AI for healthcare

Lecture 4/4 – Geometric deep learning

Jean Feydy
HeKA team, Inria Paris, Inserm, Université Paris-Cité

Thursday, 2pm–5pm – 4 lectures

Epita, rooms KB404 + SM15

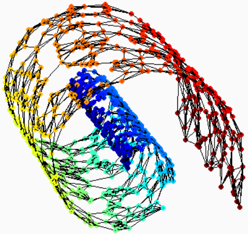
Validation: team project + quizz

Recap of the previous lecture

Lecture 3 – **Graphs**:

- **Curse of dimensionality** $\implies$ Statistics on **white noise** is hopeless.
- We must understand the **intrinsic** structure of our data,
  even when it is **embedded** in a high-dimensional space of features.
- We may need to **unwrap** the manifold of plausible data samples.

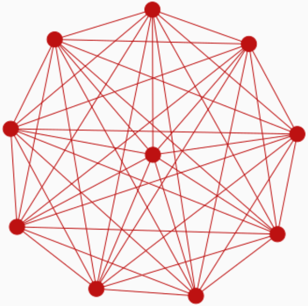
Graphs = local neighborhood structures

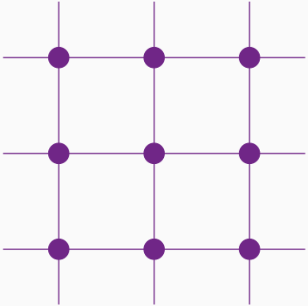


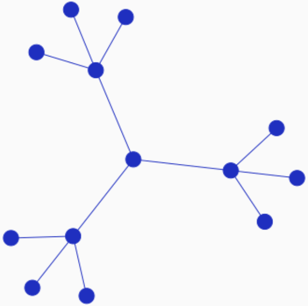
The K-NN graph describes
the **local** structure of a dataset.

Untangling a soup of edges to produce
a **global** understanding is hard.

Simple archetypes [TDGC+21]





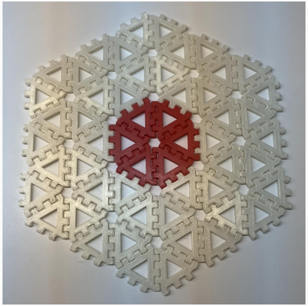


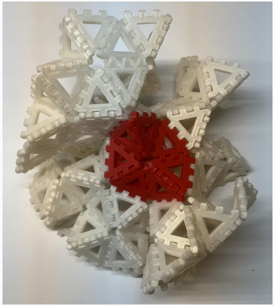
Cliques are like balls with positive curvature.

Grids are like planes with flat curvature.

Trees are like saddles with negative curvature.

Simple archetypes



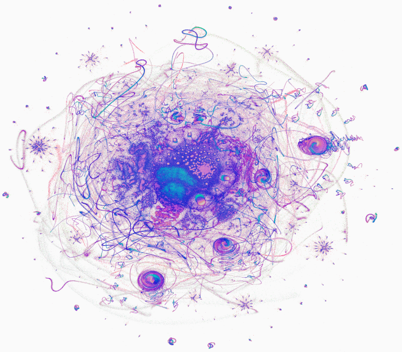


Cliques are like balls
with positive curvature.

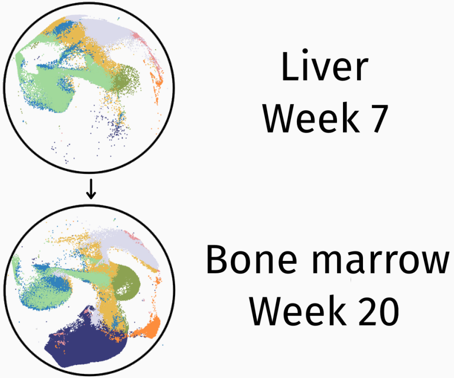
Grids are like planes
with flat curvature.

Trees are like saddles
with negative curvature.

Embedding methods such as UMAP are excellent diagnostic tools [Wil]

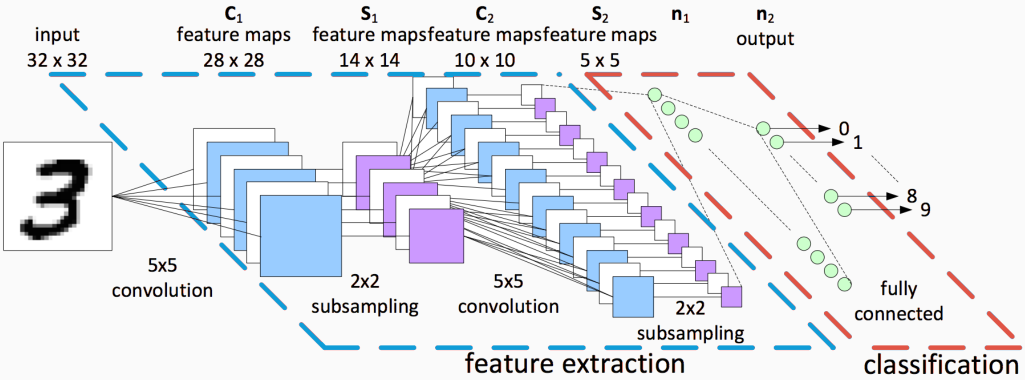


Visualizing the set of **integer numbers** 1, 2, 3, ..., 8,000,000.



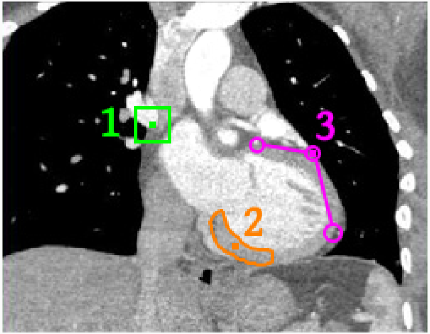
Visualizing the differentiation of Hematopoietic **stem cells**.

Since 2012, we have grown used to convolutional neural networks [PMC11]



Can we apply this methodology to higher-level descriptions? [EPW11, Man11]

1. Pixels 2. Anatomy 3. Function



Simplifying a bit, each level of analysis corresponds to a way of **grouping pixels** with their neighbors.

Today’s lecture – an active research topic

1. **Geometric deep learning is *not* a mature field:**
    - Convolutions on graphs and point clouds.
    - A stimulating environment.
    - Questionable benchmarks.
2. **Personal experience feedback:**
    - Protein docking.
    - Lung registration.
3. **Trying to learn a graph structure:**
    - Dynamic graphs, auto-encoders and transformers.
    - A continuous spectrum of models and jobs.

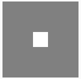
Geometric deep learning is *not* a mature field

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “a · x”,

parameterized by the coefficients of a **small filter** φ.



$\star$

$=$

φ x $\varphi \star x$

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "a · x",

parameterized by the coefficients of a **small filter** φ.

φ x $\varphi \star x$

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "a · x",

parameterized by the coefficients of a **small filter** φ.

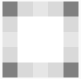
φ x $\varphi \star x$

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** “a · x”,

parameterized by the coefficients of a **small filter** φ.

 $\star$  $=$ 

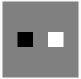
φ x $\varphi \star x$

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

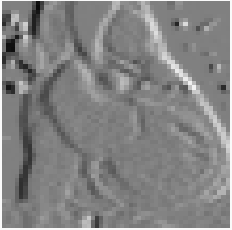
Cheap generalization of the **product** "a · x",

parameterized by the coefficients of a **small filter** φ.



$\star$

$=$

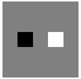


φ x $\varphi \star x$

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.



$\star$

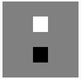
$=$

φ $\quad$ x $\quad$ $|\varphi \star x|$

Convolution (i.e. weighted average of the neighboring pixels) :

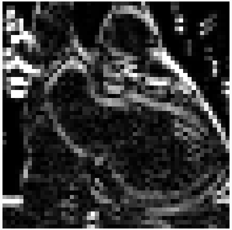
Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.



$\star$

$=$



φ $\quad$ x $\quad$ $|\varphi \star x|$

What is a convolution?

Convolution (i.e. weighted average of the neighboring pixels) :

Cheap generalization of the **product** "$a \cdot x$",

parameterized by the coefficients of a **small filter** φ.

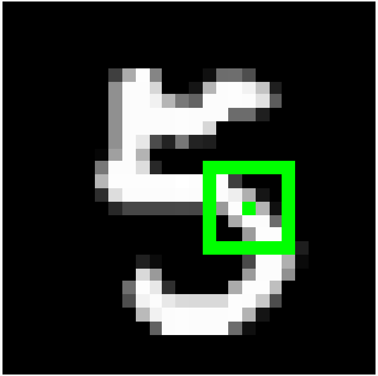
φ $\quad$ x $\quad$ $|\varphi \star x|$

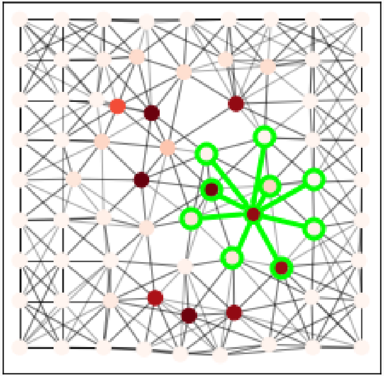
What is a convolution?

Convolutions on **grids**:

- Are **cheap**.
- Enable **pattern** detection and **texture** analysis.
- **Proven track record** since the 1960's:
  Gaussian blur, edge detectors, Laplacian pyramids,
  wavelets, JPEG2000, convolutional neural networks…

Message-passing on graphs [DJL+20]

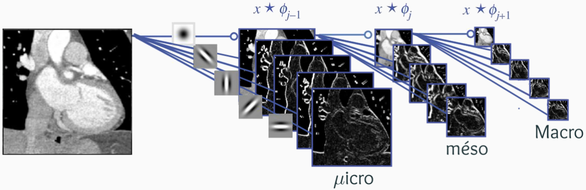


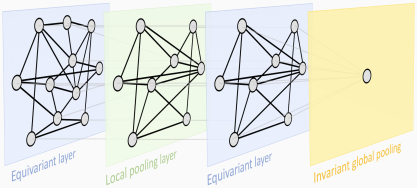


$$x[i,j] \leftarrow \sum_{k,l} \varphi[k,l] \cdot x[i-k, j-l]$$

$$x[i] \leftarrow \sum_{i \leftrightarrow k} \varphi(x[i], x[k], \mathrm{edge}[i,k])$$

Multiscale architectures on graphs [Mal16, BBCV21]





Grid convolutions
+ downsamplings.

Graph convolutions
+ downsamplings.

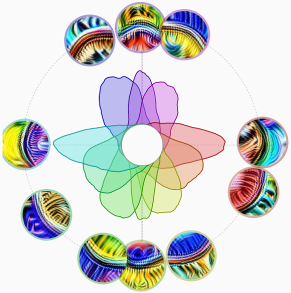
The promises of geometric deep learning

We intend to leverage the **intrinsic structure** of:

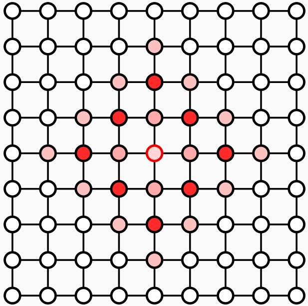
- **Point** clouds.
- Surface and volume **meshes**.
- **Molecular** graphs, proteins.
- Social and communication **networks**.

Unfortunately, things are **not that simple**.

Problem 1: How do we deal with the lack of orientation? [CGC+20, NoJ07]



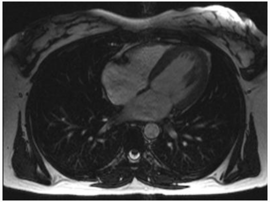
CNNs learn **oriented** curve detectors.



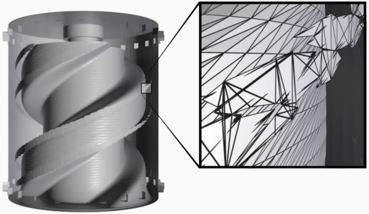
Vanilla graph convolutions define **isotropic** filters.

Hairy ball theorem: no globally consistent 2D coordinates on a sphere.

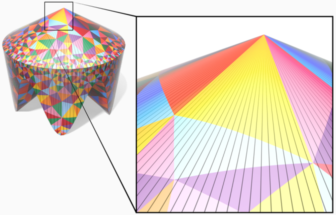
Problem 2: How do we deal with varying sampling densities? [SSC19]



MRI slice:
voxel size = 1 mm^3.

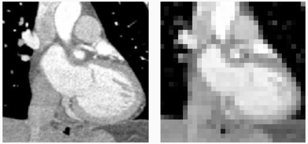


“**Hell** is other people’s meshes”
– Jean-Paul Sartre

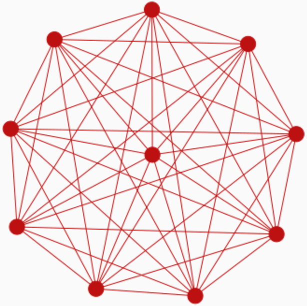


Intrinsic triangulations.
Can we use them for ML?

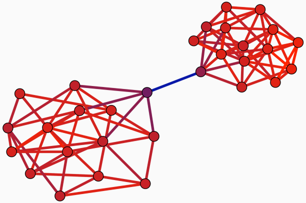
Problem 3: How do we deal with highly non-Euclidean graphs? [TDGC+21]



Downsampling a **grid** is easy.

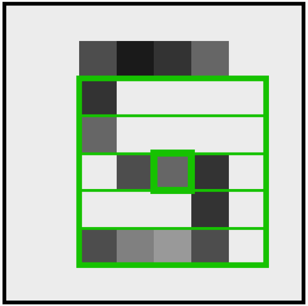


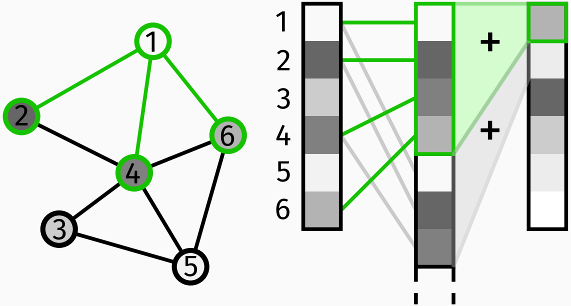
How do we downsample a **clique**?



How do we handle **bottlenecks**?

Problem 4: GPUs are not optimized for graphs





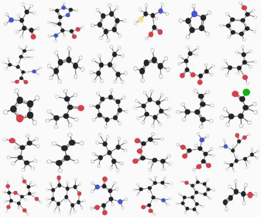
Fixed + **contiguous** neighborhoods

$\Longrightarrow$ Optimal compilation.

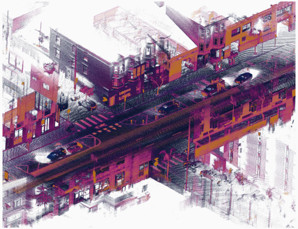
Varying sizes + **random** memory accesses

$\Longrightarrow$ x100 slow-down.

Problem 5: The range of target applications is too wide. [Dil15, Lu19, Gra19]



Molecules.



Lidar scans.

Social networks.

A fast-growing community

In 2017, geometric deep learning was **a solution in search of a problem**.

With an appealing pitch, it attracted:

- **Computer scientists**, looking for new ways of combining features.
- **Mathematicians**, looking for new applications of their insights.
- **Domain experts**, looking for a breakthrough on their data.

A blooming corpus of methods

The literature since 2017:

- Dozens of different **convolution** operators.
- Renewed interest in **theoretical graph ML**.
- Useful **extensions** for PyTorch: PyG, DGL, KeOps...
- **Cross-pollination** between different fields:
  computer graphics, signal processing, chemistry...

A field without consensus

Unfortunately, it is hard to recommend some methods over the others:

- Applications are **wildly different** from each other.
- Most theoretical and experimental works are **proof-of-concepts**.
- Benchmarks are **highly unreliable**.

We like to think that science is done in a vacuum

Archimedes – “Don’t disturb my circles”.

Louis Pasteur, alone in the lab.

Scientific literatures are shaped by structural incentives

French mathematicians – top 3 deepest ideas over a career:

- Promotes **long-term**, original thought.
- No incentives for outreach and interdisciplinary research.

INRIA researchers – 1 meaningful contribution every 5 years:

- A theorem, a piece of software, a patent, a societal breakthrough…
- Outstanding place for **applied** research.

French medical doctors/teachers/researchers – PubMed index:

- MERRI funds compensate hospitals for research activities.
- Fall behind the profitability threshold $\Longrightarrow$ shut down the lab, focus on care.
- **Inflation** of low-quality papers.

Structural incentives in US/UK CS teams

Universities – climb up the Shanghai ranking:

- 1 (foreign) student = \$30k-\$70k per year.
- Maximize the **number** of papers to impress prospective students.

Professors – maximize the hype:

- Unstable "tenure track" contract, commited to a grant, startup to grow…
- Allowed to take risks, but **not allowed to fail**.

Students – target a job in the tech companies:

- Significant debt to pay back.
- Safest route: **incremental** research, claim SOTA every 6 months.

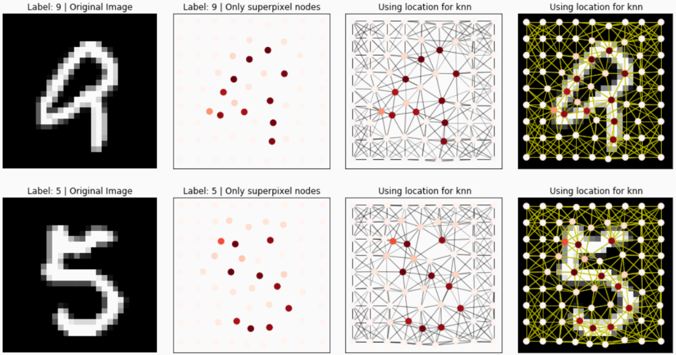
Peer-reviewed $\neq$ Scientific truth

In the current ML community, relatively **few actors are incentivized** to:

- Review papers carefully.
- Document **baselines** thoroughly.
- Admit that an idea **doesn't work**.
- Take the **massive** amount of time that is needed to make their experiments **truly** reproducible.
- Keep the codebase up to date with a software stack that breaks every 6 months.

$\Longrightarrow$ Unfortunately, the NeurIPS/ICML/... stamp means little.
We cannot **trust** the conclusions of a paper
without **reproducing** the experiments
or **checking** the proofs carefully.

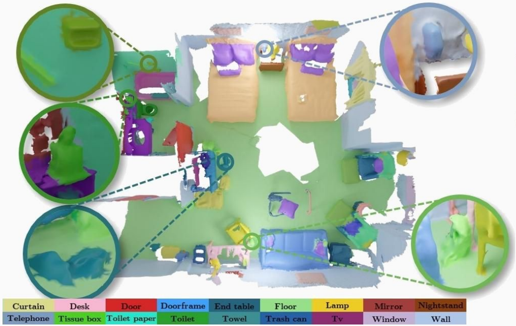
Some common benchmarks are nothing but sanity checks [LCCB98, DJL+20]

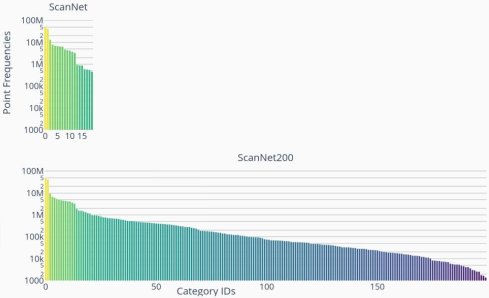


Model	L	MNIST #Param	MNIST Test Acc.±s.d.
MLP	4	104044	95.340±0.138
vanilla GCN	4	101365	90.705±0.218
GraphSage	4	104337	**97.312±0.097**
GCN	4	101365	90.120±0.145
MoNet	4	104049	90.805±0.032
GAT	4	110400	95.535±0.205
GatedGCN	4	104217	**97.340±0.143**
GIN	4	105434	96.485±0.252
RingGNN	2	105398	11.350±0.000
	2	505182	91.860±0.449
	8	506357	Diverged
3WLGNN	3	108024	95.075±0.961
	3	501690	95.002±0.419
	8	500816	Diverged

Results on **Graph-MNIST** look promising… Until you remember that test accuracy for **basic K-NN classifiers** on MNIST is 95%-99%.

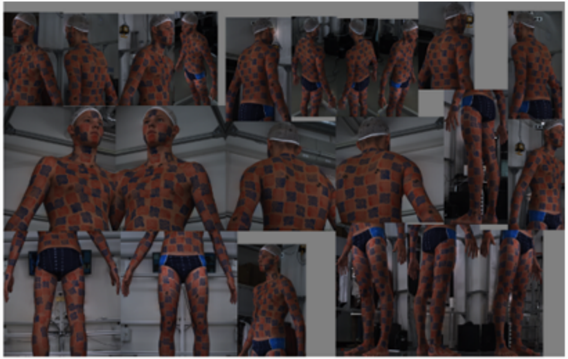
Some common benchmarks have been saturated for years [RLD22]

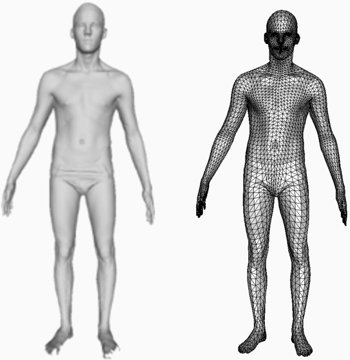




Challenging large-scale benchmarks are published every year…
But the review system pushes authors towards **overfitting on "classic" datasets**.

Some benchmarks have been taken out of context [BRLB14]





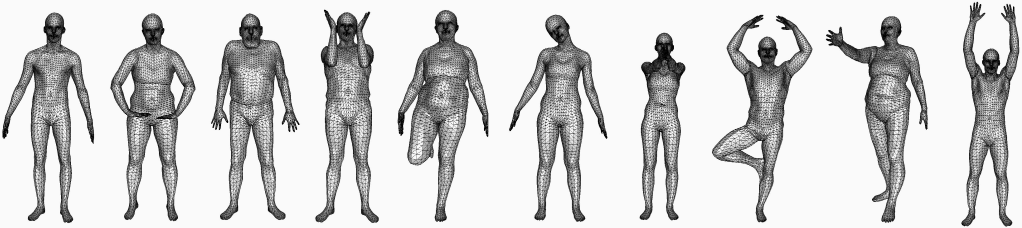
The MPI FAUST dataset contains 300 real-life scans of 10 subjects – 30 poses each.

Problem: fit a reference mesh to the **noisy 3D point clouds**.

Challenges: different topologies, missing data, self contacts.

Dense ground truth correspondence for training = 10x10 meshes with identical topology.

Some benchmarks have been taken out of context [BRLB14]



The PyG library and many papers on graph neural networks
discard the original point clouds to focus entirely on the 100 pre-aligned meshes.
New problem: use (x, y, z) coordinates as input features for each node
$\Longrightarrow$ predict node indices in $[1, 6890]$ as output signal.
Train on 80 surfaces, test on the remaining 20.

Irrelevant to shape registration: we learn to **overfit on the reference triangulation**.

How can we judge?

Keep track of both **classical** and **learning-based** baselines, in each application domain.

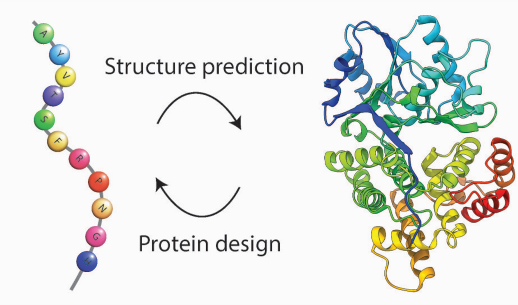
This is time-consuming – **focus on 2-3 subjects at most.**

It is likely that the wave of papers on geometric deep learning will **fade** in most applications settings, and **stick** in some productive niches: Shape analysis? Chemistry? Social sciences?

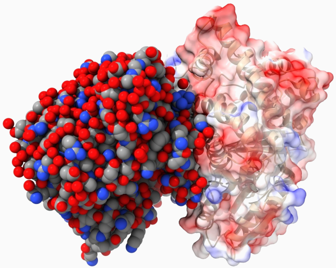
This is how scientific progress takes place. The hype cycle is a **normal** and well-documented phenomenon.

Some personal experience

Two problems in structural biology [SFCB20]

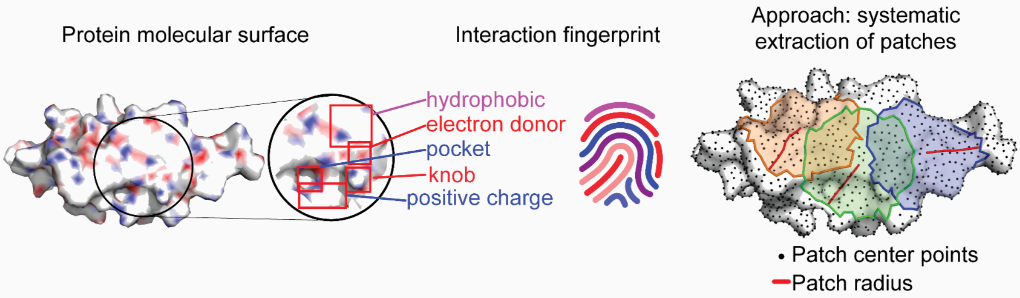


Folding and design.



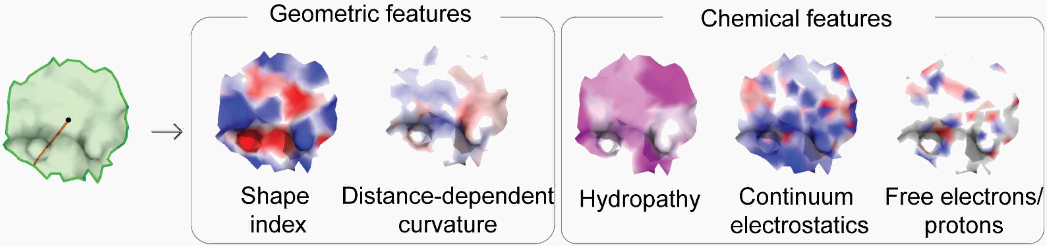
Docking – interacting surfaces.

Docking is a function of the protein surfaces [GSM+20]



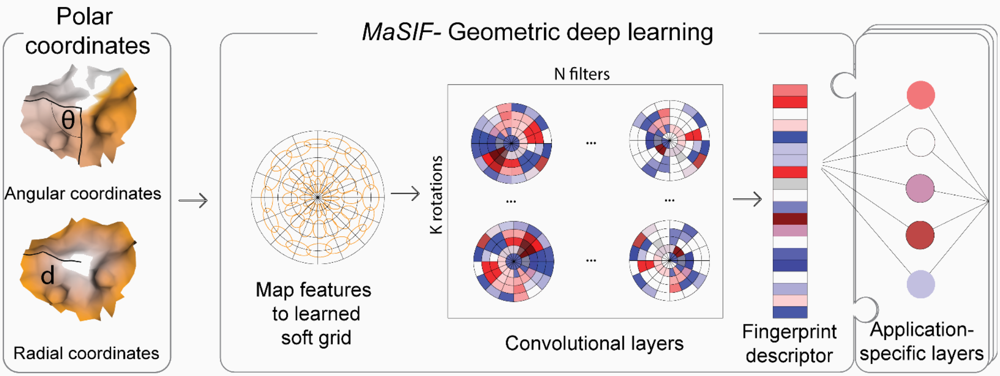
Physical prior: no need to deal with the full volumes, we are looking for **surface fingerprints**.

Docking is a function of the protein surfaces [GSM+20]



We can compute **geometric** and **chemical** **input features**
on local patches with $\simeq 1\,\mathrm{nm}$ radius.

Docking is a function of the protein surfaces [GSM+20]

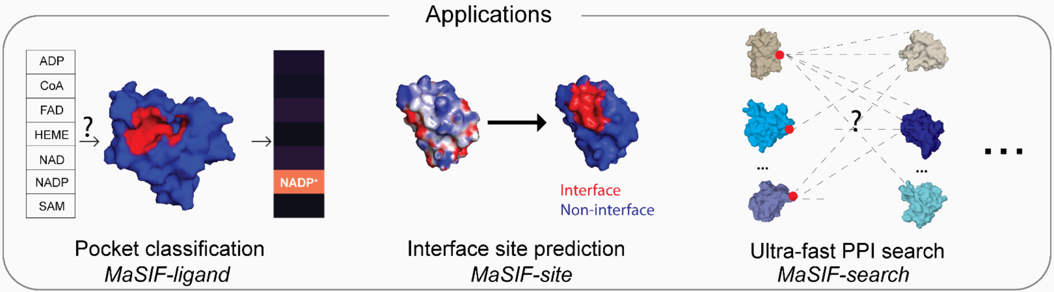


And perform all classification tasks on **geodesic patches**.

Built-in **invariance** to 3D rotations, translations and the inner content of the protein.

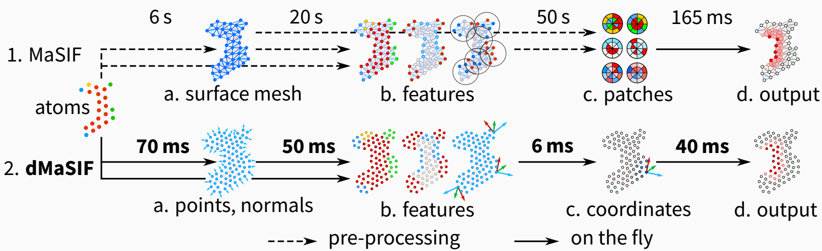
Enforces a **locality** prior: won't overfit on scattered patterns.

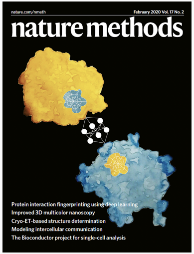
Docking is a function of the protein surfaces [GSM+20]



After training on the Protein Data Bank (with **careful** train-test split), this enables three tasks of interest.

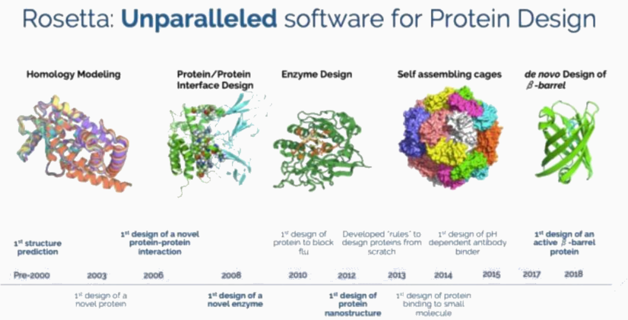
A proof-of-concept that has been received well [GSM+20, SFCB20, SFS+22]

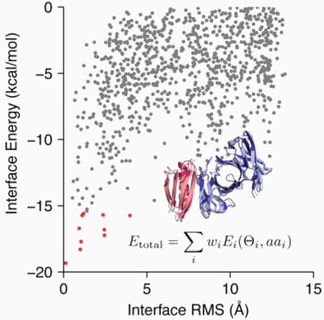




$\longrightarrow$ $\times$100 - $\times$1,000 faster, lighter and fully differentiable.

Is deep learning truly a revolution? [ALFJ+17]





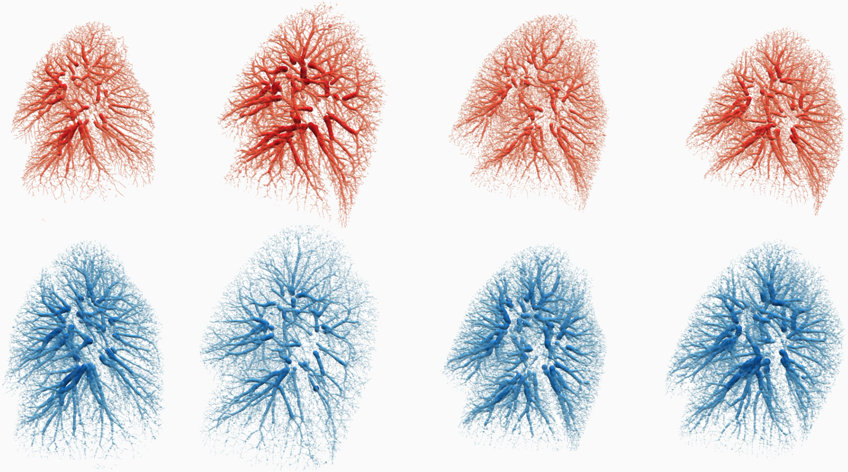
After 20 years of work, the developers of the Rosetta software have *de facto* developed a **hybrid "point neural network"** that combines **physical** potentials with **data-driven** residuals.

Recap on the MaSIF-dMaSIF project

Personal feedback:

- Inspiring **prototype** that shows how far we can go if we
  drop complex physical terms but **preserve** the symmetries of the problem.
- With clever geometry and code, we can do interesting science on a **single GPU**.
- Nowhere near close to a finished product: **maintaining** software is a full-time job.
- **Well-funded** and **mature** fields already showcase formidable baselines.
  As outsiders, we can propose **stimulating** ideas and tools.
  Always **stay humble** – understanding the full context takes years.

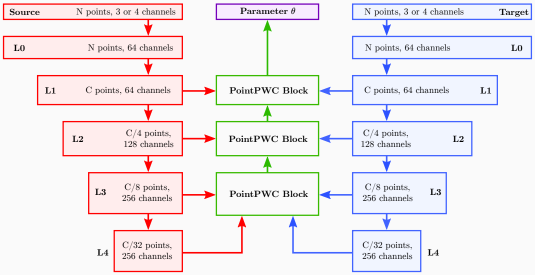
Lung registration “Exhale – Inhale” [SFL+21]



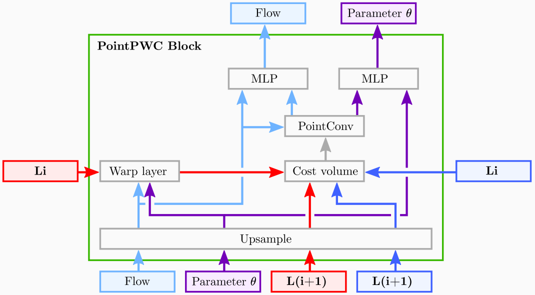
Complex deformations, high **resolution** (50k–300k points), high **accuracy** (< 1mm).

Point clouds are more challenging than volumes, but **robust** to acquisition parameters.

Point neural networks [WWL+20, SFL+21]

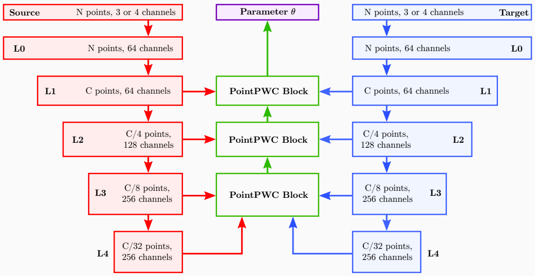


Multi-scale convolutional point neural network.



Architecture of a PointPWC **block**.

Point neural networks – strengths and limitations [SFL+21]



Multi-scale convolutional point neural network.

Point neural nets, **in practice**:

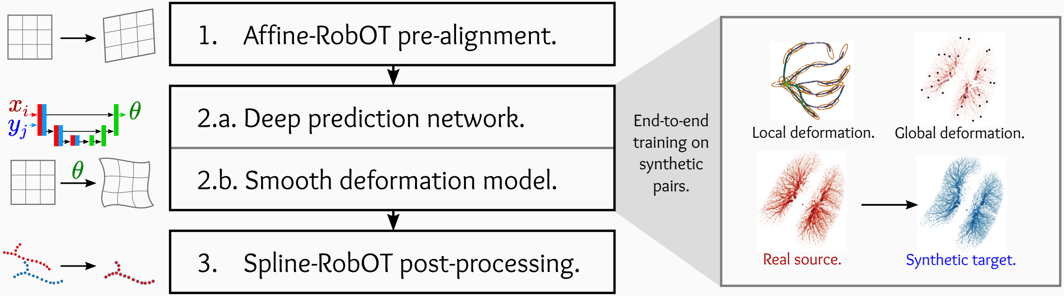
- Compute **descriptors** at all scales.
- **Match** them using geometric layers.
- Train on **synthetic** deformations.

Strengths and weaknesses:

- Good at **pairing** branches.
- Hard to train to high **accuracy**.

$\Longrightarrow$ **Complementary** to geometric methods such as optimal transport.

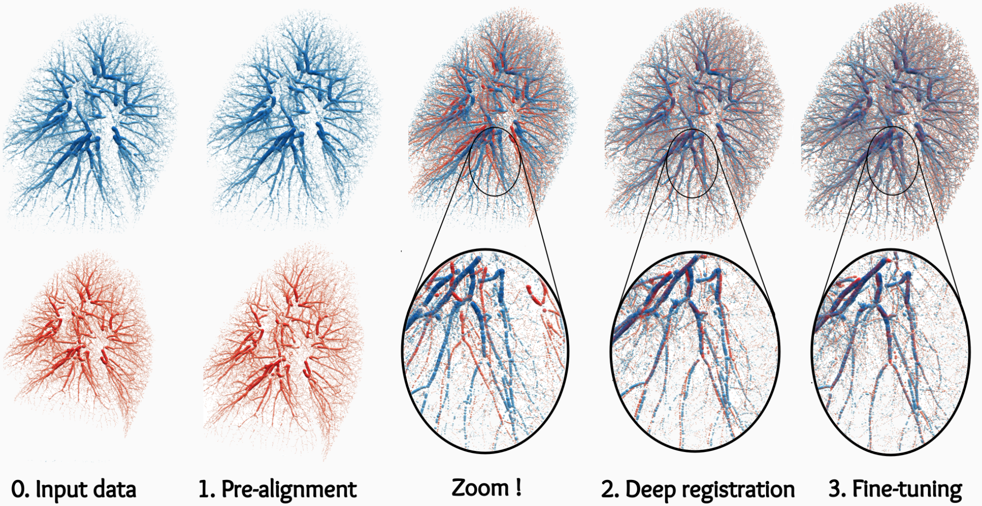
Three-steps registration [SFL+21]



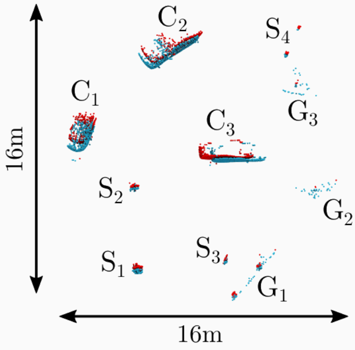
This **pragmatic** method:

- Is **easy to train** on synthetic data.
- Scales up to high-resolution: 100k points in 1s.
- Excellent results: **KITTI** (outdoors scans) and **DirLab** (lungs).

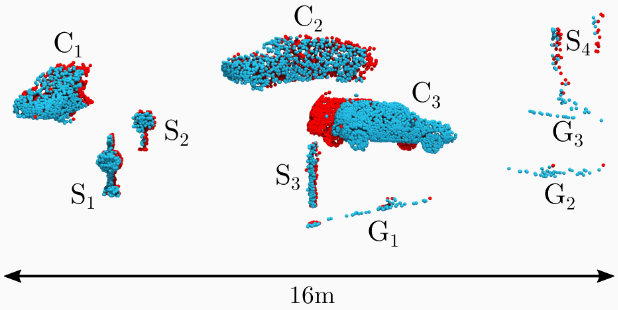
Three-steps registration



Evaluation on the 3D Kitti dataset [MG15, MHG15, SFV+19]



(a) View from above.



(b) Front view.

3D scene flow estimation task, derived from a **2D + depth** dataset.

This is a standard (but maybe questionable) benchmark.

Evaluation on the 3D Kitti dataset [SFL+21]

	Method	Points	Time ms ↓	Memory Mb ↓	EPE3D cm ↓	Acc3DS % ↑	Acc3DR % ↑	Outliers3D % ↓	EPE2D px ↓	Acc2D % ↑
Unsupervised	ICP (rigid) [7]	8k	224	**2**	51.81	6.69	16.67	87.12	27.6752	10.56
	FGR (rigid) [134]	8k	—	30	48.35	13.31	28.51	77.61	18.7464	28.76
	CPD (non-rigid) [81]	8k	34,880	798	41.44	20.58	40.01	71.46	27.0583	19.80
	PWC (self) [126]	8k	237	1,016	25.49	23.79	49.57	68.63	8.9439	32.99
	RobOT (raw)	8k	170	3	**9.12**	**60.43**	**79.39**	**33.65**	**4.9920**	**56.23**
	RobOT (raw)	30k	**166**	**89**	**4.67**	**80.43**	**91.05**	**20.21**	**1.7026**	**85.71**
Supervised training on FlyingThings3D	FlowNet3D [69]	8k	—	690	17.67	37.38	66.77	52.71	7.2141	50.92
	SPLATFlowNet [111]	8k	—	—	19.88	21.74	53.91	65.75	8.2306	41.89
	original BCL [49]	8k	—	—	17.29	25.16	60.11	62.15	7.3476	44.11
	HPLFlowNet [49]	8k	—	—	11.69	47.83	77.76	41.03	4.8055	59.38
	FLOT [49]	8k	324	2,826	5.51	75.79	90.98	23.95	3.3152	75.10
	PWC [126]	8k	**237**	1,016	5.28	85.83	94.08	18.85	3.0074	81.48
	PWC	30k	1,138	10,691	7.63	67.54	92.30	26.09	3.5212	70.74
	Pre + FLOT + Post	8k	487	2,826	5.33	76.84	91.65	23.56	3.2786	75.31
	Pre + PWC + Post	8k	279	1,034	3.35	90.10	97.32	**16.20**	**1.4301**	**93.85**
	Pre + PWC + Post	30k	1,207	10,691	3.50	90.04	96.71	17.28	1.5917	90.37
	D-RobOT (spline)	8k	268	**396**	**3.15**	**90.51**	**97.42**	16.26	1.4532	93.76
	D-RobOT (spline)	30k	**547**	**610**	**2.23**	**95.88**	**99.19**	**12.89**	**1.0336**	**96.75**

Surprise: a **baseline** optimal transport solver with smoothing (RobOT) outperforms many **deep learning** methods on all metrics.

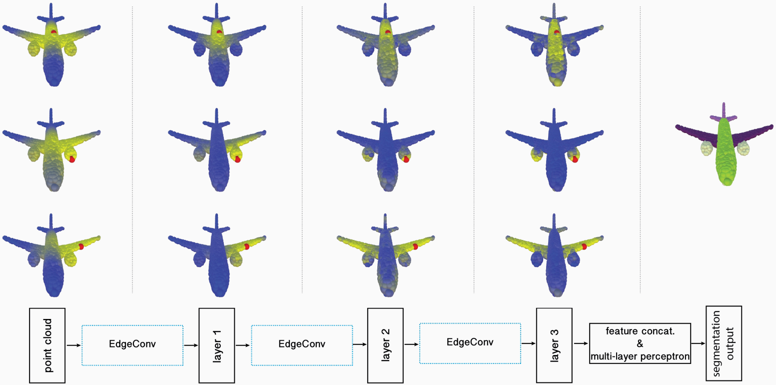
Recap on the registration project [CCF+13]

Personal feedback:

- Pragmatic approach: get the **best of both** **data-driven** and **deterministic** methods.
- **Tables** of numbers look impressive – but can **hide** critical information.
- **Code** base is not clean – top priority going forward.
- Evaluation on 3D scene flow is **questionable**.
  Shouldn't we work with the original 2D data?
- Evaluation on **medical data** is hard:
  - We release a new dataset of **1,000 pairs for training** – without annotations.
  - We only have access to **10 pairs** with dense annotations **for evaluation**.

Should we learn our graphs?

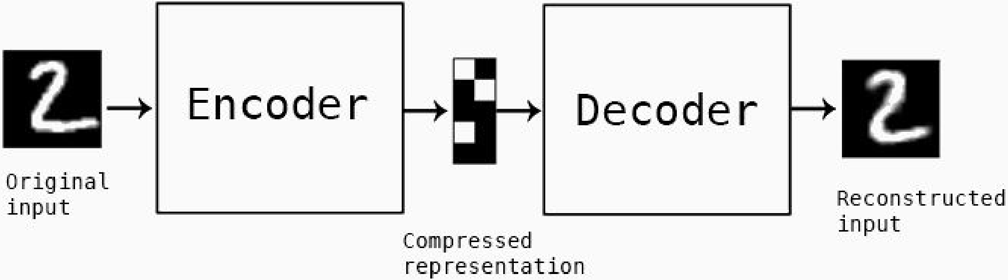
Dynamic graph CNNs [WSL+19]



At every layer, use the **K-NN graph** of the **feature vectors**.

This may promote **semantic** neighborhoods.

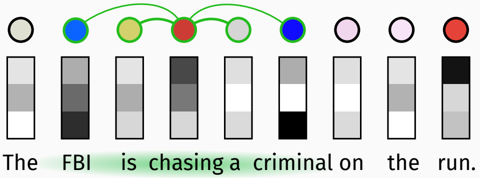
Embeddings and autoencoders [Cho16]



Use the **K-NN graph** of a **low-dimensional** representation:

UMAP, an encoder-decoder architecture…

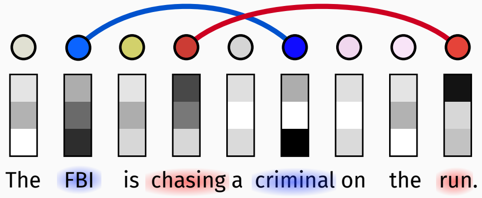
Attention layers and transformers [CDL16, VSP+17]

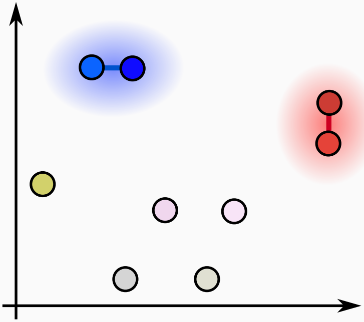


We would like to define convolutional architectures on **sentences**.

But the **1D structure** of the text is not always relevant.

Attention layers and transformers [CDL16, VSP+17]





Instead, we **embed feature vectors** in a semantic space.

Attention $=$ data-driven embedding + convolution.

Transformer $=$ stack of attention layers.

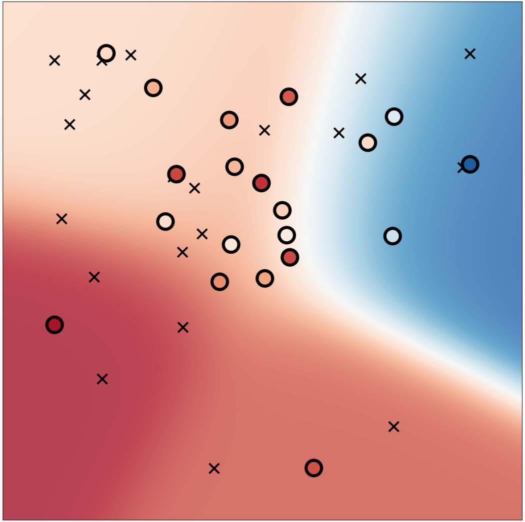
Attention layers and transformers [CDL16, VSP+17]

Cheap and expressive convolution on the feature vectors x_i:

- Query vectors $q_i \leftarrow \text{MLP}_q(x_i)$.
- Key vectors $k_i \leftarrow \text{MLP}_k(x_i)$.
- Value vectors $v_i \leftarrow \text{MLP}_v(x_i)$.
- **Nadaraya–Watson interpolation** with a kernel $k(x, y) = \exp(x \cdot y)$:

$$x_i \leftarrow \frac{\sum_j \exp(q_i \cdot k_j)\, v_j}{\sum_j \exp(q_i \cdot k_j)}$$

This architecture trains remarkably well on **web-scale** corpuses.



Problem 1: In most important settings, data is expensive and fragmented [Ora17]

The Web and the Amazon Mechanical Turk are **the exceptions**, not the rule:

- **1** protein structure in 3D
  $=$ **months** of work for a biologist.
- **50** fully documented patients
  $=$ a great **thesis** for a radiologist.
- **1** patient in a clinical trial
  $\simeq$ **\$40k**, up to \$1M for surgery!

- Getting **legal** access to large datasets *and* computers is extremely difficult in any field that has a signficant **societal impact**.

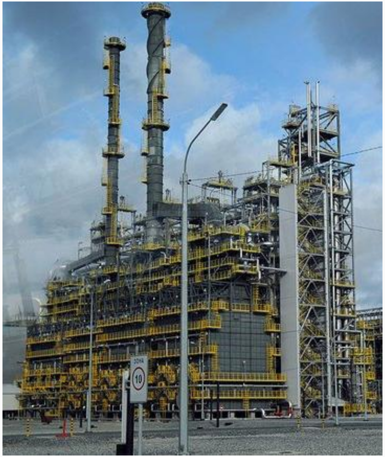
Problem 2: Bigger models are slower, heavier and cumbersome [otRF98]

We must **strike a balance** between:

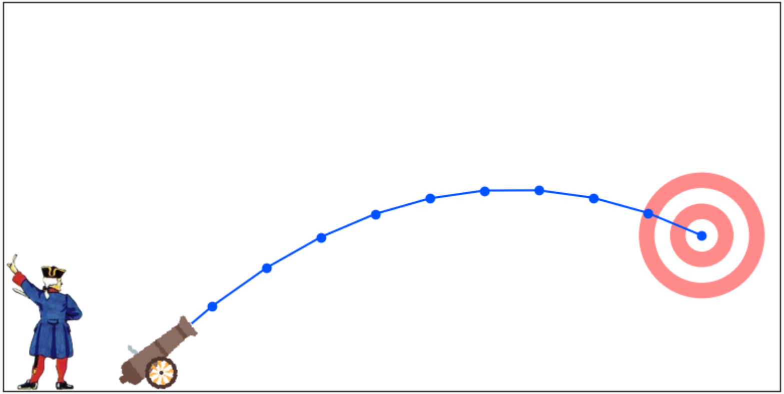
- Speed.
- Accuracy.
- **Hardware** and **development** costs.
- **Energy** consumption $=$ battery life.
- **Interpretability** $=$ easy to debug
  $=$ easy to **maintain**.

Tons of equations $\simeq$ Tons of parameters.

Only worth the **trouble** if there is no other option.



Problem 3: In mature fields, structure is key to generalize

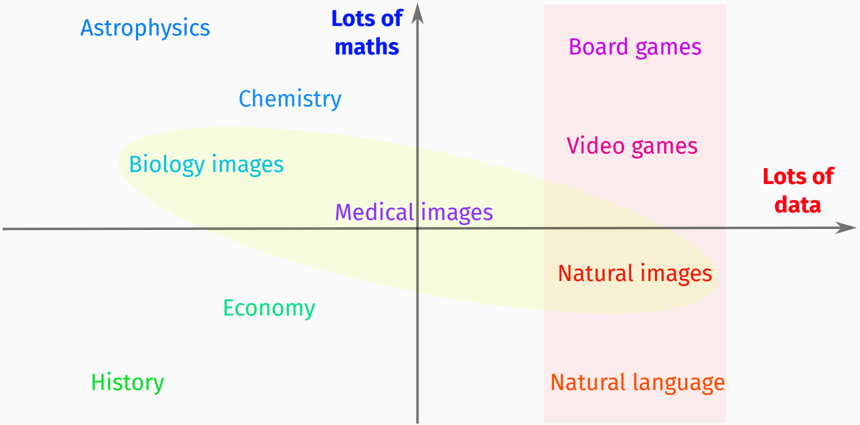


This is especially true in fields that are close to physics, robotics…

Fitting piecewise **linear** functions to **parabolas** or **exponentials**

is fine for **interpolation**, but very limiting for **extrapolation**.

A continuous spectrum of scientific fields



"The **unreasonable effectiveness** of mathematics in the natural sciences" and "the **bitter lesson** of AI" apply to **different ends** of this spectrum.

Monopolies claim that “bigger is better”

Electric bikes travel
at **30 km/h**.

Private jets travel
at **800 km/h**.

Rockets travel
at **30,000 km/h**.

Raw performance metrics **do not tell the whole story.**

Choose your own path

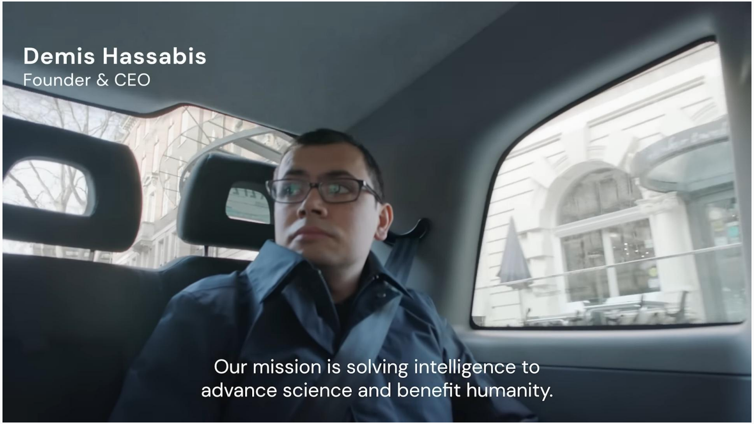
Tech companies target your "fear of missing out"

"We should **stop** training radiologists **now**.

It's just **completely obvious** that within **five years**, deep learning is going to do **better** than radiologists."

– Geoffrey Hinton, Google Brain and UToronto, **in 2016**.

Tech companies want you to forget that targeting ads $\neq$ solving cancer



"Welcome to **DeepMind**: Embarking on **one of the greatest adventures** in scientific history", available on YouTube since September 29, 2022.

2020-2050: tough times ahead. Which problem are you going to solve?

Quantum computers

Chess engine++

Killer drones

Efficient batteries

Autonomous cars

Automated MRI screening

Nuclear fusion

Targeted ads

Drug safety

Protein design

Efficient rail network

Machine translation

The French education system is giving you **genuine freedom** of choice.

Use it **wisely**.

Find your own balance

References

References i

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al.

The Rosetta all-atom energy function for macromolecular modeling and design.

Journal of chemical theory and computation, 13(6):3031–3048, 2017.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic.

Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.

arXiv preprint arXiv:2104.13478, 2021.

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black.

FAUST: Dataset and evaluation for 3D mesh registration.

In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

Richard Castillo, Edward Castillo, David Fuentes, Moiz Ahmad, Abbie M Wood, Michelle S Ludwig, and Thomas Guerrero.

A reference dataset for deformable image registration spatial accuracy evaluation using the COPDgene study archive.

Physics in Medicine & Biology, 58(9):2861, 2013.

Jianpeng Cheng, Li Dong, and Mirella Lapata.

Long short-term memory-networks for machine reading.

arXiv preprint arXiv:1601.06733, 2016.

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah.

Curve detectors.

Distill, 2020.

https://distill.pub/2020/circuits/curve-detectors.

François Chollet.

Building autoencoders in keras.

https://blog.keras.io/building-autoencoders-in-keras.html, 2016.

James Dillon.

Molecules.

https://github.com/chemplexity/molecules, 2015.

MIT License.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson.

Benchmarking graph neural networks.

arXiv preprint arXiv:2003.00982, 2020.

References vi

Olivier Ecabert, Jochen Peters, and Matthew Walker.

Segmentation of the heart and great vessels in ct images using a model-based adaptation framework.

Medical Image Analysis, (15):863–876, 2011.

Martin Grandjean.

Social network analysis visualization.

https://fr.wikipedia.org/wiki/Fichier:Social_Network_Analysis_Visualization.png, 2019.

CC BY-SA 3.0.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia.

Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning.

Nature Methods, 17(2):184–192, 2020.

Yann LeCun, Corinna Corinna Cortes, and Christopher J.C. Burges.

The mnist database of handwritten digits.

http://yann.lecun.com/exdb/mnist/, 1998.

Daniel L. Lu.

Ouster os1-64 lidar point cloud of intersection of folsom and dore st, san francisco.

https://en.wikipedia.org/wiki/File:Ouster_OS1-64_lidar_point_cloud_of_intersection_of_Folsom_and_Dore_St,_San_Francisco.png, 2019.

CC BY 4.0.

Stéphane Mallat.

Understanding deep convolutional networks.

Phil. Trans. R. Soc. A, 374(2065):20150203, 2016.

Tomaso Mansi.

A statistical model for quantification and prediction of cardiac remodelling: Application to tetralogy of fallot.

IEEE transactions on medical imaging, 2011.

Moritz Menze and Andreas Geiger.

Object scene flow for autonomous vehicles.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

References x

Moritz Menze, Christian Heipke, and Andreas Geiger.

Joint 3d estimation of vehicles and scene flow.

ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 2:427, 2015.

NoJhan.

Hairy head of a one year old girl.

https://commons.wikimedia.org/wiki/File:Baby_hairy_head_DSCN2483.jpg, 2007.

CC BY-SA 2.0.

Grégoire Orain.

Sur amazon turk, les forçats du clic.

https://www.lemonde.fr/pixels/article/2017/05/22/les-damnes-de-la-toile_5131443_4408996.html, 2017.

Rubrique Pixels du Monde.

President of the Russian Federation.

Tobolsk-polimer chemical plant.

https://commons.wikimedia.org/wiki/File:Tobolsk-Polimer_chemical_plant.jpeg, 1998.

CC BY-SA 4.0.

Maurice Peemen, Bart Mesman, and Henk Corporaal.

Speed sign detection and recognition by convolutional neural networks.

In *Proceedings of the 8th International Automotive Congress*, pages 162–170, 2011.

David Rozenberszki, Or Litany, and Angela Dai.

Language-grounded indoor 3d semantic segmentation in the wild.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Freyr Sverrisson, Jean Feydy, Bruno E. Correia, and Michael M. Bronstein.

Fast end-to-end learning on protein surfaces.

bioRxiv, 2020.

Zhengyang Shen, Jean Feydy, Peirong Liu, Ariel H Curiale, Ruben San Jose Estepar, Raul San Jose Estepar, and Marc Niethammer.

Accurate point cloud registration with robust optimal transport.

Advances in Neural Information Processing Systems, 34:5373–5389, 2021.

Freyr Sverrisson, Jean Feydy, Joshua Southern, Michael M Bronstein, and Bruno Correia.

Physics-informed deep neural network for rigid-body protein docking.

In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

Thibault Séjourné, Jean Feydy, François-Xavier Vialard, Alain Trouvé, and Gabriel Peyré.

Sinkhorn divergences for unbalanced optimal transport.

arXiv preprint arXiv:1910.12958, 2019.

- Nicholas Sharp, Yousuf Soliman, and Keenan Crane.

  Navigating intrinsic triangulations.

  ACM Transactions on Graphics (TOG), 38(4):1–16, 2019.

- Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein.

  Understanding over-squashing and bottlenecks on graphs via curvature.

  arXiv preprint arXiv:2111.14522, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.

Attention is all you need.

Advances in neural information processing systems, 30, 2017.

John Williamson.

What do numbers look like?

https://johnhw.github.io/umap_primes/index.md.html.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon.

Dynamic graph CNN for learning on point clouds.

Acm Transactions On Graphics (tog), 38(5):1–12, 2019.

Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin.

PointPWC-net: Cost volume on point clouds for (self-) supervised scene flow estimation.

In *European conference on computer vision*, pages 88–107. Springer, 2020.