning of images.

Some of my research topics

- Statistical machine learning and optimization
- Representation learning of images.
- **Machine learning for science (pluri-disciplinary collaborations)**
  - scientific imaging (ex: exoplanet detection, molecular microscopy).
  - Earth observation (remote sensing).

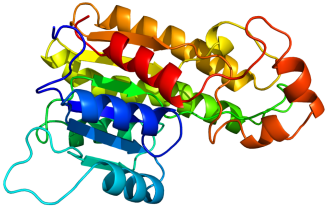


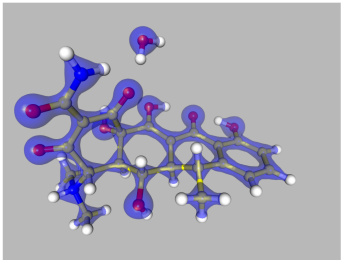


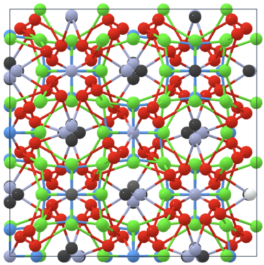


Some of my research topics

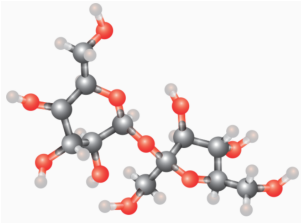
- Statistical machine learning and optimization
- Representation learning of images.
- **Machine learning for science (pluri-disciplinary collaborations)**
  - scientific imaging (ex: exoplanet detection, molecular microscopy).
  - Earth observation (remote sensing).
  - **Graph representations for material science and computational biology**.



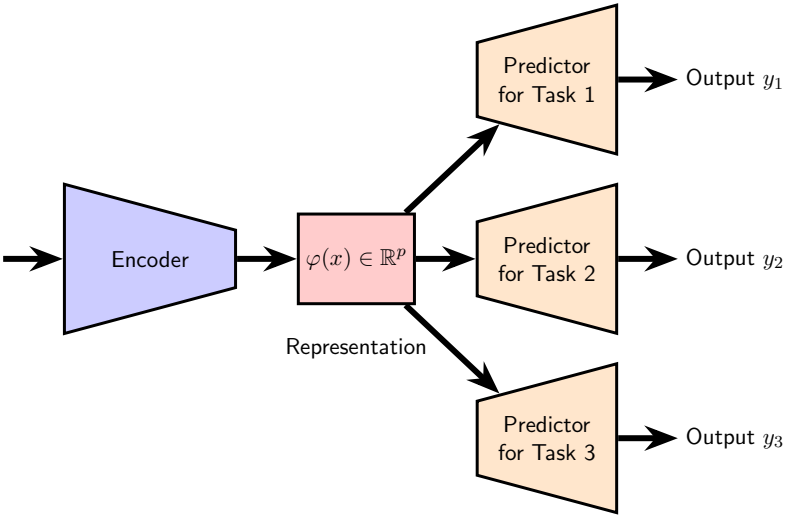




Representation learning for molecules



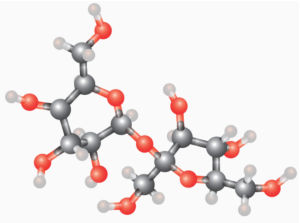
Input x



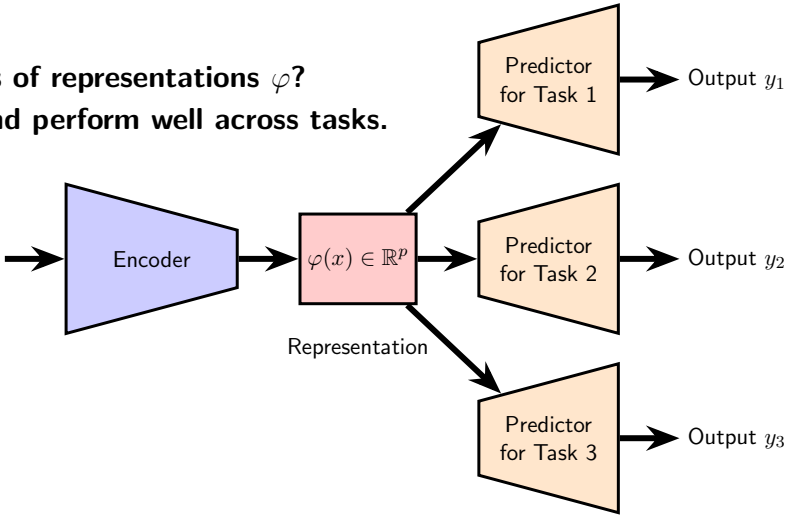
Representation learning for molecules

What is the right class of representations φ?

φ should be generic and perform well across tasks.



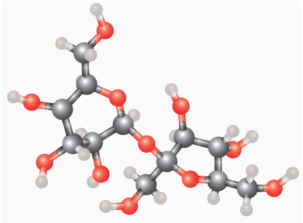
Input x



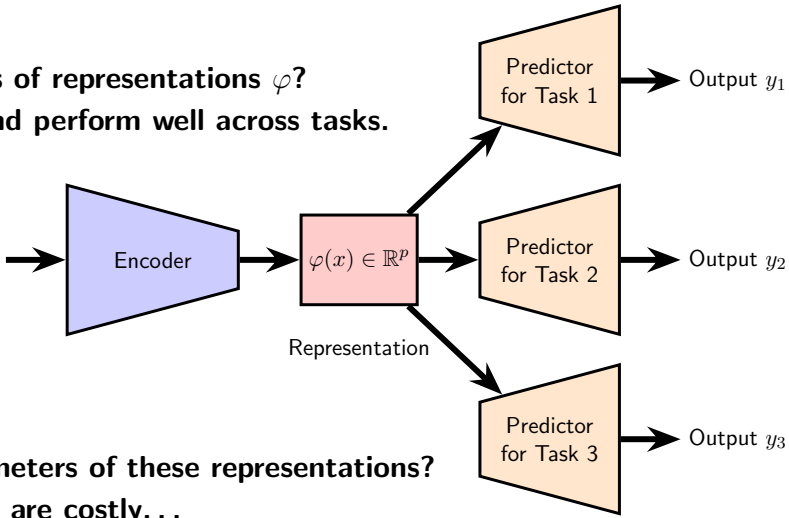
Representation learning for molecules

What is the right class of representations φ?

φ should be generic and perform well across tasks.



Input x



How to learn the parameters of these representations?

Annotations are costly...

This talk:

1. A story that has been **very** successful in computer vision to learn generic image representations $\varphi(x)$, trained on a large corpus of images with no annotations.
   What are the opportunities/difficulties for chemistry?

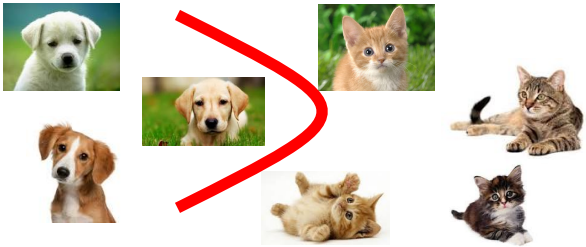
This talk:

1. A story that has been **very** successful in computer vision to learn generic image representations $\varphi(x)$, trained on a large corpus of images with no annotations.
   What are the opportunities/difficulties for chemistry?
2. A **very** short survey of classical graph representations in machine learning.
   What are the current challenges?

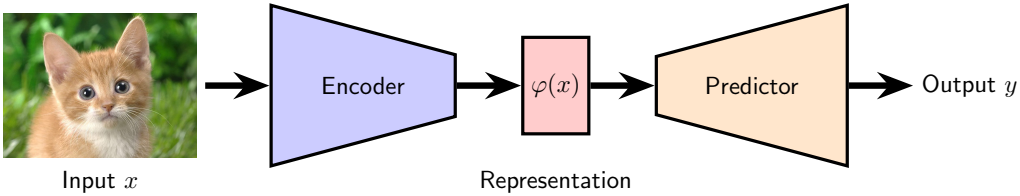
What is supervised learning?

The goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

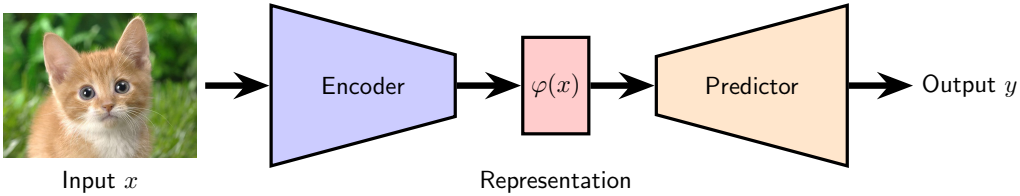
$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$



What is representation learning?



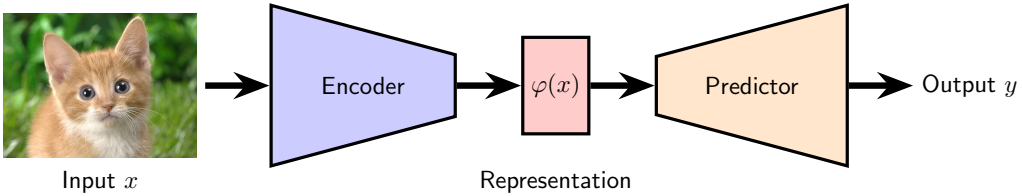
What is representation learning?



Handcrafted representations (encoder is predefined)

- traditional representations based on domain knoweldge (*e.g.*, SIFT [Lowe, 2004]).
- the predictor is **typically linear** $f(x) = W\varphi(x)$.
- $\varphi(x)$ may be very **high-dimensional** (reasonable expressiveness).

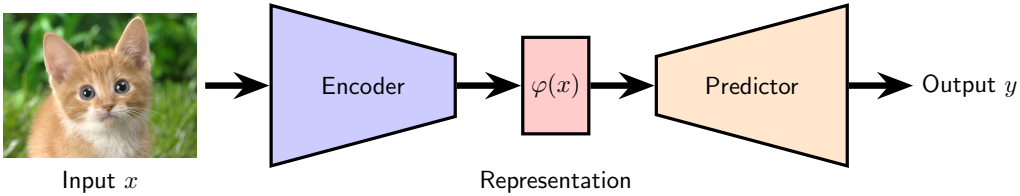
What is representation learning?



Learned representations with neural networks

- the encoder's architecture is **adapted to images** (*e.g.*, convolutional neural networks).
- the predictor is often **simple** (linear model or multilayer perceptron).
- for more complex tasks, the predictor is also **adapted to the output structure** (*e.g.*, U-Net decoder for semantic segmentation in images).

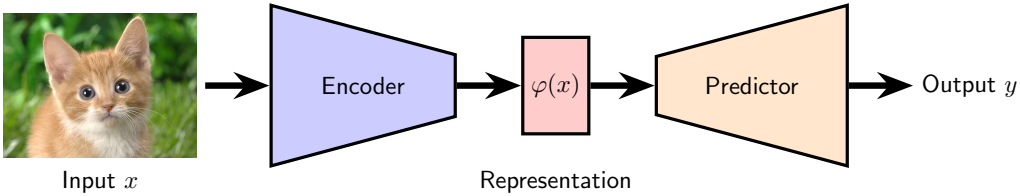
What is self-supervised learning?



Tentative definition and remarks

- learning "good" representations $\varphi(x)$ **with prediction tasks in mind**, but...
- **without having access to any label** y (unsupervised learning).
- achieved by **finding supervisory signals within the data** and/or with **pretext tasks**.

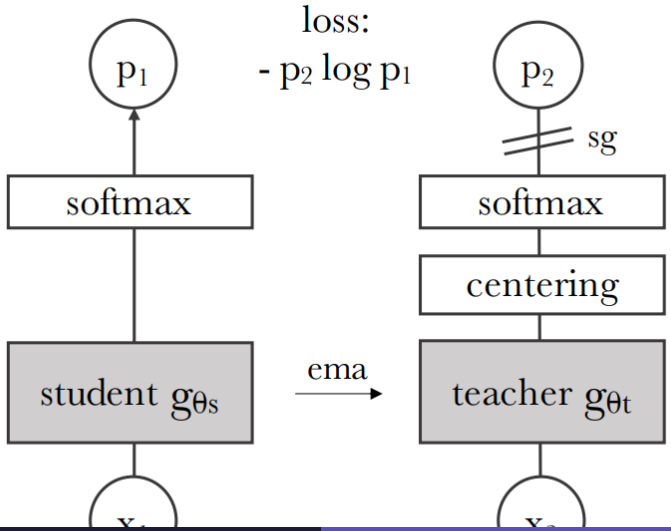
What is self-supervised learning?



Multiple purposes

- finding representations for learning simple predictors when annotations are scarce.
- harnessing information from massive unannotated databases.
- finding **generic** representations that perform well on all visual recognition tasks (foundation models).

from SwAV to DINO with self-distillation



Pretext tasks?

I want to solve task A but I do not have (much) annotated data.

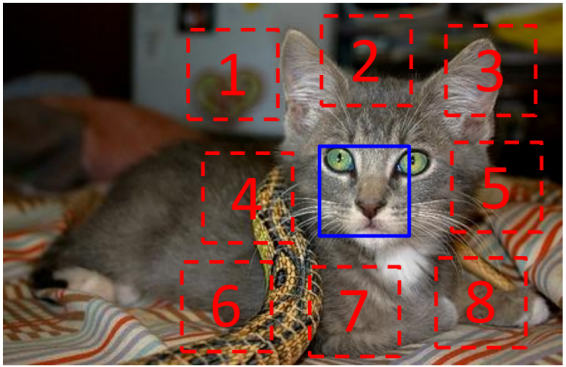
Pretext tasks?

I want to solve task A but I do not have (much) annotated data.

Perhaps a representation $\varphi(x)$ that is good for task B will also be good for task A?

Example: Spatial context prediction

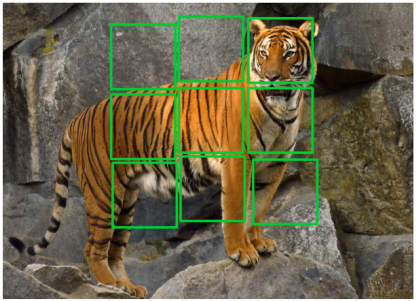
Picture courtesy of Doersch et al. [2015]



$X = ($  ,  $); Y = 3$

Example: Spatial context prediction

Picture courtesy of Noroozi and Favaro [2016]



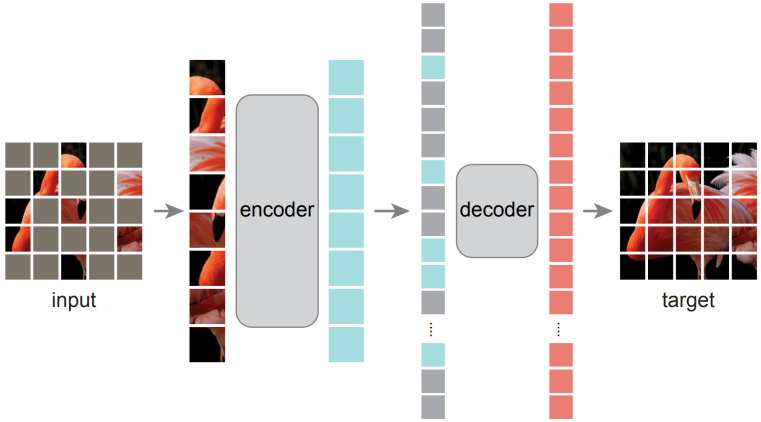
(a)

(b)

(c)

Example: Masked auto-encoders (also context prediction)

Picture courtesy of He et al. [2022]



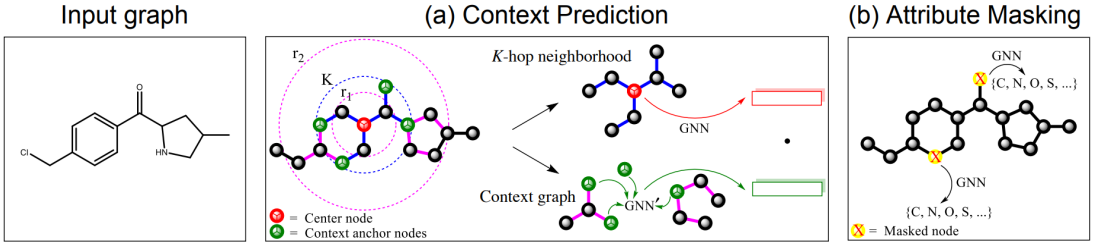
- inspired from masked language modeling [Devlin et al., 2018], revolution in NLP.

Your turn: which pretext tasks for molecular representations?

Your turn: which pretext tasks for molecular representations?

Picture courtesy of Hu et al. [2019].

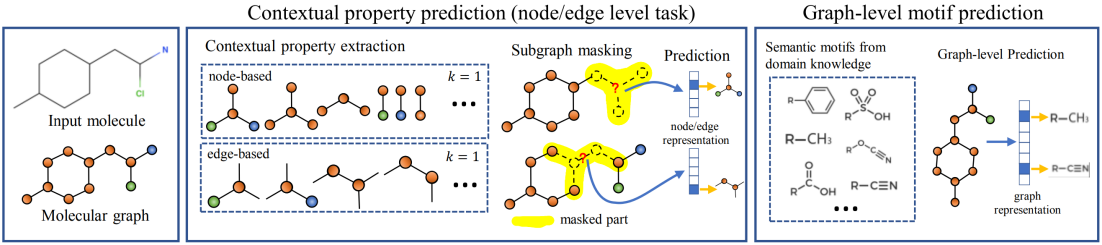
First idea: a good representation φ should be useful for context prediction tasks



Your turn: which pretext tasks for molecular representations?

Picture courtesy of Rong et al. [2020].

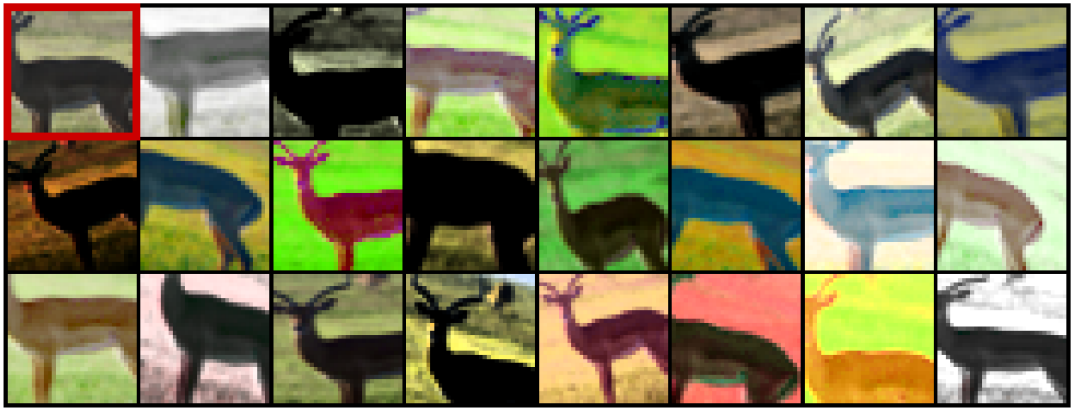
First idea: a good representation φ should be useful for context prediction tasks



Back to computer vision: Harnessing data augmentation

Picture courtesy of Dosovitskiy et al. [2014]

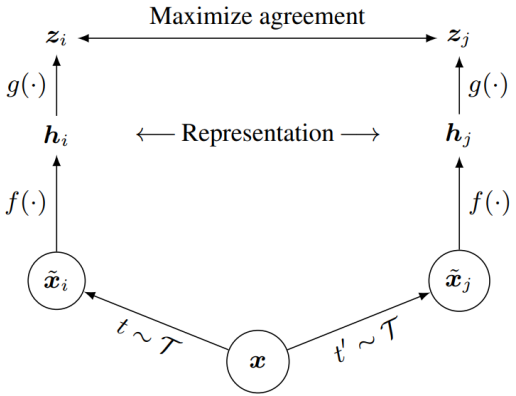
Use data augmentation to create "classes" around each sample.



Harnessing data augmentation and contrastive learning

SimCLR, Picture courtesy of Chen et al. [2020]

Second idea: a good representation φ should make augmented views of the same image closer and push apart different images.

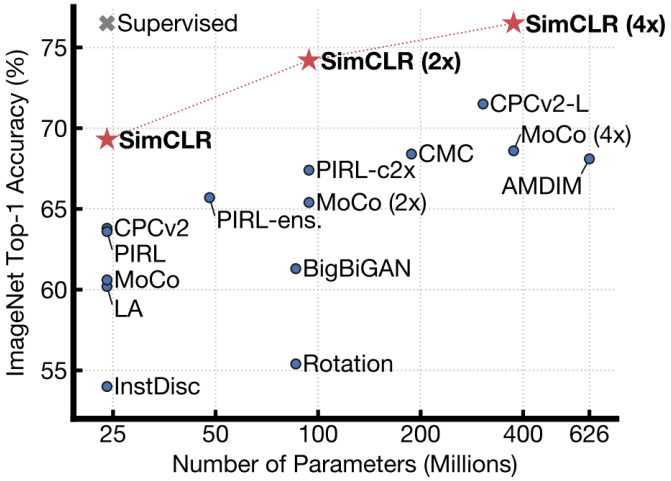


$$\ell_{i,j} = -\log\left(\frac{e^{\mathsf{sim}(\mathbf{z}_i,\mathbf{z}_j)}}{\sum_{i\neq k} e^{\mathsf{sim}(\mathbf{z}_i,\mathbf{z}_k)}}\right).$$

- trained online with large batch sizes.
- strong data augmentation.

Harnessing data augmentation and contrastive learning

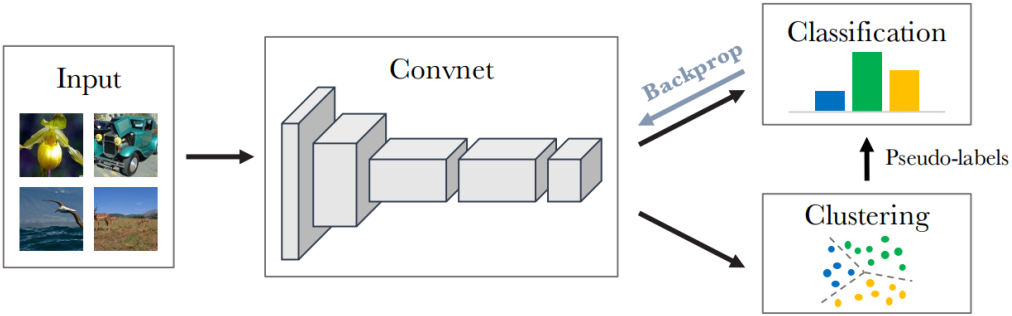
SimCLR, Picture courtesy of Chen et al. [2020]



Uncovering hidden structures in images: DeepCluster

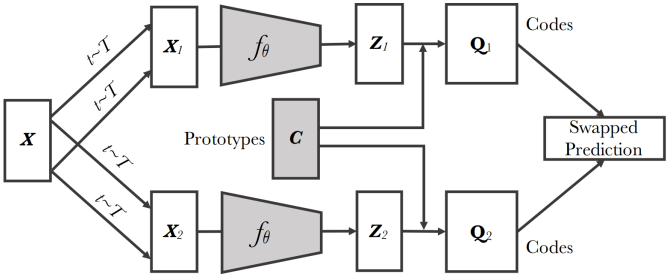
Picture courtesy of Caron et al. [2018]

Third idea: a good representation φ should uncover data clusters.



Clustering, contrastive learning, and context prediction: SwAV

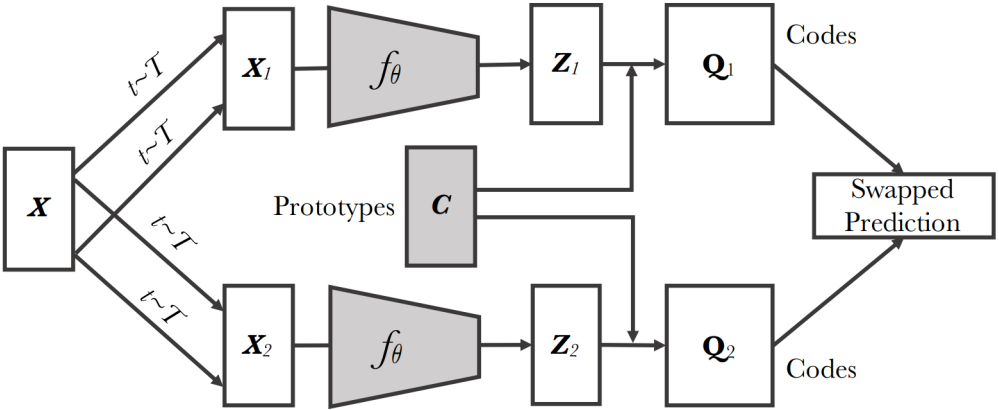
Picture courtesy of Caron, Misra, Mairal, Goyal, Bojanowski, and Joulin [2020]



Recipe

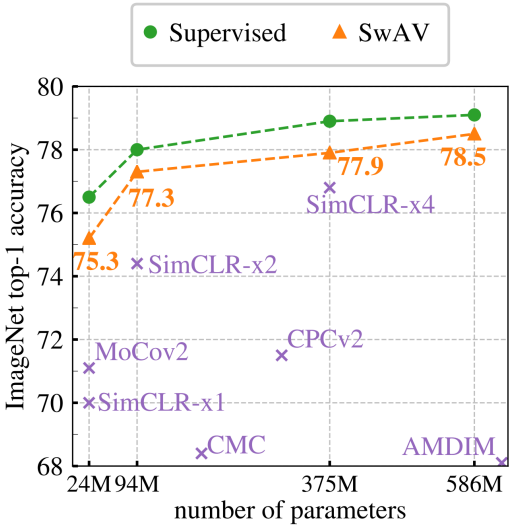
- **clustering**: prototypes $\approx$ centroids. Trivial solutions avoided by optimal transport.
- **contrastive learning with data augmentation** but no explicit negative pairs.
- **context prediction**: predicting global crops from local crops (multicrop).

Clustering, contrastive learning, and context prediction: SwAV



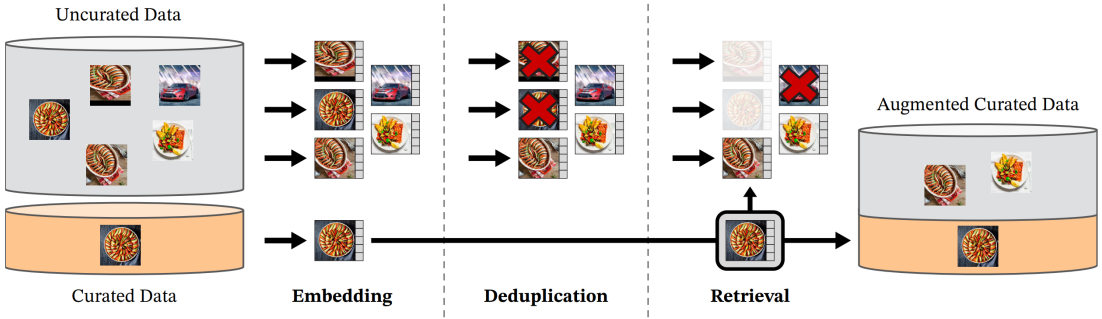
Clustering, contrastive learning, and context prediction: SwAV

Picture courtesy of Caron, Misra, Mairal, Goyal, Bojanowski, and Joulin [2020]



A foundation model for images: DINOv2

- DINO: a more recent model with self-distillation [Caron et al., 2021];
- DINOv2: foundation model trained **well-engineered data** [Oquab et al., 2024].



Opportunities/challenges for molecular representation

What would be a foundation model for molecules/materials?

1. Which **model architecture**? (see second part of this talk).

Opportunities/challenges for molecular representation

What would be a foundation model for molecules/materials?

1. Which **model architecture**? (see second part of this talk).
2. Which **learning algorithm**? Should we follow the self-supervised computer vision recipe? How to design data augmentation strategies?

Opportunities/challenges for molecular representation

What would be a foundation model for molecules/materials?

1. Which **model architecture**? (see second part of this talk).
2. Which **learning algorithm**? Should we follow the self-supervised computer vision recipe? How to design data augmentation strategies?
3. What for? What are the **downstream tasks of interest**?

Opportunities/challenges for molecular representation

What would be a foundation model for molecules/materials?

1. Which **model architecture**? (see second part of this talk).
2. Which **learning algorithm**? Should we follow the self-supervised computer vision recipe? How to design data augmentation strategies?
3. What for? What are the **downstream tasks of interest**?
4. **How to engineer a good dataset?**

Part II: A few machine learning models for molecules

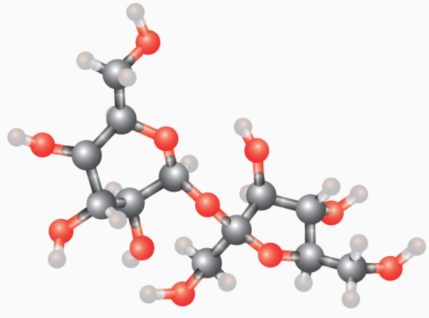
Part II: A few ~~machine~~ learning models for ~~molecules~~

Part II: A few deep learning models for graphs

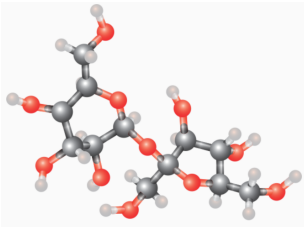
Molecular Graphs for Deep Learning Models

Ex: ZINC or OGB datasets

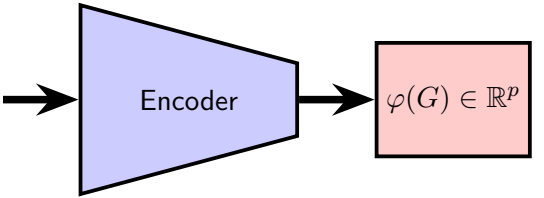
- Nodes are *atoms*, edges are *bonds*.
- Node features can be *atom-type*, *spatial position*, ...
- Edge features are bond types (*single*,*double*, *triple*).



Learning graph representations



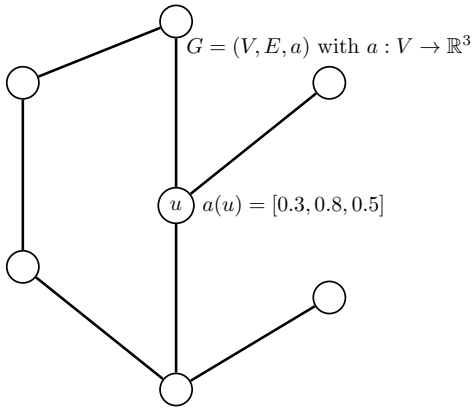
Input G



Representation

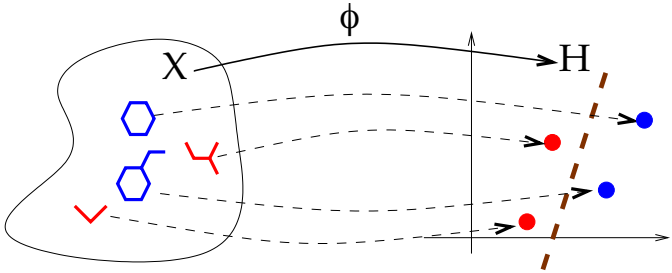
- **Expressiveness**: Find a representation (vector) that is able to discriminate graphs with different structures (distinguish non-isomorphic graphs as best as possible).
- **Tractability**: The representation should be efficiently computable on modern hardware.
- **Learnable**: One should be able to adapt the representation to the task and to the data.
- **Taking into account physics**: long-range potentials, 3D geometry, symmetries...

Graphs with node attributes



- We consider graphs $G = (V, E, a)$ where V and E are the sets of vertices and edges,
- and $a : V \to \mathbb{R}^p$ is a function assigning attributes to each node.

Classical (non-deep) graph representations



- Map each graph G to a vector $\varphi(G)$ in $\mathbb{R}^p$, which lends itself to learning tasks.
- A large class of kernel mappings can be written in the form

$$\varphi(G) := \sum_{u \in \mathcal{V}} \varphi_{\text{base}}(\ell_G(u)) \quad \text{where } \varphi_{\text{base}} \text{ embeds some local patterns } \ell_G(u) \text{ to } \mathbb{R}^p.$$

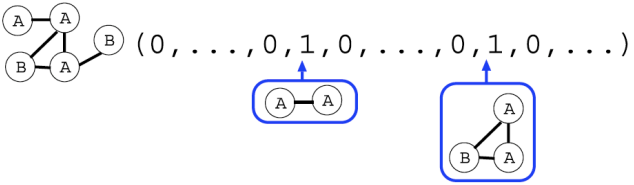
[Shervashidze et al., 2011, Lei et al., 2017, Kriege et al., 2019]

Classical (non-deep) graph representations

Find a high-dimensional representation $\varphi(G)$ for which we can efficiently compute

$$K(G, G') = \langle \varphi(G), \varphi(G') \rangle.$$

There is a very rich literature about graph kernels performing (implicitly or explicitly) **substructure enumeration**.



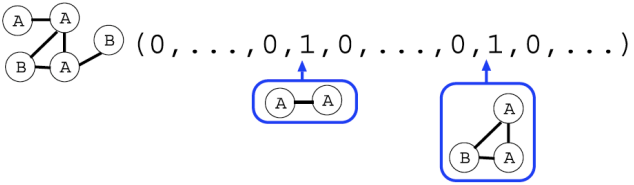
- **subgraphs and path** kernels (NP-hard, [Gärtner et al., 2003]).
- **walk** kernels [Kashima et al., 2003, Mahé et al., 2004].
- **shortest-path** kernels [Borgwardt and Kriegel, 2005].
- **graphlets** kernels [Shervashidze et al., 2009].
- **Weisfeiler-Lehman** kernel [Shervashidze et al., 2011].

Classical (non-deep) graph representations

Find a high-dimensional representation $\varphi(G)$ for which we can efficiently compute

$$K(G, G') = \langle \varphi(G), \varphi(G') \rangle.$$

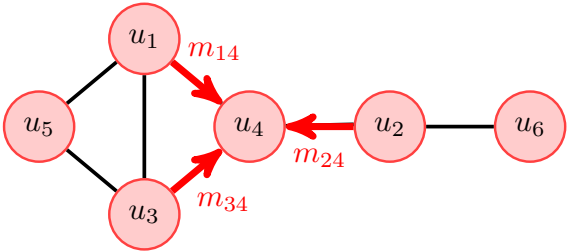
There is a very rich literature about graph kernels performing (implicitly or explicitly) **substructure enumeration**.



For a review, see the course material

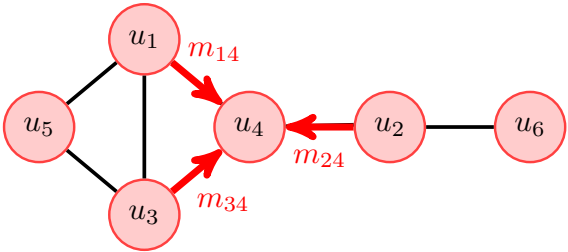
- `https://mva-kernel-methods.github.io/course-2023-2024/`

Graph neural networks with message passing



- a **multi-layer representation**: for each node u and layer k, we store a vector $\varphi_k(u)$.
- by increasing k, $\varphi_k(u)$ contains information about a **larger neighborhood**.
- final graph representation is obtained by **pooling** $\varphi(G) = \sum_{u \in V} \varphi_K(u) \in \mathbb{R}^p$.

Graph neural networks with message passing

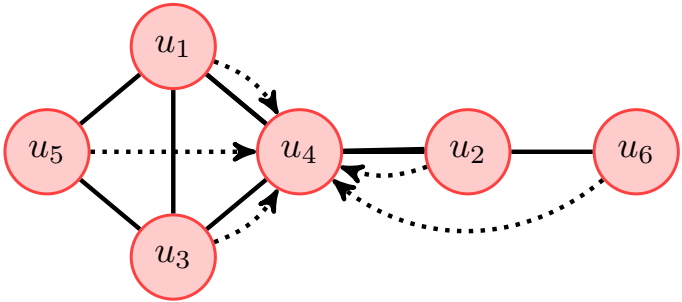


- Layer k is built from layer $k-1$ by **message passing**

$$\begin{aligned}\varphi_k(u) &= \mathsf{Process}(\varphi_{k-1}(u), \{\varphi_{k-1}(v) : v \in \mathcal{N}(u)\}) \\ &= \sum_{v \in \mathcal{N}(u) \cup u} \mathsf{ReLU}(Z_k^\top \varphi_{k-1}(v)) \quad \text{(for example)}\end{aligned}$$

- There are many, many variants (*e.g.*, GCN [Kipf and Welling, 2017]).

Graph transformers



- G. Mialon, D. Chen, M. Selosse, and J. Mairal. GraphiT: Encoding Graph Structure in Transformers. *arXiv:2106.05667*. 2021.
- R. Menegaux, E. Jehanno, M. Selosse and J. Mairal. Self-Attention in Colors: Another Take on Encoding Graph Structure in Transformers. *TMLR*. 2023.

From GNNs to Graph transformers

An example of GNN layer (GCN, Kipf and Welling, 2017)

$$\varphi_k(u) = \mathsf{ReLU}\left(Z_k^\top \left(\frac{1}{|\mathcal{N}(u)|+1} \sum_{v \in \mathcal{N}(u) \cup u} \varphi_{k-1}(v)\right)\right).$$

From GNNs to Graph transformers

An example of GNN layer (GCN, Kipf and Welling, 2017)

$$\varphi_k(u) = \mathsf{ReLU}\left(Z_k^\top \left(\frac{1}{|\mathcal{N}(u)|+1} \sum_{v \in \mathcal{N}(u) \cup u} \varphi_{k-1}(v)\right)\right).$$

The basic transformer layer with self attention

$$\varphi_k(u) = \mathsf{ReLU}\left(Z_k^\top \left(\varphi_{k-1}(u) + \sum_{v \in V} A_k[u,v]\varphi_{k-1}(u)\right)\right)$$

$$\text{with} \quad A_k = \mathsf{Softmax}\left(\frac{\varphi_{k-1} Q_k^\top K_k \varphi_{k-1}^\top}{\sqrt{d}}\right).$$

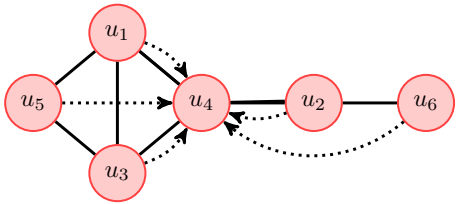
(Note that a classical residual connection has been removed for simplicity).

From GNNs to Graph transformers

The basic transformer layer with self attention

$$\varphi_k(u) = \mathsf{ReLU}\left(Z_k^\top\left(\varphi_{k-1}(u) + \sum_{v\in V} A_k[u,v]\varphi_{k-1}(u)\right)\right)$$

$$\text{with} \quad A_k = \mathsf{Softmax}\left(\frac{\varphi_{k-1}Q_k^\top K_k \varphi_{k-1}^\top}{\sqrt{d}}\right).$$



From GNNs to Graph transformers

The basic transformer layer with self attention

$$\varphi_k(u) = \text{ReLU}\left(Z_k^\top \left(\varphi_{k-1}(u) + \sum_{v \in V} A_k[u,v]\varphi_{k-1}(u)\right)\right)$$

$$\text{with} \quad A_k = \text{Softmax}\left(\frac{\varphi_{k-1} Q_k^\top K_k \varphi_{k-1}^\top}{\sqrt{d}}\right).$$

Challenges

- How to encode the **graph structure**? (note that if we multiply elementwise the attention matrix by the adjacency matrix, we are back to message passing)
- How to take into account **edge features**?

Graph transformers: recipes

How to take into account edge features?

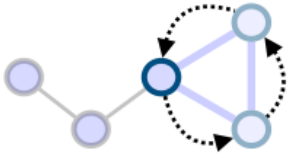
- treat edge features as node features with additional variables $E_k(u, v)$ undergoing "similar" updates.

Local structure encoding

- Enrich input features. A successful feature is based on the diagonals of random walk kernels

$$p(u) = [RW_{uu}, \ldots, RW^p_{uu}]$$

where RW^p_{uu} probability for a p-step random walk to loop back to node u:



[Dwivedi and Bresson, 2020, Rampášek et al., 2022, Lim et al., 2022]

Graph transformers: recipes

Modulate the attention matrix with relative positional encoding

- Graphormer computes an average of the dot-products of edge feature and a learnable embedding along shortest paths

$$A = \text{Softmax}\left(\frac{f_{k-1} W_Q^\top W_K f_{k-1}^\top}{\sqrt{d}} + B_k^{\text{shortest-paths}}\right).$$

- GraphiT weights the attention with a diffusion kernel. This captures both short-range and long-range graph topology

$$A = \text{Normalize}\left(\text{Exp}\left(\frac{f_{k-1} W_Q^\top W_K f_{k-1}^\top}{\sqrt{d}}\right) \circ K_\sigma\right).$$

[Ying et al., 2021, Mialon et al., 2021]

Graph transformers: recipes

Modulate the attention matrix with relative positional encoding

- GraphiT uses a hard-coded kernel and does not include edge features in the attention.
- CSA first enriches original edge features with random walks kernels:

$$E_{uv}^{\mathsf{rw}} = [RW_{uv}, \ldots, RW_{uv}^{p}]$$

and then learns how to exploit these features to modulate the attention matrix

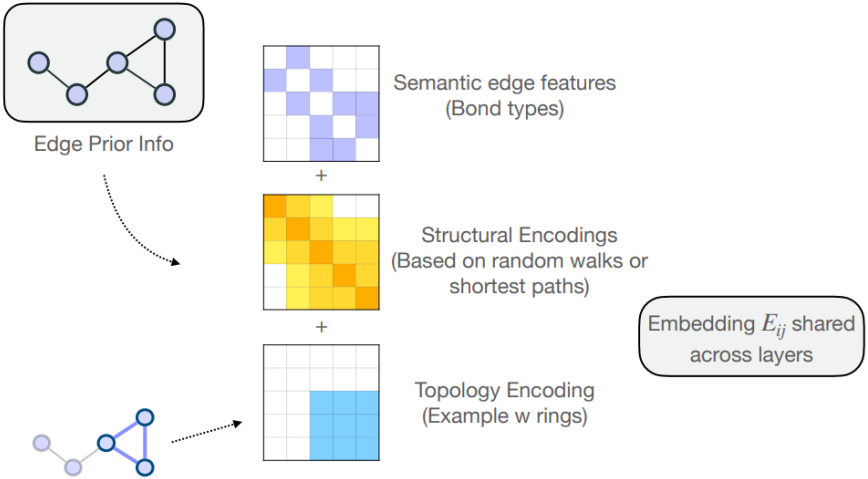
$$A = \mathsf{Softmax}\left(\frac{f_{k-1}W_Q^\top W_K f_{k-1}^\top}{\sqrt{d}} + W_E^\top E_{k-1}\right).$$

Additional tricks

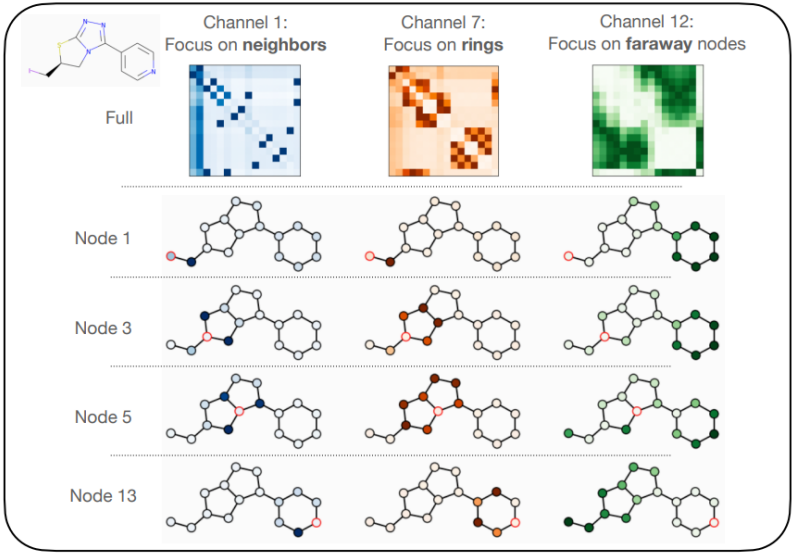
- introduce features for structures that are known to be useful (carbon rings).

[Menegaux et al., 2023]

All of this summarized in a pretty picture



Visualizing self attention



Benchmarks

	Model	ZINC MAE ↓
MPNN	GCN (Kipf & Welling, 2017)	0.367 ± 0.011
	GatedGCN (Dwivedi et al., 2022a)	0.090 ± 0.001
	GPS (Rampášek et al., 2022)	0.070 ± 0.004
h-MPNN	CIN (Bodnar et al., 2021a)	0.079 ± 0.006
	CRaWL (Toenshoff et al., 2021)	0.085 ± 0.004
	GIN-AK+ (Zhao et al., 2022)	0.080 ± 0.001
Transformers	SAN (Kreuzer et al., 2021)	0.139 ± 0.006
	Graphormer (Ying et al., 2021)	0.122 ± 0.006
	SAT (Chen et al., 2022)	0.094 ± 0.008
	EGT (Hussain et al., 2022)	0.108 ± 0.009
	GRPE (Park et al., 2022)	0.094 ± 0.002
	CSA (ours)	$\mathbf{0.070 \pm 0.003}$
	CSA-rings (ours)	$\mathbf{0.056 \pm 0.002}$

(12k graphs)

Benchmarks

	Model	PCQM4Mv2 (4M graphs)	
		Validation MAE ↓	# Param.
MPNN	GCN	0.1379	2.0M
	GCN-virtual	0.1153	4.9M
	GIN	0.1195	3.8M
	GIN-virtual	0.1083	6.7M
Transformers	Graphormer	0.0864	48.3M
	EGT	0.0869	89.3M
	GRPE	0.0890	46.2M
	GPS-small	0.0938	6.2M
	GPS-medium	0.0858	19.4M
	CSA-small (ours)	0.0898	2.8M
	CSA-deep (ours)	**0.0853**	8.3M

Physics and Geometry

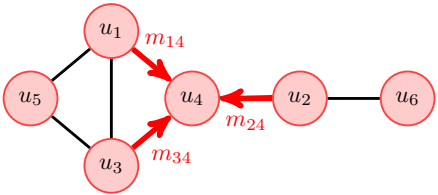
Challenges (not addressed in this presentation)

- Is there **another structure within** the graph? (*e.g.*, chain of amino acids for proteins).
- Is the graph part of a **larger structure** (crystallography)?
- Does the representation model the **right symmetries and inv/equivariances**?
- Is the graph construction satisfactory? What about **long-range interactions**?

Recap: graph representations with deep learning

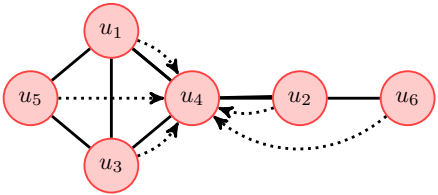
Graph neural networks with message passing

- multi-layer construction.
- sequence of local operations.
- limited expressivity [Xu et al., 2019].



Graph transformers

- non-local operations with attention.
- how to encode the graph structure?



For a detailed review, see

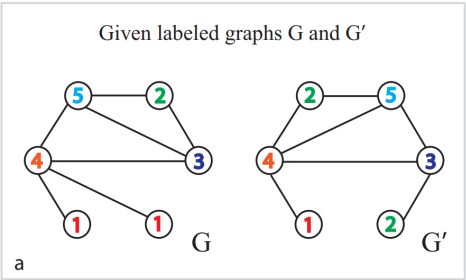
- graph neural networks for 3D atomic systems: [Duval et al., 2023].
- survey on graph transformers: [Müller et al., 2023].
- course material from Xavier Bresson `https://lnkd.in/dZZWay3Z`.

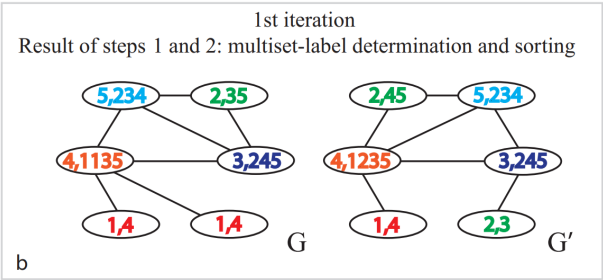
Bonus: Relation between Weisfeler-Lehman and graph neural networks

Principles of the Weisfeiler-Lehman kernel

Consider a graph $G = (V, E, a)$ with discrete labels $l_0(u) = a(u)$ at each vertex u.

- This is a **multi-layer** construction producing new labels $l_k(u)$ for each vertex at layer k.
- A label $l_k(u)$ represents $(l_{k-1}(u), \{l_{k-1}(v) : v \in \mathcal{N}(u)\})$.
- Based on the **graph isomorphism test** of Weisfeiler and Lehman, 1968.



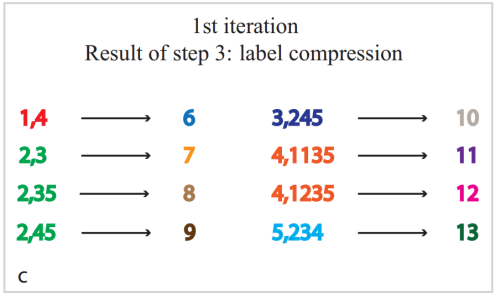


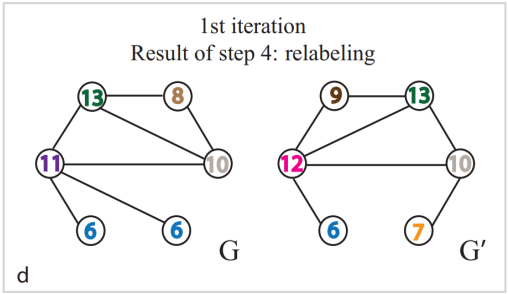
Pictures courtesy of Shervashidze et al. [2011].

Principles of the Weisfeiler-Lehman kernel

Consider a graph $G = (V, E, a)$ with discrete labels $l_0(u) = a(u)$ at each vertex u.

- This is a **multi-layer** construction producing new labels $l_k(u)$ for each vertex at layer k.
- A label $l_k(u)$ represents $(l_{k-1}(u), \{l_{k-1}(v) : v \in \mathcal{N}(u)\})$.
- Based on the **graph isomorphism test** of Weisfeiler and Lehman, 1968.

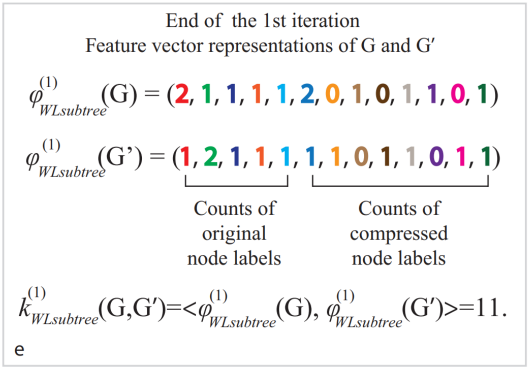




Pictures courtesy of Shervashidze et al. [2011].

Principles of the Weisfeiler-Lehman kernel

- The final representation is a histogram of **label occurences.**
- Extensions with **substructure enumeration**.



Pictures courtesy of Shervashidze et al. [2011].

Principles of the Weisfeiler-Lehman kernel

Given a graph $G = (V, E, a)$ with discrete labels $l_0(u) = a(u)$ in $\mathcal{A}_0$ for all u in V.

The Weisfeiler-Lehmann kernel representation

- **Representation at layer** k: Label $l_k(u) \in \mathcal{A}_k$ for all u in V.
- **Construction of layer** k (**message passing**):

$$l_k(u) = \text{Relabel}(l_{k-1}(u), \{l_{k-1}(v) : v \in \mathcal{N}(u)\}).$$

- **Last layer representation** with **global aggregation**:

$$\varphi_{\text{WL}}(G) = \sum_{v \in V} \text{one-hot-encoding}(l_K(u)) \in \mathbb{R}^{|\mathcal{A}|}.$$

Principles of graph neural networks with message passing

Given a graph $G = (V, E, a)$ with continous attributes $\varphi_0(u) = a(u)$ in $\mathbb{R}^{p_0}$ for all u in V.

Canonical form of message passing architecture

- **Representation at layer** k: $\varphi_k(u) \in \mathbb{R}^{p_k}$ for all u in V.
- **Construction of layer** k (**message passing**):

$$\begin{aligned}\varphi_k(u) &= \mathsf{Process}(\varphi_{k-1}(u), \{\varphi_{k-1}(v) : v \in \mathcal{N}(u)\}) \\ &= \sum_{v \in \mathcal{N}(u) \cup u} \mathsf{ReLU}(Z_k^\top \varphi_{k-1}(v)) \quad \text{(for example)}\end{aligned}$$

- **Last layer representation** with **global pooling**:

$$\varphi_{\mathsf{GNN}}(G) = \sum_{v \in V} \varphi_L(u) \in \mathbb{R}^{p_K}.$$

References I

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020.

References II

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *CVPR*, 2015.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 2014.

Alexandre Duval, Simon V Mathis, Chaitanya K Joshi, Victor Schmidt, Santiago Miret, Fragkiskos D Malliaros, Taco Cohen, Pietro Lio, Yoshua Bengio, and Michael Bronstein. A hitchhiker's guide to geometric gnns for 3d atomic systems. *arXiv preprint arXiv:2312.07511*, 2023.

VP Dwivedi and X Bresson. A generalization of transformer networks to graphs. arxiv. *arXiv preprint arXiv:2012.09699*, 2020.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *COLT*, pages 129–143. Springer, 2003.

References III

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 321–328, 2003.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Nils M Kriege, Marion Neumann, Christopher Morris, Kristian Kersting, and Petra Mutzel. A unifying view of explicit and implicit feature maps of graph kernels. *Data Mining and Knowledge Discovery*, 33(6):1505–1547, 2019.

Tao Lei, Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Deriving neural architectures from sequence and graph kernels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

References IV

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 70, 2004.

Romain Menegaux, Emmanuel Jehanno, Margot Selosse, and Julien Mairal. Self-attention in colors: Another take on encoding graph structure in transformers. *Transactions on Machine Learning Research (TMLR)*, 2023.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *arXiv preprint arXiv:2302.04181*, 2023.

References V

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR, 2009.

References VI

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research (JMLR)*, 12: 2539–2561, 2011.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.